ANHANG Q

Themen aus Vorauflagen

Cisco ändert seine Prüfungen regelmäßig ab. Dabei werden die Prüfungsbezeichnungen geändert, desgleichen die Prüfungsnummern, wann immer Cisco eine Prüfung durch einen neuen Entwurf ersetzt. Und jedes Mal, wenn Cisco seine Prüfungen ändert, erstellen wir neue Auflagen dieser Lehrbücher (und einmal haben wir sogar eine neue Version gemacht, ohne dass es eine neue Prüfungsversion gegeben hätte). Aber auch wenn der Verlag die Auflagennummer bei jeder größeren Umbenennung auf 1 zurücksetzt, handelt es sich hierbei um eines von zwei Büchern, die ein Gesamtprogramm für die CCNA R&S-Prüfung bilden, und bezüglich der Buchinhalte sind wir mittlerweile eigentlich schon bei der 8. Auflage angekommen.

Bei jeder Neuauflage bestimmen die Themen, die nach Maßgabe von Cisco einer Prüfung zugeordnet werden, natürlich auch die Buchinhalte. Deswegen verschwinden auch ab und zu einige Themen aus dem in den Büchern enthaltenen Material. Und manchmal habe ich den Drang, solche Themen demjenigen unter 1000 Lesern verfügbar zu machen, der sich dafür interessiert. Aus diesem Grund haben wir entschieden, einen Teil des alten Materials in diesen DVD-Anhang zu integrieren.

Der vorliegende Anhang enthält also eine Reihe von Themen, die zu dieser Kategorie gehören. Es handelt sich um Themen aus Vorauflagen und in ein oder zwei Fällen um Material im Entwurfszustand, das gar nicht zur Veröffentlichung kam. Trotzdem finden Sie diesen Stoff hier – und ich hoffe, dass Sie ihn nützlich finden. Ganz gewiss jedoch müssen Sie diesen Anhang nicht lesen, um die Prüfung in ihrer aktuellen Form zu bestehen.

Folgende Themen sind in diesem Anhang enthalten:

- Interne Verarbeitung auf Cisco-Switches
- IOS-Version und weitere neustartspezifische Informationen
- Sekundäre IP-Adressierung
- Interne Verarbeitung auf Cisco-Routern
- OSPF-Konfiguration
- Umsortierung von ACEs durch das IOS
- NAT-Overloading (PAT) auf Routern der Consumer-Klasse
- Dynamische Routen mit OSPFv3

HINWEIS Der Inhalt unter der Überschrift »Interne Verarbeitung auf Cisco-Switches« wurde 2013 für die Prüfung 100-101 in Kapitel 6 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100-101* veröffentlicht.

Interne Verarbeitung auf Cisco-Switches

Wenn ein Cisco-Switch beschließt, einen Frame weiterzuleiten, dann kann die sich aus dieser Entscheidung ergebende interne Verarbeitung unterschiedlich ausfallen. Praktisch alle neueren Switches verwenden das Store-and-Forward-Switching, doch wird jede der drei nachfolgend beschriebenen Methoden der internen Verarbeitung von mindestens einem Cisco-Switch unterstützt.

Einige Switches und grundsätzlich alle transparenten Bridges arbeiten mit dem *Store-and-Forward-Switching*. Hierbei muss der Switch den gesamten Frame vollständig empfangen haben, bevor er das erste Bit dieses Frames weiterleitet. Allerdings bietet Cisco zwei weitere interne Verarbeitungsmethoden für Switches an: *das Cut-Through-Switching* und das *Fragment-Free-Switching*. Weil die Empfänger-MAC-Adresse sehr früh im Ethernet-Header auftaucht, kann der Switch die Weiterleitungsentscheidung bereits deutlich vor Erhalt des letzten Bits im Frame treffen. Beim Cut-Through- und beim Fragment-Free-Switching startet der Switch deswegen die Weiterleitung des Frames bereits, bevor dieser vollständig empfangen wurde. So verkürzt sich die Übertragungsdauer (Latenz) für den Frame.

Beim *Cut-Through-Switching* startet der Switch den Frame-Versand über den betreffenden Ausgangsport zum frühestmöglichen Zeitpunkt. Dies kann die Latenz verringern, gleichzeitig aber zur Verbreitung von Fehlern führen. Weil sich nämlich das FCS-Feld (Frame Check Sequence) im Ethernet-Trailer befindet, kann der Switch vor Beginn der Weiterleitung nicht feststellen, ob der Frame fehlerhaft ist. Deswegen senkt der Switch zwar die Latenz des Frames, er tut dies jedoch unter dem Risiko, potenziell fehlerhafte Frames weiterzuleiten.

Das *Fragment-Free-Switching* funktioniert ähnlich wie das Cut-Through-Switching, jedoch wird hierbei versucht, die Anzahl fehlerhafter weitergeleiteter Frames möglichst klein zu halten. Eine interessante im Zusammenhang mit der CSMA/CD-Logik für Ethernet zu erwähnende Tatsache ist, dass Kollisionen innerhalb der ersten 64 Bytes eines Frames erkannt werden müssen. Das Fragment-Free-Switching nutzt die gleiche Logik wie das Cut-Through-Switching, wartet jedoch, bis die ersten 64 Bytes empfangen wurden, bevor die Weiterleitung des Frames startet. So ist die Latenz der Frames niedriger als beim Store-and-Forward-Switching und geringfügig höher als beim Cut-Through-Switching, doch werden Frames, die aufgrund von Kollisionen fehlerhaft sind, nicht weitergeleitet.

Heutzutage sind viele Desktop-PCs mit mindestens 100 Mbit/s angebunden und die Uplinks erreichen oft 1 Gbit/s oder – dank der Verwendung von ASICs (Application-Specific Integrated Circuits) – noch höhere Werte. Deswegen nutzen moderne Switches in der Regel das Store-and-Forward-Switching, weil der Latenzgewinn der beiden anderen Switching-Methoden bei solchen Datenraten als vernachlässigbar zu betrachten ist.

Die auf den Switches verwendeten internen Verarbeitungsalgorithmen unterscheiden sich von Hersteller zu Hersteller und Modell zu Modell. Trotzdem lassen sie sich stets einer der Methoden zuordnen, die in Tabelle Q.1 aufgeführt sind.

HINWEIS Der Inhalt unter der Überschrift »IOS-Version und weitere neustartspezifische Informationen« wurde 2013 für die Prüfung 100-101 in Kapitel 7 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100-101* veröffentlicht.

Tabelle Q.1 Interne Verarbeitung auf dem Switch

Switching-Methode	Beschreibung
Store-and-Forward-Switching	Der Switch empfängt alle Bits des Frames und startet erst dann die Weiterleitung. So kann der Switch das FCS-Feld überprüfen, bevor er den Frame weiterleitet.
Cut-Through-Switching	Der Switch leitet den Frame schnellstmöglich weiter. Dies verringert die Latenz, ermöglicht aber kein Verwerfen von Frames mit negativer FCS-Prüfung.
Fragment-Free-Switching	Der Switch leitet den Frame weiter, nachdem er die ersten 64 Bytes empfangen hat. Hierdurch wird die Weiterleitung von Frames vermieden, bei denen aufgrund einer Kollision Fehler aufgetreten sind.

IOS-Version und weitere neustartspezifische Informationen

In diesem Abschnitt befassen wir uns mit dem Switch-Befehl **show version**.

Wenn ein Switch das IOS lädt, muss er eine Menge Aufgaben erledigen. Die IOS-Software muss ins RAM geladen werden. Das IOS muss die vorhandene Hardware erkennen (beispielsweise die unterschiedlichen LAN-Interfaces des Switchs). Nach dem Laden der Software muss das IOS einige Statistiken im Zusammenhang mit dem aktuellen Switch-Betrieb erfassen. Hierzu gehören etwa die seit dem letzten Laden des IOS verstrichene Zeit und die Ursache des zuletzt erfolgten IOS-Ladevorgangs.

Mit dem Befehl **show version** können Sie diese und andere Angaben auflisten. Wie Sie dem Befehlsnamen vielleicht schon entnehmen können, zeigt **show version** Informationen zum IOS einschließlich der Version der IOS-Software an. Hinzu kommt eine große Zahl weiterer interessanter Angaben (siehe Listing Q.1).

Listing Q.1 **show version**-Beispielausgabe auf einem Cisco-Switch

```
SW1# show version
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(1)SE3, RELEASE
  SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Wed 30-May-12 14:26 by prod_rel_team

ROM: Bootstrap program is C2960 boot loader
BOOTLDR: C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(44)SE5, RELEASE SOFTWARE (fc1)

SW1 uptime is 2 days, 22 hours, 2 minutes
System returned to ROM by power-on
System image file is "flash:c2960-lanbasek9-mz.150-1.SE3.bin"
```

```
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use…
! Zeilen aus Platzgründen gekürzt


cisco WS-C2960-24TT-L (PowerPC405) processor (revision P0) with 65536K bytes of
   memory.
Processor board ID FCQ1621X6QC
Last reset from power-on
1 Virtual Ethernet interface
24 FastEthernet interfaces
2 Gigabit Ethernet interfaces
The password-recovery mechanism is enabled.

64K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address       : 18:33:9D:7B:13:80
Motherboard assembly number     : 73-11473-11
Power supply part number        : 341-0097-03
Motherboard serial number       : FCQ162103ZL
Power supply serial number      : ALD1619B37W
Model revision number           : P0
Motherboard revision number     : A0
Model number                    : WS-C2960-24TT-L
System serial number            : FCQ1621X6QC
Top Assembly Part Number        : 800-29859-06
Top Assembly Revision Number    : C0
Version ID                      : V10
CLEI Code Number                : COMCX00ARB
Hardware Board Revision Number  : 0x01


Switch Ports Model              SW Version            SW Image
------ ----- -----              ----------            ----------
*    1 26    WS-C2960-24TT-L    15.0(1)SE3            C2960-LANBASEK9-M


Configuration register is 0xF
```

Wie die hervorgehobenen Teile des Listings zeigen, sind der Ausgabe des Befehls von oben nach unten die folgenden Angaben zu entnehmen:

- IOS-Version
- Seit dem letzten Laden des IOS verstrichene Zeit
- Grund für den letzten IOS-Ladevorgang
- Anzahl der Fast Ethernet-Interfaces (24)
- Anzahl der Gigabit Ethernet-Interfaces (2)
- Switch-Modellnummer

HINWEIS Der Inhalt unter der Überschrift »Sekundäre IP-Adressierung« wurde 2013 für die Prüfung 100-101 in Kapitel 16 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100-101* veröffentlicht.

Sekundäre IP-Adressierung

Die meisten Netzwerke arbeiten heute entweder mit Routern mit VLAN-Trunks oder Layer-3-Switches. Dieses nächste Thema widmet sich einem interessanten, aber offen gesagt selten eingesetzten Feature, das dabei hilft, bestimmte zunehmende Probleme mit einem IP-Netzwerk zu beheben.

Stellen Sie sich vor, Sie hätten bereits ein IP-Adressierungsschema für ein Netzwerk geplant. Im Nachhinein wird nun ein bestimmtes Subnetz größer, aber Sie haben bereits alle gültigen IP-Adressen in diesem Subnetz vergeben. Was sollen Sie tun? Es gibt grundsätzlich drei Möglichkeiten:

- Machen Sie das vorhandene Subnetz größer, indem Sie eine Maske mit mehr Hostbits wählen. Vorhandene Hosts müssen die Einstellungen für ihre Subnetzmasken ändern und neue Hosts können IP-Adressen aus dem erweiterten Adressbereich nutzen.
- Migrieren Sie zu einem komplett neuen (größeren) Subnetz. Alle vorhandenen Geräte ändern ihre IP-Adresse.
- Fügen Sie mit sekundärer Adressierung am gleichen Standort ein zweites Subnetz hinzu.

Die ersten Optionen funktionieren gut, solange das neue Subnetz sich nicht mit vorhandenen Subnetzen überschneidet. Wenn beim Design z. B. mit 172.16.2.0/24 gearbeitet wird und hier die Adressen ausgehen, könnte der Techniker stattdessen die Maske /23 verwenden. So wird das Subnetz 172.16.2.0/23 mit dem Adressbereich von 172.16.2.1 bis 172.16.3.254 erstellt. Falls das Subnetz 172.16.3.0/24 jedoch bereits einem anderen Teil des Netzwerks zugewiesen worden ist, gäbe es im Adressierungsplan keinen Raum mehr, um das vorhandene Subnetz zu vergrößern.

Die zweite Option wird wahrscheinlich eher funktionieren: Der Techniker schaut sich die nicht verwendeten IP-Adressen in diesem IP-Netzwerk an und wählt ein neues Subnetz. Es müssten jedoch alle vorhandenen IP-Adressen geändert werden. Dies ist ein vergleichsweise einfacher Vorgang, wenn die meisten oder alle Hosts DHCP benutzen; setzen allerdings viele Hosts statisch konfigurierte IP-Adressen ein, sieht die Geschichte schon etwas anders aus.

Die dritte Option verwendet eine Funktionalität auf Cisco-Routern, die als *sekundäre IP-Adressierung* bezeichnet wird. Die sekundäre Adressierung verwendet mehrere Netzwerke oder Subnetze über dieselbe Datenleitung. (Dieses Feature bricht die Subnetting-Regeln, die in diesem Buch bereits diskutiert wurden, funktioniert aber.) Durch Verwendung mehrerer Subnetze in der gleichen Layer-2-Broadcast-Domäne erhöhen Sie die Anzahl der verfügbaren IP-Adressen.

Abbildung Q.1 zeigt das der sekundären Adressierung zugrunde liegende Konzept. Host A und B befinden sich im gleichen LAN, tatsächlich sogar im gleichen VLAN. Für R1 gilt das auch. Es muss auch kein Trunking vorgenommen werden. Wenn Sie die Zahlen ignorieren würden, dann gehörten A, B und R1 normalerweise alle zum gleichen Subnetz.

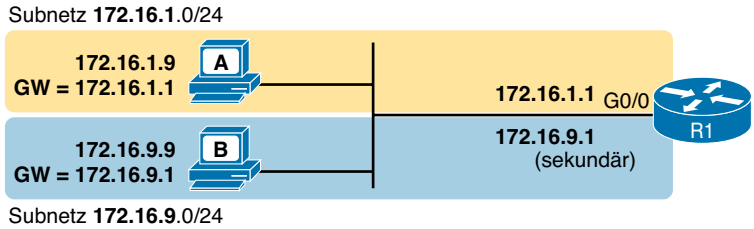

Abbildung Q.1 TCP/IP-Netzwerk mit sekundären Adressen

Durch die sekundäre Adressierung können bestimmte Hosts Adressen in einem IP-Subnetz haben, andere in einem zweiten IP-Subnetz und der Router hat Adressen in beiden. Beide IP-Subnetze befinden sich dann in der gleichen Layer-2-Broadcast-Domäne (VLAN). Als Ergebnis wird der Router über direkt verbundene Routen für beide Subnetze verfügen. Also kann er Pakete an beide Subnetze weiterleiten und sogar zwischen beiden Subnetzen.

Listing Q.2 zeigt die Konfiguration auf R1 entsprechend Abbildung Q.1. Beachten Sie, dass der zweite **ip address** -Befehl das Schlüsselwort **secondary** enthalten muss, mit dem die sekundäre Adressierung implementiert wird. Dem entnimmt der Router dann, dass dies als zusätzlich IP-Adresse aufgenommen werden soll. Ohne dieses Schlüsselwort würde der Router die andere IP-Adresse ersetzen.

Listing Q.2 Konfiguration der sekundären IP-Adressierung und Ausgabe des Befehls **show ip route**

```
! Es folgt ein Ausschnitt aus der show running-config …
interface gigabitethernet 0/0
 ip address 172.16.9.1  255.255.255.0 secondary
 ip address 172.16.1.1  255.255.255.0

R1# show ip route connected
! Zeilen aus Platzgründen gekürzt
      172.16.0.0/16 is variably subnetted, 8 subnets, 2 masks
C        172.16.1.0/24 is directly connected, GigabitEthernet0/0
L        172.16.1.1/32 is directly connected, GigabitEthernet0/0
C        172.16.9.0/24 is directly connected, GigabitEthernet0/0
L        172.16.9.1/32 is directly connected, GigabitEthernet0/0
```

Die sekundäre Adressierung hat aber noch einen Nachteil: Ein Verkehr zwischen Hosts im gleichen VLAN, aber unterschiedlichen Subnetzen, muss den Router durchlaufen. Wenn wie in Abbildung Q.1 Host A in Subnetz 172.16.1.0 ein Paket zu Host B in Subnetz 172.16.9.0 schickt, ist die Logik von Host A, das Paket zu seinem Standard-Gateway zu senden. Also schickt der sendende Host das Paket an den Router, der wiederum das Paket an Host B (befindet sich im anderen IP-Subnetz, aber im gleichen Layer-2-VLAN) weiterleitet.

HINWEIS Der Inhalt unter der Überschrift »Interne Verarbeitung auf Cisco-Routern« wurde 2013 für die Prüfung 100-101 in Kapitel 16 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100-101* veröffentlicht.

Interne Verarbeitung auf Cisco-Routern

Um auf dem Router-Markt konkurrenzfähig zu bleiben, muss Cisco dafür sorgen, dass seine Router den Routing-Prozess sehr gut und schnell in allen möglichen Umgebungen durchführen können. Wäre das nicht der Fall, könnte die Konkurrenz auf die besseren Leistungen ihrer Router verweisen, die mehr Pakete pro Sekunde (packets per second [pps]) weiterleiten und Cisco Marktanteile abgraben.

Als nächstes Thema beschäftigen wir uns eingehender damit, wie Cisco in einem Router das IP-Routing tatsächlich implementiert. Die Ausführungen in diesem Kapitel waren bisher eher allgemein gehalten, aber entsprechen einem frühen Typ der internen Verarbeitung auf Cisco-Routern namens *Process Switching*. In diesem Abschnitt werden die Probleme diskutiert, die Cisco dazu veranlasst haben, den internen Routing-Prozess zu verbessern und zum gleichen Resultat zu kommen: Ein Paket in einem Frame trifft ein, eine Entscheidung wird getroffen und das Paket verlässt den Router in einem anderen Frame.

Potenzielle Performanceprobleme beim Routing

Wenn man sich mit dem IP-Routing beschäftigt, ist es hilfreich, alle Details des Routing-Prozesses zu durchdenken. Allerdings wenden Router kaum Verarbeitungszeit auf, um ein einzelnes IP-Paket weiterzuleiten. Tatsächlich müssen auch langsamere Router pro Sekunde mehrere Zehntausende Pakete weiterleiten. Darum können die Router kaum viel Zeit in die Verarbeitung eines jeden Pakets stecken.

Der Prozess, die Zieladresse eines Pakets aus der IP-Routing-Tabelle herauszufinden, kann ggf. viel CPU-Zeit beanspruchen. Unternehmensnetzwerke haben Tausende von IP-Routen, während im Kern des Internets agierende Router über Hunderttausende von Routen verfügen. Nun stellen Sie sich eine Router-CPU vor, die für jedes Paket eine 100.000 Einträge lange Liste durchsuchen soll, und das bei einem Router, der Hunderttausende Pakete pro Sekunde weiterzuleiten hat! Und wie wäre es, wenn der Router dann noch Subnetzberechnungen anstellen und dafür den Adressbereich für jedes Subnetz und jede Route berechnen müsste? Diese Aktionen bräuchten zu viele CPU-Zyklen.

Im Laufe der Jahre hat Cisco mehrere Wege entwickelt, wie der interne Prozess der Paketweiterleitung optimiert werden kann. Manche Methoden sind mit speziellen Router-Modellserien verknüpft. Layer-3-Switches erledigen die Weiterleitung in Application Specific Integrated Circuits (ASIC). Dabei handelt es sich um Computerchips, die extra für die Weiterleitung von Frames bzw. Paketen gebaut wurden. All diese Optimierungen entnehmen ihre Basislogik aus der erwähnten Liste der fünf Schritte, arbeiten aber in der Hard- und Software des Routers unterschiedlich, um möglichst wenige CPU-Zyklen zu nutzen und den Overhead der IP-Paketweiterleitung zu reduzieren.

Cisco Router Fast Switching und CEF

Historisch gesehen gibt es bei Cisco drei wesentliche Varianten der internen Routing-Logik, die produktübergreifend eingesetzt wird. Zuerst arbeiteten Cisco-Router in der Anfangszeit der Cisco-Router, also etwa Ende der 1980er-Jahre bis Anfang der 1990er-Jahre, mit einer internen Logik namens *Process Switching*. Das Process Switching funktioniert im Grunde so wie der Routing-Prozess ohne zusätzliche Optimierungen.

Anfang der 1990er-Jahre führte Cisco eine alternative Routing-Logik namens *Fast Switching* ein. Beim Fast Switching kamen verglichen mit dem älteren Process Switching verschiedene Optimierungen zum Einsatz. Erstens wurde dabei eine weitere Liste zusätzlich zur Routing-Tabelle genutzt, in der spezielle IP-Adressen für kürzlich weitergeleitete Pakete geführt wurden. Dieser Fast-Switching-Cache enthielt auch eine Kopie der neuen Data-Link-Header, die für die Paketweiterleitung an jedes Ziel benutzt wurden. Anstatt also für jedes Paket, das an eine bestimmte IP-Adresse geschickt werden soll, jedes Mal einen neuen Data-Link-Header zu erstellen, sparte sich der Router die Mühe und kopierte einfach den alten Header.

Cisco verbesserte das Fast Switching später in den 1990er-Jahren durch Einführung von Cisco Express Forwarding (CEF). Wie beim Fast Switching nutzt auch CEF zusätzliche Tabellen für schnellere Suchen und speichert ausgehende Data-Link-Header. Allerdings organisiert CEF seine Tabellen für alle Routing-Ziele schon vorab, nicht nur für einige der speziellen Empfänger-IP-Adressen. CEF arbeitet verglichen mit Fast Switching auch mit deutlich ausgefeilteren Suchalgorithmen und binären Baumstrukturen. Als Folge davon benötigen CEF-Tabellen-Lookups, die die Routing-Tabellen ersetzen, weitaus weniger Zeit als mit Fast Switching. Und auch die Data-Link-Header sind darin gecacht.

Heutzutage arbeiten aktuelle Cisco-Router und IOS-Versionen standardmäßig mit CEF. In Tabelle Q.2 werden die wesentlichen Vergleichspunkte zwischen Process Switching, Fast Switching und CEF aufgeführt.

Tabelle Q.2 Vergleich von Process Switching, Fast Switching und CEF

Verbessert die Routing-Effizienz durch ...	Process Switching	Fast Switching	CEF
... Speichern der Data-Link-Header, die zur Kapselung von Paketen benutzt werden	Nein	Ja	Ja
... Verwendung anderer Tabellen mit schnellerer Zugriffszeit, bevor die Routing-Tabelle herangezogen wird	Nein	Ja	Ja
... Organisation der Tabellen in Baumstrukturen für sehr schnelles Suchen und weniger Zeit für die Weiterleitung von Paketen	Nein	Nein	Ja

HINWEIS Der Inhalt unter der Überschrift »OSPF-Konfiguration« wurde 2013 für die Prüfung 100-101 in Kapitel 17 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100-101* veröffentlicht. Der folgende kurze Abschnitt legt den Schwerpunkt auf die Hostnamensauflösung.

OSPF-Konfiguration

Bei der OSPF-Konfiguration sind nur wenige Schritte zwingend, es gibt allerdings eine ganze Reihe optionaler Maßnahmen. Nach Auswahl eines OSPF-Designs – einer Aufgabe, die in größeren IP-Netzwerkverbunden recht komplex sein kann – kann die Konfiguration unter Umständen sogar lediglich darin bestehen, OSPF auf allen Router-Interfaces zu aktivieren und diese dann in der jeweils korrekten OSPF-Area zu platzieren.

In diesem Abschnitt zeigen wir diverse Konfigurationsbeispiele. Wir beginnen mit einem aus nur einer Area bestehenden OSPF-Netzwerkverbund. Auf diese Beispiele folgend werden wir verschiedene weitere optionale Konfigurationseinstellungen behandeln. Der Übersicht halber führt die folgende Liste die in diesem Anhang beschriebenen Konfigurationsschritte auf und nennt die erforderlichen Befehle:

Schritt 1: Wechseln Sie mit dem globalen Befehl **router ospf** *prozess-id* in den OSPF-Konfigurationsmodus für einen bestimmten OSPF-Prozess.

Schritt 2: Konfigurieren Sie die OSPF-Router-ID (optional):

- **a.** Konfigurieren Sie den Router-Subbefehl **router-id***id-wert*.
- **b.** Konfigurieren Sie eine IP-Adresse auf einem Loopback-Interface.

Schritt 3: Konfigurieren Sie einen oder mehrere Router-Subbefehle **network** *ip-adresse wildcard-maske* **area** *area-id*. Hierbei werden alle passenden Interfaces zur angegebenen Area hinzugefügt.

Für eine bessere Veranschaulichung der OSPFv2-Konfiguration sehen Sie sich in Abbildung Q.2 die Beziehung zwischen den wesentlichen OSPF-Konfigurationsbefehlen an. Beachten Sie, dass die Konfiguration einen Routing-Prozess in einem Teil der Konfiguration erstellt und dann indirekt OSPF auf jedem Interface aktiviert. Die Konfiguration benennt die Interfaces nicht, auf denen OSPF aktiviert ist, und lässt das IOS dann etwas Logik anwenden, indem der OSPF-Befehl **network** mit den Interface-Befehlen **ip address** verglichen wird. Das folgende Beispiel erläutert diese Logik eingehender.

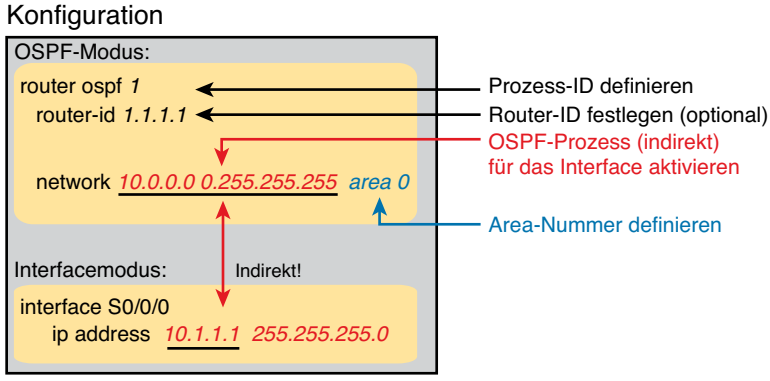

Abbildung Q.2 Organisation der OSPFv2-Konfiguration

OSPF für eine Area konfigurieren

Abbildung Q.3 zeigt ein Netzwerkbeispiel, das für die OSPF-Konfiguration verwendet werden soll. Alle Verbindungen befinden sich in Area 0. Darin sind vier Router, die alle mit einem oder zwei LANs verbunden sind. Beachten Sie allerdings, dass die beiden Router R3 und R4 oben in der Abbildung mit den gleichen beiden VLANs/Subnetzen verbunden sind. Sie haben also über die jeweiligen VLANs auch nachbarschaftliche Beziehungen zueinander.

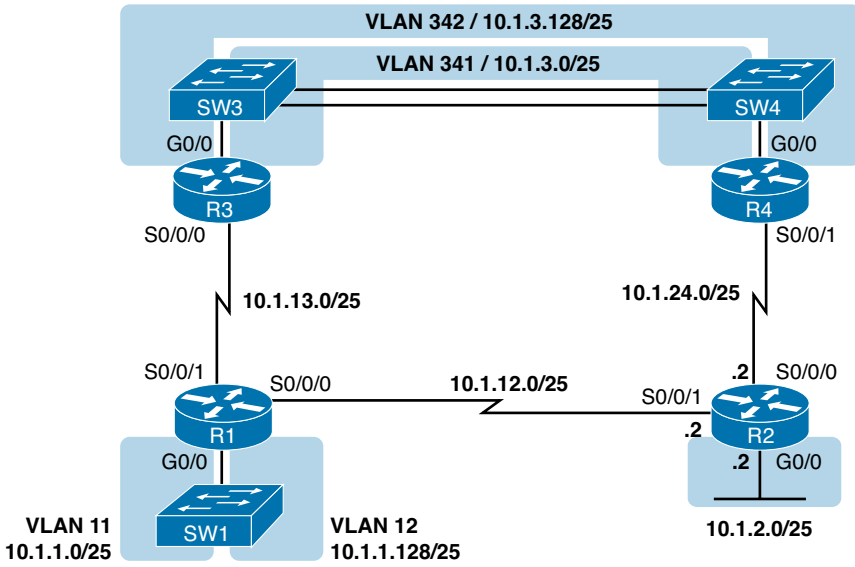

Abbildung Q.3 Beispielnetzwerk für die OSPF-Konfiguration bei nur einer Area

Listing Q.3 zeigt die IP-Adresskonfiguration auf Router R3, bevor es in die OSPF-Details geht. Die Konfiguration aktiviert 802.1Q-Trunking auf dem G0/0-Interface von R3 und weist jedem Subinterface eine IP-Adresse zu. (Nicht gezeigt wird, dass auf dem Switch SW3 auf der anderen Seite dieser Ethernet-Verbindung Trunking konfiguriert wurde.)

Listing Q.3 IPv4-Adresskonfiguration auf R3 (inklusive VLAN-Trunking)

```
interface gigabitethernet 0/0.341
 encapsulation dot1q 341
 ip address 10.1.3.1 255.255.255.128
!
interface gigabitethernet 0/0.342
 encapsulation dot1q 342
 ip address 10.1.3.129 255.255.255.128
!
interface serial 0/0/0
 ip address 10.1.13.3 255.255.255.128
```

Die anfängliche Single-Area-Konfiguration auf R3 (siehe Listing Q.4) aktiviert OSPF auf allen in Abbildung Q.3 gezeigten Interfaces. Zuerst versetzt der globale Befehl **router ospf 1** den

Benutzer in den OSPF-Konfigurationsmodus und setzt die OSPF-*process-id*. Diese Zahl muss nur auf dem lokalen Router einmalig sein und erlaubt dem Router durch unterschiedliche Prozess-IDs, mehrere OSPF-Prozesse auf einem einzigen Router zu unterstützen. (Der Befehl **router** verwendet die *Prozess-ID*, um die Prozesse zu unterscheiden.) Die *Prozess-ID* kann einen Wert zwischen 1 und 65.535 annehmen und muss nicht auf jedem Router gleich sein.

Listing Q.4 OSPF-Konfiguration auf R3 für eine Area mit einem network-Befehl

```
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
```

Unabhängig von diesem Beispiel formuliert, weist der OSPF-Befehl **network** einen Router an, die lokalen Interfaces zu finden, die zu den ersten beiden Parametern des Befehls **network** passen. Dann aktiviert der Router für jedes passende Interface OSPF, entdeckt Nachbarn, erstellt Nachbarschaftsbeziehungen und weist dem Interface die Area zu, die im Befehl **network** aufgeführt ist.

Für den spezifischen Befehl in Listing Q.4 werden alle passenden Interfaces der Area 0 zugewiesen. Allerdings sollten die ersten beiden Parameter – *ip-adresse* und *wildcard-maske* mit den Werten 10.0.0.0 und 0.255.255.255 – näher erläutert werden. In diesem Fall trifft der Befehl auf alle drei für R3 gezeigte Interfaces zu; beim nächsten Thema wird dies genauer erklärt.

Der OSPF-Befehl **network**

Dieser Befehl vergleicht den ersten Parameter im Befehl mit jeder IP-Interface-Adresse auf dem lokalen Router und versucht, eine Übereinstimmung zu finden. Doch der Router vergleicht nicht die gesamte Zahl im **network**-Befehl mit der gesamten IPv4-Adresse des Interface, sondern er kann basierend auf der Wildcard-Maske eine Teilmenge der Oktette vergleichen:

- **Wildcard 0.0.0.0:** Vergleiche alle vier Oktette. Anders gesagt, die Zahlen müssen exakt übereinstimmen.
- **Wildcard 0.0.0.255:** Vergleiche nur die drei ersten Oktette. Ignoriere dabei das letzte Oktett.
- **Wildcard 0.0.255.255:** Vergleiche nur die beiden ersten Oktette. Ignoriere dabei die letzten beiden Oktette.
- **Wildcard 0.255.255.255:** Vergleiche nur das erste Oktett. Ignoriere dabei die letzten drei Oktette.
- **Wildcard 255.255.255.255:** Vergleiche nichts – diese Wildcard-Maske bedeutet, dass alle Adressen zum Befehl **network** passen.

Im Wesentlichen weist der Wert 0 bei der Wildcard-Maske das IOS an, die Zahlen auf Übereinstimmung zu vergleichen, und dem Wert 255 entnimmt das IOS, dieses Oktett beim Vergleich zu ignorieren.

Aufgrund der Wildcard-Maske bietet der Befehl **network** viele flexible Optionen. Bei Router R3 könnten beispielsweise viele **network**-Befehle verwendet werden, wobei manche auf alle Interfaces zutreffen und andere auf eine Teilmenge der Interfaces. Tabelle Q.3 zeigt verschiedene Beispieloptionen mit Anmerkungen.

Tabelle Q.3 Beispiele für OSPF-**network**-Befehle für R3 mit zu erwartenden Ergebnissen

Befehl	Logik im Befehl	Übereinstimmende Interfaces
network 10.1.0.0 0.0.255.255	Suche passende IP-Interface-Adressen, die mit 10.1 beginnen	G0/0.341 G0/0.342 S0/0/0
network 10.0.0.0 0.255.255.255	Suche passende IP-Interface-Adressen, die mit 10 beginnen	G0/0.341 G0/0.342 S0/0/0
network 0.0.0.0 255.255.255.255	Suche alle passenden IP-Interface-Adressen	G0/0.341 G0/0.342 S0/0/0
network 10.1.13.0 0.0.0.255	Suche passende IP-Interface-Adressen, die mit 10.1.13 beginnen	S0/0/0
network 10.1.3.1 0.0.0.0	Suche eine passende IP-Adresse: 10.1.3.1	G0/0.341

Aus der Wildcard-Maske entnimmt der lokale Router seine Regeln, um entsprechend eigene Interfaces zu suchen. Listing Q.4 zeigt etwa, wie R3 den Befehl **network 10.0.0.0 0.255.255.255 area 0** verwendet. Im gleichen Internetwork würden die Router R1 und R2 die in Listing Q.5 gezeigte Konfiguration mit zwei anderen Wildcard-Masken verwenden. Auf beiden Routern ist für alle in Abbildung Q.3 gezeigten Interfaces OSPF aktiviert.

Listing Q.5 OSPF-Konfigurationen der Router R1 und R2

```
! R1-Konfiguration: Ein network-Befehl aktiviert OSPF
! für alle drei Interfaces
router ospf 1
 network 10.1.0.0 0.0.255.255 area 0
```

```
! R2-Konfiguration: Ein network-Befehl je Interface
router ospf 1
 network 10.1.12.2 0.0.0.0 area 0
 network 10.1.24.2 0.0.0.0 area 0
 network 10.1.2.2 0.0.0.0 area 0
```

Beachten Sie zum Schluss, dass für die Wildcard-Masken auch andere Werte verwendet werden können, sodass der Vergleich mit spezifischen Bits in den 32-Bit-Zahlen stattfindet. In Kapitel 25, »Einfache IPv4-ACLs«, werden Wildcard-Masken ausführlicher besprochen, darunter auch diese anderen Maskenoptionen.

HINWEIS Der **network**-Befehl verwendet eine andere Konvention für den ersten Parameter (die Adresse): Falls ein Oktett wegen des Oktettwerts 255 für die Wildcard-Maske ignoriert wird, sollte der Adressparameter 0 lauten. Das IOS wird dann doch einen **network**-Befehl akzeptieren, der sich nicht an diese Regel hält, ändert aber das Oktett dieser Adresse auf 0, bevor es in eine laufende Konfigurationsdatei eingefügt wird. Das IOS wird beispielsweise einen eingetippten Befehl, der mit **network 1.2.3.4 0.0.255.255** beginnt, auf **network 1.2.0.0 0.0.255.255** abändern.

OSPF überprüfen

Wie bereits erwähnt, arbeiten OSPF-Router in einem dreistufigen Prozess. Zuerst erstellen sie Nachbarschaftsbeziehungen. Dann erstellen und fluten sie LSAs, sodass jeder Router in der gleichen Area eine Kopie der LSDB besitzt. Schließlich berechnet jeder Router unabhängig seine eigenen IP-Routen und fügt sie in seine Routing-Tabelle ein.

Die Befehle **show ip ospf neighbor**, **show ip ospf database** und **show ip route** stellen jeweils Informationen für jeden dieser drei Schritte dar. Um OSPF zu verifizieren, können Sie nach der gleichen Sequenz arbeiten. Oder Sie werfen einen Blick auf die IP-Routing-Tabelle und wenn die Routen korrekt aussehen, hat OSPF wahrscheinlich funktioniert.

Zuerst untersuchen Sie die Liste der Nachbarn, die für Router R3 bekannt sind. R3 sollte eine Nachbarschaftsbeziehung über die serielle Verbindung zu R1 haben. Er hat auch zwei Nachbarschaftsbeziehungen mit R4 über die beiden verschiedenen VLANs, mit denen beide Router verbunden sind. Listing Q.6 zeigt alle drei.

Listing Q.6 OSPF-Nachbarn auf Router R3 aus Abbildung Q.3

```
R3# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           0   FULL/  -        00:00:33    10.1.13.1       Serial0/0/0
10.1.24.4         1   FULL/DR         00:00:35    10.1.3.130      GigabitEthernet0/0.342
10.1.24.4         1   FULL/DR         00:00:36    10.1.3.4        GigabitEthernet0/0.341
```

Die Details in der Ausgabe liefern mehrere wichtige Fakten und für die meisten Leute funktioniert es am besten, von rechts nach links zu lesen. Betrachten Sie beispielsweise die Überschriften.

- **Interface:** Dies ist das Interface des lokalen Routers, der mit dem Nachbarn verbunden ist. Der erste Nachbar in der Liste ist z. B. über das R3-Interface S0/0/0 erreichbar.
- **Address:** Dies ist die IP-Adresse des Nachbarn in dieser Verbindung. Noch einmal: Für diesen ersten Nachbarn verwendet der Nachbar (R1) die IP-Adresse 10.1.13.1.
- **State:** Es gibt zwar viele mögliche Zustände, aber für die in diesem Kapitel diskutierten Details ist FULL in diesem Fall der korrekte und voll funktionsfähige Zustand.
- **Neighbor ID:** Dies ist die Router-ID des Nachbarn.

Als Nächstes zeigt Listing Q.7 die Inhalte der LSDB auf Router R3. Interessanterweise werden alle Router die gleichen LSDB-Inhalte haben, wenn OSPF in einem Internetwork mit einem

Single-Area-Design korrekt funktioniert. Also sollte der Befehl **show ip ospf database** in Listing Q.7 die exakt gleichen Informationen auflisten, egal, auf welchem der vier Router er aufgerufen wird.

Listing Q.7 OSPF-Datenbank auf Router R3 aus Abbildung Q.3

```
R3# show ip ospf database

            OSPF Router with ID (10.1.13.3) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         498         0x80000006 0x002294 6
2.2.2.2         2.2.2.2         497         0x80000004 0x00E8C6 5
10.1.13.3       10.1.13.3       450         0x80000003 0x001043 4
10.1.24.4       10.1.24.4       451         0x80000003 0x009D7E 4

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.3.4        10.1.24.4       451         0x80000001 0x0045F8
10.1.3.130      10.1.24.4       451         0x80000001 0x00546B
```

Für die Zwecke dieses Buchs brauchen Sie sich nicht näher mit den Einzelheiten in der Ausgabe dieses Befehls zu beschäftigen. Doch für den Zusammenhang sollten Sie beachten, dass die LSDB einen »Router Link State« (Typ-1-Router LSA) für jeden der vier Router in diesem Design auflistet, wie es im Beispiel gekennzeichnet ist.

Als Nächstes zeigt Listing Q.8 über den Befehl **show ip route** die IPv4-Routing-Tabelle von R3. Beachten Sie, dass damit sowohl die direkt verbundenen als auch die OSPF-Routen aufgelistet werden. Nehmen Sie sich einen Moment Zeit, sich noch einmal die Abbildung Q.3 anzuschauen und nach den Subnetzen zu suchen, die nicht lokal mit R3 verbunden sind. Dann suchen Sie diese Routen in der Ausgabe von Listing Q.7.

Listing Q.8 Über OSPF hinzugefügte IPv4-Routen auf Router R3 aus Abbildung Q.3

```
R3# show ip route
Codes: L - local, C - connected, S - static, 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
! ! Zeilen aus Platzgründen gekürzt


     10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
O        10.1.1.0/25 [110/65] via 10.1.13.1, 00:13:28, Serial0/0/0
O        10.1.1.128/25 [110/65] via 10.1.13.1, 00:13:28, Serial0/0/0
O        10.1.2.0/25 [110/66] via 10.1.3.130, 00:12:41, GigabitEthernet0/0.342
                     [110/66] via 10.1.3.4, 00:12:41, GigabitEthernet0/0.341
```

```
C        10.1.3.0/25 is directly connected, GigabitEthernet0/0.341
L        10.1.3.1/32 is directly connected, GigabitEthernet0/0.341
C        10.1.3.128/25 is directly connected, GigabitEthernet0/0.342
L        10.1.3.129/32 is directly connected, GigabitEthernet0/0.342
O        10.1.12.0/25 [110/128] via 10.1.13.1, 00:13:28, Serial0/0/0
C        10.1.13.0/25 is directly connected, Serial0/0/0
L        10.1.13.3/32 is directly connected, Serial0/0/0
O        10.1.24.0/25
           [110/65] via 10.1.3.130, 00:12:41, GigabitEthernet0/0.342
           [110/65] via 10.1.3.4, 00:12:41, GigabitEthernet0/0.341
```

Zuerst wollen wir uns die umfassendere Konzepte einmal anschauen, die durch diese Ausgabe bestätigt werden. Der Code »O« identifiziert eine Route, die über OSPF erlernt wurde. In der Ausgabe werden fünf solcher IP-Routen aufgeführt. Aus der Abbildung lässt sich entnehmen, dass es fünf Subnetze gibt, bei denen es sich nun nicht um direkt verbundene Subnetze außerhalb von Router R3 handelt. Wenn man im Diagramm die Zahl der OSPF-Routen kurz mit den nicht verbundenen Routen abgleicht, erkennt man, ob OSPF alle Routen gelernt hat.

Als Nächstes sehen Sie sich die erste Route (ins Subnetz 10.1.1.0/25) an. Hier werden Subnetz-ID und Maske aufgelistet, die das Subnetz identifizieren. Außerdem stehen da noch zwei Zahlen in Klammern. Die erste, 110, ist die administrative Distanz der Route. Alle OSPF-Routen in diesem Beispiel arbeiten mit dem Standard 110. Die zweite Zahl (65) ist die OSPF-Metrik für diese Route.

Außerdem wird auch gerne der Befehl **show ip protocols** genommen, um mal schnell zu prüfen, wie ein beliebiges Routing-Protokoll funktioniert. Dieser Befehl listet eine Gruppe von Nachrichten für jedes Routing-Protokoll auf, das auf einem Router läuft. Listing Q.9 zeigt ein Beispiel, dieses Mal aus Router R3.

Listing Q.9 Der Befehl **show ip protocols** auf R3

```
R3# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.13.3
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    1.1.1.1              110      06:26:17
    2.2.2.2              110      06:25:30
    10.1.24.4            110      06:25:30
  Distance: (default is 110)
```

Aus der Ausgabe gehen mehrere interessante Fakten hervor. Die erste hervorgehobene Zeile wiederholt den globalen Konfigurationsbefehl **router ospf 1**. Das zweite markierte Element zeigt die Router-ID von R3 und wird eingehender im nächsten Abschnitt diskutiert. Die dritte markierte Zeile wiederholt weitere Konfigurationen und listet die Parameter des OSPF-Subbefehls **network 10.0.0.0 0.255.255.255 area 0** auf. Schließlich agiert das letzte hervorgehobene Element als Überschrift vor einer Liste mit nach Router-ID sortierten bekannten OSPF-Routern.

OSPF-Router-ID konfigurieren

Zwar hat OSPF viele andere optionale Funktionen, doch die meisten Unternehmensnetzwerke, die mit OSPF arbeiten, entscheiden sich dafür, die OSPF-Router-ID aller Router zu konfigurieren. Mit OSPF arbeitende Router benötigen für den korrekten Betrieb eine Router-ID (RID). Standardmäßig werden Router eine IP-Interface-Adresse wählen, die als RID verwendet werden soll. Doch viele Netzwerktechniker ziehen für Router die Router-ID vor, damit die Ausgabe eines Befehls wie **show ip ospf neighbor** leichter erkennbare Router-IDs auflistet.

Um seine RID zu ermitteln, verwendet ein Cisco-Router den folgenden Vorgang, wenn er neu lädt und den OSPF-Prozess startet. Beachten Sie, dass der Prozess endet, sobald einer dieser Schritte die RID identifiziert.

1. Wenn der OSPF-Befehl **router-id** *rid* konfiguriert wird, wird dessen Wert als RID benutzt.
2. Wenn eine oder mehrere Loopback-Interfaces IP-Adressen aufweisen und das Interface den Protokollstatus *up* vorweisen kann, wählt der Router die höchste numerische IP-Adresse unter den Loopback-Interfaces.
3. Der Router wählt die höchste numerische IP-Adresse unter allen anderen betriebsbereiten Interfaces (d. h. solchen mit dem Status *up*) aus. (Anders formuliert wird ein Interface mit dem Status *up/down* durch OSPF eingebunden, wenn dessen Router-ID gewählt wird.)

Das erste und dritte Kriterium sollte direkt nachvollziehbar sein: Die RID wird entweder konfiguriert oder aus der funktionierenden IP-Adresse eines Interface entnommen. Allerdings haben wir in diesem Buch noch nicht das Konzept des in Schritt 2 erwähnten *Loopback-Interface* erläutert.

Ein Loopback-Interface ist ein virtuelles Interface, das mit dem Befehl **interface loopback** *interfacenummer* konfiguriert werden kann. Hierbei ist *interfacenummer* eine ganze Zahl. Loopback-Interfaces haben stets den Status *up/up*, sofern sie nicht administratorseitig in den Status *shutdown* versetzt wurden. Beispielsweise würde ein einfacher Befehl **interface loopback 0**, gefolgt von **ip address 2.2.2.2 255.255.255.0**, ein Loopback-Interface erstellen und ihm eine IP-Adresse zuweisen. Da Loopback-Interfaces auf keine Hardware angewiesen sind, können sie immer *up/up* sein, wenn das IOS ausgeführt wird. Dies macht sie zu geeigneten Kandidaten für OSPF-RIDs.

Listing Q.10 zeigt die Konfiguration, die auf den Routern R1 und R2 existierte, bevor die Ausgabe des Befehls **show** in den Listings Q.6, Q.7 und Q.8 erstellt wurde. R1 hat seine Router-ID anhand der direkten Methode festgelegt, während R2 eine Loopback-IP-Adresse genommen hat.

Listing Q.10 Konfigurationsbeispiele für OSPF-Router-IDs

```
! R1 Configuration first
router ospf 1
 router-id 1.1.1.1
 network 10.1.0.0 0.0.255.255 area 0
 network 10.0.0.0 0.255.255.255 area 0
! R2 Configuration next
!
interface Loopback2
 ip address 2.2.2.2 255.255.255.255
```

Jeder Router wählt seine OSPF-RID, wenn OSPF initialisiert wird. Das passiert, wenn der Router startet oder wenn der Benutzer via CLI (Kommandozeile) den OSPF-Prozess stoppt und erneut startet (über den Befehl **clear ip ospf process**). Wenn also OSPF startet und sich später die Konfiguration auf eine Weise ändert, die sich auf die OSPF-RID auswirkt, ändert OSPF die RID nicht sofort. Stattdessen wartet das IOS, bis der OSPF-Prozess das nächste Mal neu gestartet wird.

Listing Q.11 zeigt die Ausgabe des Befehls **show ip ospf** für R1, nachdem die Konfiguration von Listing Q.10 vorgenommen und der Router erneut gestartet wurde, wodurch die Router-ID-Änderung durch OSPF ausgelöst wurde.

Listing Q.11 Die aktuelle OSPF-Router-ID bestätigen

```
R1# show ip ospf
 Routing Process "ospf 1" with ID 1.1.1.1
! Zeilen aus Platzgründen gekürzt
```

Weitere OSPF-Konfigurationseinstellungen

Die letzten Themen dieses Kapitels behandeln einige optionale OSPF-Konfigurationseinstellungen, die hiermit nichts zu tun haben, insbesondere wie man ein Router-Interface für OSPF passiv machen kann und wie man eine Default-Route von OSPF ausgehen lassen und fluten kann.

Passive OSPF-Interfaces

Nachdem OSPF auf einem Interface aktiviert wurde, versucht der Router, benachbarte OSPF-Router zu entdecken und eine Nachbarschaftsbeziehung zu etablieren. Dafür sendet der Router in regelmäßigen Intervallen (Hello-Intervall genannt) OSPF-Hello-Nachrichten. Der Router lauscht ebenfalls auf eingehende Hello-Nachrichten von potenziellen Nachbarn.

Manchmal braucht ein Router keine Nachbarschaftsbeziehungen mit Nachbarn auf einem Interface zu bilden. Oftmals existieren keine anderen Router auf einer bestimmten Verbindung, also muss er auch nicht dauernd seine OSPF-Hello-Nachrichten versenden.

Wenn ein Router keine Nachbarn über ein Interface entdecken muss, verfügt der Netzwerktechniker über ein paar Konfigurationsoptionen. Erstens könnte er nichts machen und der Router versendet diese Nachrichten und verschwendet so ein wenig CPU-Zyklen und Aufwand. Alternativ kann der Techniker das Interface als passives OSPF-Interface konfigurieren, indem er den Router wie folgt anweist:

- Höre auf, OSPF-Hellos über das Interface zu versenden.
- Ignoriere über das Interface empfangene OSPF-Hellos.
- Bilde über das Interface keine Nachbarschaftsbeziehungen.

Indem das Interface passiv gemacht wird, bildet OSPF keine Nachbarschaftsbeziehungen über das Interface, gibt aber über das mit diesem Interface verbundene Subnetz noch weitere bekannt. Das bedeutet, die OSPF-Konfiguration aktiviert OSPF auf dem Interface (anhand des Router-Subbefehls **network**) und macht das Interface dann passiv (vermittels des Router-Subbefehls **passive-interface**).

Um ein Interface als passiv zu konfigurieren, gibt es zwei Optionen. Zuerst können Sie den folgenden Befehl im Router-Konfigurationsmodus in die Konfiguration des OSPF-Prozesses einfügen:

passive-interface *Typnummer*

Alternativ kann die Konfiguration die Standardeinstellung ändern, sodass alle Interfaces standardmäßig passiv sind, und dann wird der Befehl **no passive-interface** für alle Interfaces eingefügt, die nicht passiv sein müssen:

passive-interface default
no passive interface *Typnummer*

Zum Beispiel hat der Router R1 im Beispielnetzwerkverbund in Abbildung Q.3 unten links ein LAN-Interface, das für VLAN-Trunking konfiguriert ist. Der einzige mit beiden VLANs verbundene Router ist R1. Also wird R1 nie einen OSPF-Nachbarn in diesen Subnetzen entdecken. Listing Q.12 zeigt zwei alternative Konfigurationen, um die beiden LAN-Subinterfaces für OSPF passiv zu machen.

Listing Q.12 Passive Interfaces auf R1 und R2 aus Abbildung Q.3 konfigurieren

```
! Erstens jedes Subinterface direkt passiv machen
router ospf 1
 passive-interface gigabitethernet0/0.11
 passive-interface gigabitethernet0/0.12

! Oder Standardeinstellung auf passiv ändern und dann die anderen
! Interfaces nicht passiv machen

router ospf 1
 passive-interface default
 no passive-interface serial0/0/0
 no passive-interface serial0/0/1
```

In realen Netzwerkverbunden reduziert sich die Wahl des Konfigurationsstils darauf, welche Option die wenigsten Befehle erfordert. Ein Router hat beispielsweise 20 Interfaces, von denen für OSPF 18 passiv sind. Dann braucht man deutlich weniger Konfigurationsbefehle, wenn man den Standard mit **passive-interface default** auf passiv setzt. Wenn nur zwei dieser zwanzig Interfaces passiv sein sollen, nehmen Sie die Standardeinstellung, in der alle Interfaces nicht passiv sind, um die Konfiguration kürzer zu halten.

Interessanterweise wird es bei OSPF zu einer kleinen Herausforderung, wenn man anhand der **show**-Befehle herausfinden will, ob ein Interface passiv ist. Der Befehl **show running-config** listet die Konfiguration direkt auf, aber wenn Sie nicht in den Enable-Modus gelangen, um diesen Befehl zu nutzen, beachten Sie diese beiden Fakten:

- Der Befehl **show ip ospf interface brief** listet alle Interfaces auf, bei denen OSPF aktiviert ist, *einschließlich passiver Interfaces*.
- Der Befehl **show ip ospf interface** gibt eine einzelne Zeile aus, in der steht, dass das Interface passiv ist.

Listing Q.13 zeigt diese beiden Befehle auf Router R1 zusammen mit der in Listing Q.12 gezeigten Konfiguration. Beachten Sie, dass die Subinterfaces G0/0.11 und G0/0.12 beide in der Ausgabe von **show ip ospf interface brief** erscheinen.

Listing Q.13 Passive Interfaces darstellen

```
R1# show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Gi0/0.12     1     0               10.1.1.129/25      1     DR    0/0
Gi0/0.11     1     0               10.1.1.1/25        1     DR    0/0
Se0/0/0      1     0               10.1.12.1/25       64    P2P   0/0
Se0/0/1      1     0               10.1.13.1/25       64    P2P   0/0

R1# show ip ospf interface g0/0.11
GigabitEthernet0/0.11 is up, line protocol is up
  Internet Address 10.1.1.1/25, Area 0, Attached via Network Statement
  Process ID 1, Router ID 10.1.1.129, Network Type BROADCAST, Cost: 1
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           1         no          no            Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.1.129, Interface address 10.1.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    No Hellos (Passive interface)
! Zeilen aus Platzgründen gekürzt
```

OSPF-Default-Routen

Wie in Kapitel 18, »IPv4-Adressen und -Routen konfigurieren«, behandelt, profitieren Router manchmal von einer Default-Route. Kapitel 18 zeigt, wie man einen Router so konfiguriert, dass er eine statische Default-Route kennt, die nur dieser eine Router nutzt. In diesem Kapitel beschäftigen wir uns abschließend mit einer anderen Strategie für Standard-IP-Routen, bei der ein OSPF-Router eine Default-Route erstellt und sie ebenfalls per OSPF bekanntmacht, damit andere Router dynamisch die Default-Routen kennenlernen.

Der klassische Fall, um eine Default-Route anhand eines Routing-Protokolls bekanntzugeben, hat damit zu tun, wie ein Unternehmen sich mit dem Internet verbindet. Als Strategie arbeitet der Netzwerktechniker des Unternehmens mit diesen drei Designzielen:

- Alle Router lernen spezifische Routen für Subnetze innerhalb der Firma; eine Default-Route wird nicht benötigt, wenn Pakete an diese Ziele weitergeleitet werden.
- Ein Router baut die Verbindung zum Internet auf und hat eine Default-Route, die zum Internet zeigt.
- Alle Router sollten dynamisch eine Default-Route lernen, die für den gesamten Netzwerkverkehr ins Internet verwendet wird, damit alle Pakete für Ziele im Internet über diesen einen Router gehen.

Abbildung Q.4 zeigt, wie OSPF die Default-Route über die spezifische OSPF-Konfiguration bekanntmacht. In diesem Fall verbindet sich eine Firma mit ihrem ISP über ihren Router R1. Dieser Router arbeitet mit dem Befehl **default-information originate** (Schritt 1). Infolgedessen gibt der Router anhand von OSPF (Schritt 2) den Remote-Routern (B1, B2 und B3) eine Default-Route bekannt.

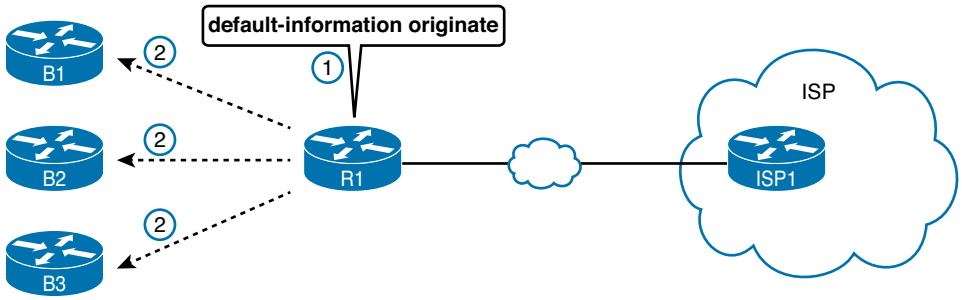

Abbildung Q.4 Default-Route mit OSPF erstellen und fluten

Abbildung Q.5 zeigt die Default-Routen, die sich aus den OSPF-Bekanntmachungen von Abbildung Q.4 ergeben. Ganz links haben alle Zweigstellen-Router ihre Default-Routen über OSPF erfahren und zeigen auf R1. R1 braucht selbst auch eine Default-Route, die auf den ISP-Router zeigt, damit R1 den gesamten, fürs Internet bestimmten Datenverkehr an den ISP weiterleiten kann.

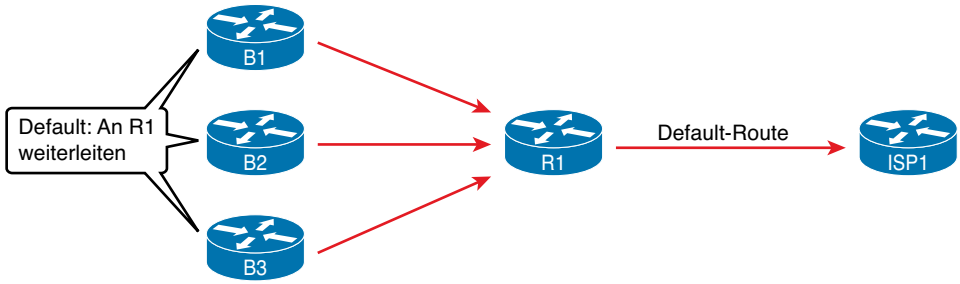

Abbildung Q.5 Default-Routen, die sich aus dem Befehl *default-information originate* ergeben

Schließlich kann der Techniker über dieses Feature steuern, wann der Router diese Default-Route ausgibt. Zuerst braucht R1 eine Default-Route, entweder statisch definiert oder vom ISP gelernt. Der Befehl **default-information originate** weist R1 nun an, eine Default-Route bekanntzugeben, wenn seine eigene Default-Route funktioniert, und sie als *down* anzugeben, wenn seine eigene Default-Route nicht funktioniert.

HINWEIS Interessanterweise sagt der Router-Subbefehl **default-information originate always** dem Router, dass er immer die Default-Route bekanntgeben soll, egal, ob seine eigene Default-Route funktioniert oder nicht.

HINWEIS Der Inhalt unter der Überschrift »Namensauflösung mit DNS« wurde 2013 für die Prüfung 100-101 in Kapitel 18 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100-101* veröffentlicht. Der folgende kurze Abschnitt legt den Schwerpunkt auf die Hostnamensauflösung.

Namensauflösung mit DNS

Betrachten Sie die in Abbildung Q.6 gezeigte Netzwerktopologie. Sie bildet die Grundlage für die nachfolgenden Erläuterungen.

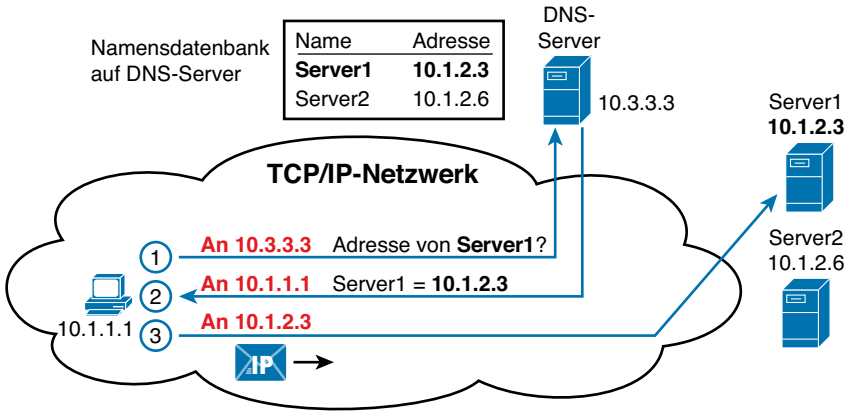

Abbildung Q.6 Ein Host löst den Namen in eine IP-Adresse auf, bevor er das Paket an Server1 sendet.

Wenn Sie sich Probleme mit Hosts anschauen, sollten Sie die DNS-Einstellungen überprüfen, um herauszufinden, welche DNS-Serveradressen der Host verwenden will. Gleichzeitig kann der Benutzer den Host veranlassen, DNS zu verwenden. Ein Beispiel:

- Öffnen Sie einen Browser und tippen Sie den Namen des Webservers ein. DNS löst den Namen auf, der zwischen dem // und dem ersten / steht.
- Verwenden Sie einen Befehl wie **nslookup** *hostname*✎ Der wird von den meisten PC-Betriebssystemen unterstützt und sendet einen DNS-Request an den DNS-Server, der das Ergebnis zeigt.

Listing Q.14 zeigt den Befehl **nslookup**, der bestätigt, dass der DNS-Server des Hosts auf 209.18.47.61 eingestellt ist. Das Ende der Ausgabe zeigt, dass der DNS-Request funktioniert hat.

Listing Q.14 nslookup-Befehl (Mac)

```
Wendell-Odoms-iMac: wendellodom$ nslookup www.certskills.com
Server:          209.18.47.61
Address:         209.18.47.61#53

Non-authoritative answer:
www.certskills.com    canonical name = certskills.com.
Name:    certskills.com
Address: 173.227.251.150
```

Eine kurze Randbemerkung: Router und Switches haben ein paar mit DNS zusammenhängende Einstellungen. Doch diese Einstellungen erlauben es dem Router oder Switch nur, als DNS-Resolver (Client) zu agieren. Das heißt, Router und Switch werden DNS-Nachrichten nutzen, um den DNS-Server aufzufordern, den Namen mit der entsprechenden IP-Adresse aufzulösen. Um einen Router oder Switch so zu konfigurieren, dass die Hostnamen in deren entsprechende Adressen aufgelöst werden (alle globalen Befehle), lautet der Befehl:

- **ip name-server** *server_IP…*: Sie können mit diesem Befehl bis zu sechs DNS-Server konfigurieren.
- **ip host** *name address*: Sie können auf diesem Router oder Switch einen Namen und die entsprechende IP-Adresse statisch konfigurieren. Der lokale Router bzw. Switch wird diese IP-Adresse nur dann verwenden, wenn sich ein Befehl auf diesen Namen bezieht.
- **no ip domain-lookup**: Deaktiviert die DNS-Resolver-Funktion, sodass der Router oder Switch keinen DNS-Server zum Auflösen der Namen auffordert. (Der Befehl **ip domain-lookup** ist eine Standardeinstellung und veranlasst den Router, einen DNS-Server zu verwenden.)

HINWEIS Das Material in diesem Abschnitt ist nicht in den Vorauflagen erschienen. Es war für eine frühere Auflage vorgesehen, wurde aber aus Platzgründen weggelassen. Wir führen es an dieser Stelle für interessierte Leser auf.

Umsortierung von ACEs durch das IOS

Unter bestimmten Umständen gewinnt man den Eindruck, dass das IOS sich an Ihrer ACL zu schaffen gemacht hat. Dieses Verhalten ist vor allem für solche Benutzer verwirrend, die erst begonnen haben, sich mit ACLs auseinanderzusetzen. Im nun folgenden Thema zeigen wir zwei Szenarien, in denen das IOS Ihre ACL tatsächlich modifiziert. Wir beschreiben, was genau das IOS tut und warum keine dieser Änderungen sich auf die Logik Ihrer ACL auswirkt.

Wenn Sie sich mit ACLs vertraut machen, hören und lesen Sie vom allerersten Moment an Sätze wie die folgenden:

- Die Reihenfolge der ACL-Anweisungen – der sogenannten ACEs (Access Control Entrys) – ist wichtig, weil das IOS auf Grundlage des ersten ACE-Treffers in einer ACL handelt.
- Wenn Sie beim Konfigurieren der ACL die Zeilennummern ignorieren, bestimmt das IOS anhand der Reihenfolge, in der die Befehle eingegeben wurden, den Suchablauf in den ACL-ACEs.
- Haben Sie Zeilennummern konfiguriert, dann geht aus diesen hervor, in welcher Reihenfolge die Anweisungen beim ACL-Vergleich bearbeitet werden.

Zunächst einmal kann das IOS einschätzen, ob die Reihenfolge der Anweisungen ohne Beeinträchtigung der ACL-Logik geändert werden kann, um die interne Verarbeitung der ACL zu beschleunigen. Dies ist nicht in allen, aber durchaus in manchen Fällen möglich. Generell gilt: Je weniger Überschneidungen es zwischen den Anweisungen in der ACL gibt, desto höher ist die Wahrscheinlichkeit, dass das IOS Anweisungen umsortieren kann – und wird.

Betrachten wir exemplarisch die ersten drei Zeilen der ACL *anyorder* in Listing Q.15. Jeder ACE vergleicht einen Host.

Listing Q.15 Beispiel für die Umsortierung von ACEs in einer ACL durch das IOS

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip access-list standard anyorder
R1(config-std-nacl)# permit 9.9.9.1
R1(config-std-nacl)# permit 9.9.9.2
R1(config-std-nacl)# permit 9.9.9.3
R1(config-std-nacl)# ^Z
R1#
```

Sie könnten die ACEs in der ACL aus Listing Q.15 in beliebiger Reihenfolge umsortieren: Die ACL würde trotzdem genauso funktionieren wie vorher, weil es bei der Vergleichslogik der drei Anweisungen keine Überschneidungen gibt. Und weil eine Umsortierung dieser drei ACEs keine Auswirkungen hätte, macht sich das IOS diese Tatsache zunutze und sortiert die ACEs um. Aber warum?

Das Cisco IOS untersucht und optimiert jede ACL für den internen Suchalgorithmus, der beim Vergleich von Paketen mit der ACL vom Router verwendet wird. Was die Besonderheiten der internen Algorithmen angeht, hält sich Cisco eher bedeckt. Wichtig ist jedoch, dass das IOS die

Reihenfolge zur Suchoptimierung nur dann ändert, wenn diese Änderung keine Auswirkung auf die Logik der ACL hat. Wir fassen zusammen:

Schlüssel-thema

> Das IOS kann die Vergleichsreihenfolge für ACEs in einer ACL ändern, dies jedoch nur, wenn sich hieraus keine Auswirkungen auf die von der ACL getroffenen Entscheidungen ergeben.

Die Person, die die in Listing Q.15 gezeigte ACL erstellt hat, hätte die einzelnen ACEs in beliebiger Reihenfolge anordnen können, ohne die Logik zu ändern. Weil die Anweisungen jedoch im Konfigurationsmodus eingegeben wurden, gibt es für sie eine bestimmte Reihenfolge. Möglicherweise bevorzugt das IOS jedoch eine andere Reihenfolge. Listing Q.16 zeigt Auszüge aus der Ausgabe zweier **show**-Befehle, die zeigen, wie das IOS die Abfolge geändert hat, um sie an die interne Verarbeitung anzupassen.

Listing Q.16 Nachweis dafür, dass die Umsortierung keine Auswirkungen auf die Entscheidungen der ACL hat

```
R1# show access-list anyorder
Standard IP access list anyorder
    10 permit 9.9.9.1
    30 permit 9.9.9.3
    20 permit 9.9.9.2

R1# show running-config | section ip access-list standard anyorder
ip access-list standard anyorder
 permit 9.9.9.1
 permit 9.9.9.3
 permit 9.9.9.2
R1#
```

Wenn das IOS die Umsortierung für die interne Verarbeitung durchführt, zeigt uns die Ausgabe eines passenden **show**-Befehls, in welcher Reihenfolge das IOS die Anweisungen abarbeiten wird. In unserem Beispiel vergleicht es zuerst die Anweisung mit 9.9.9.1, danach folgen 9.9.9.3 und schließlich 9.9.9.2. Beachten Sie jedoch, dass sich die ACL-Zeilennummern der einzelnen ACEs nicht geändert haben.

Doch noch einmal: Wenn es Überschneidungen bei der ACE-Vergleichslogik gibt, dann ist die Reihenfolge der ACEs wichtig und sie wird deswegen vom IOS nicht geändert. Listing Q.17 zeigt ein Beispiel. Die ACL, die wir dort sehen, verwendet ACEs für den Vergleich dreier überschneidender Gruppen: den Host 9.9.9.9, alle mit 9.9.9 beginnenden Adressen und schließlich alle Adressen, die mit 9.9 beginnen. Da sich diese Adressbereiche überschneiden, würde jede Umsortierung der drei ACEs die Logik der ACL ändern. Folglich zeigt uns der Befehl **show access-lists** am Ende des Listings, dass das IOS die Reihenfolge der ACEs nicht angerührt hat: Wiederum entnehmen wir der Ausgabe die Reihenfolge, in der das IOS die ACL verarbeitet.

Listing Q.17 ACL-Beispiel, bei dem eine Umsortierung nicht möglich ist

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip access-list standard thisorderonly
R1(config-std-nacl)# permit 9.9.9.9
R1(config-std-nacl)# deny 9.9.9.0 0.0.0.255
R1(config-std-nacl)# permit 9.9.0.0 0.0.255.255
R1(config-std-nacl)# ^Z
R1#
*Aug 20 14:17:52.247: %SYS-5-CONFIG_I: Configured from console by console
R1#
R1# show run | section ip access-list standard thisorderonly
ip access-list standard thisorderonly
 permit 9.9.9.9
 deny   9.9.9.0 0.0.0.255
 permit 9.9.0.0 0.0.255.255
R1#
```

Neunummerierung der ACL-Zeilen in Zehnerschritten beim Neuladen

Unter dem IOS können Sie eine ACL direkt bearbeiten. Sie können ACL-Anweisungen zwischen anderen Anweisungen einfügen, indem Sie Zeilennummern angeben. Ferner können Sie auch einzelne ACL-Zeilen durch Angabe der Zeilennummern löschen und Sie können vollkommen beliebige Zeilennummern verwenden. Wenn Sie jedoch den Router neu starten, dann nummeriert das IOS die Zeilennummern fortlaufend neu, und zwar in Zehnerschritten beginnend bei 10. Ich scherze nicht!

Es zeigt sich, dass das IOS ein paar subtile Maßnahmen an den Zeilennummern durchführt. Der Zweck der Zeilennummern besteht schließlich darin, sich ihre relative Reihenfolge zu merken. Das IOS tut dies intern und deswegen speichert es die Zeilennummern auch nicht in der Konfiguration. Vielleicht ist es Ihnen aufgefallen: In der Ausgabe des Befehls **show running-config** in den zwei obigen Listings sind keine Zeilennummern vorhanden. Das liegt daran, dass das IOS die relative Reihenfolge in den Konfigurationsdateien vermerkt, nicht aber die konkreten Zeilennummern.

Dieses spezielle, einigermaßen verborgene Verhalten kann beim Erlernen von ACLs verwirrend sein. Sie konfigurieren eine ACL, verwenden Zeilennummern und fügen vielleicht irgendwo eine Zeile mit der Nummer 15 zwischen den Zeilen 10 und 20 ein. Schließlich sind Zeilennummern genau dafür da. Später setzen Sie den Befehl **copy running-config startup-config** ab und fahren Ihr Lab herunter. Am nächsten Morgen schalten Sie die Geräte wieder ein – nur um festzustellen, dass Ihre ACL ganz neue Nummern hat und die von Ihnen konfigurierte Nummer 15 weg ist! Die ACL-Anweisung ist natürlich noch vorhanden, aber sie hat eine andere Zeilennummer.

Beim Neunummerieren und Umsortieren im Rahmen eines Neustarts führt das IOS die folgenden Schritte aus:

Schritt 1: Suche beim Neustart nach allen vorhandenen ACLs.

Schritt 2: Ordne jedem ACL-Eintrag eine Zeilennummer auf Grundlage der relativen Reihenfolge zu. Beginne dabei bei 10 und erhöhe die Zeilennummern in Zehnerschritten.

Schritt 3: Führe danach gegebenenfalls die Umsortierung der ACL-Zeilen für die IOS-Verarbeitung durch.

HINWEIS Der Inhalt unter der Überschrift »NAT-Overloading (PAT) auf Consumer-Routern« wurde 2013 für die Prüfung 100-101 in Kapitel 24 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100–101* veröffentlicht. Der folgende kurze Abschnitt legt den Schwerpunkt auf die Verwendung der NAT durch Consumer-Router.

NAT-Overloading (PAT) auf Consumer-Routern

Cisco-Router der Consumer-Klasse aktivieren standardmäßig viele Features (darunter auch PAT). Die Strategie bei solchen Consumer-Routern erlaubt es, dass die Anwender einfach den Router mit seinen Kabeln installieren, ohne den Router konfigurieren zu müssen. Hier soll es nun darum gehen, wie man auf solchen Verbrauchersystemen PAT aktiviert, auch wenn man nur die Standardeinstellungen verwendet.

Zuerst sei – wie schon ganz früh in Kapitel 3, »Grundlagen zu WANs« – gesagt, dass Produkte, die man im Laden als »Router« kaufen kann, tatsächlich über viele Features verfügen: Darin steckt ein Router, ein LAN-Switch, oft mit einem WLAN-Access-Point und einer Sicherheits-Firewall. Auch PAT gehört oft schon dazu. Was die Hardware angeht, haben diese Router mehrere RJ-45-Ports, die mit »LAN« gekennzeichnet sind. Dies sind die Ports für die LAN-Switch-Funktion. Sie haben auch einen RJ-45-Port mit der Bezeichnung »WAN«. Dabei handelt es sich um einen weiteren Ethernet-Port, der wie ein Router-Interface agiert. Üblicherweise ist er an eine DSL-Leitung oder ein Kabelmodem angeschlossen, das wiederum die Verbindung ins Internet herstellt (siehe Abbildung Q.7).

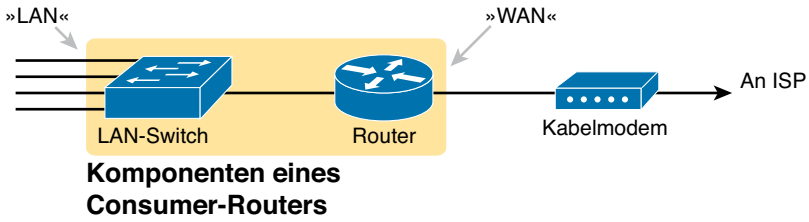

Abbildung Q.7 »Consumer-Router« mit LAN- und WAN-Ports

Um besser zu verstehen, wie ein Consumer-Router standardmäßig PAT ausführt, müssen Sie zuerst begreifen, wie er DHCP macht. Der Router agiert auf der LAN-Seite wie ein DHCP-Server und verwendet ein privates IP-Netzwerk, das vom Hersteller des Routers bereits vorab gewählt wurde. (Cisco-Produkte verwenden oft das private Klasse-C-Netzwerk 192.168.1.0.) Auf der WAN-Seite hingegen agiert der Router als DHCP-Client und least sich eine IP-Adresse vom DHCP-Server des ISP. Aber die Adresse vom ISP ist keine private, sondern eine öffentliche IPv4-Adresse (siehe Abbildung Q.8).

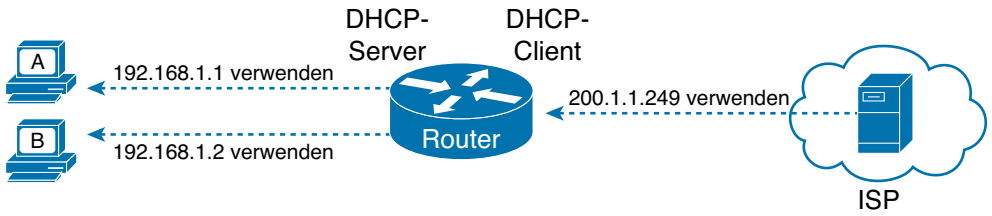

Abbildung Q.8 Consumer-Router als DHCP-Server im LAN, DHCP-Client im WAN

HINWEIS Damit ein Router der Unternehmensklasse wie der in Abbildung Q.8 auf dem WAN-Port eine IP-Adresse bezieht, nehmen Sie den Interfacesubbefehl **ip address dhcp**. Dieser Befehl weist den Router einfach an, die eigene Interface-IP-Adresse über DHCP selbst herauszufinden. Die DHCP-Server-Konfiguration arbeitet dann mit den gleichen Befehlen, wie sie detailliert in Kapitel 20, »DHCP und IP-Netzwerke auf Hosts«, zu finden sind.

Indem ein Consumer-Router sowohl auf der LAN- als auch der WAN-Seite mit DHCP arbeitet, hat er sich perfekte IP-Adressen gesucht, um PAT nutzen zu können. Die Computer im LAN verwenden alle »privaten« IP-Adressen, während der WAN-Anschluss mit einer »öffentlichen« IP-Adresse konfiguriert ist. Der Consumer-Router muss nun bloß noch PAT aktivieren, wobei seine LAN-Seite innerhalb das NAT vornimmt und der WAN-Port das NAT außerhalb macht (siehe Abbildung Q.9).

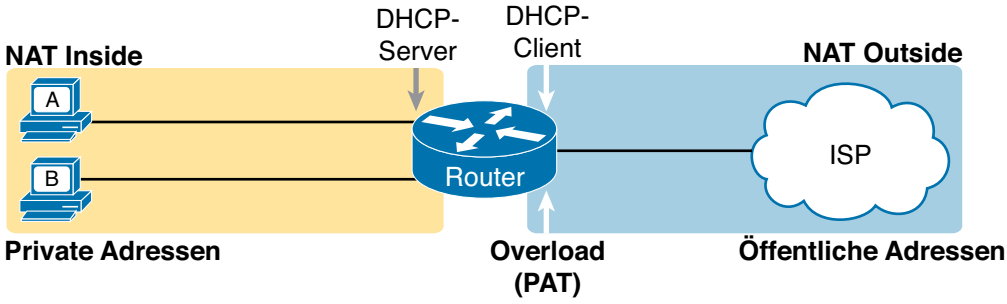

Abbildung Q.9 Positionen von DHCP- und NAT/PAT-Rollen in einem Consumer-Router

HINWEIS Der Inhalt unter der Überschrift »Dynamische Routen mit OSPFv3« wurde 2013 für die Prüfung 100-101 in Kapitel 29 des *Offiziellen Cisco-Zertifizierungshandbuchs zu CCENT/CCNA ICND1 100-101* veröffentlicht. Der folgende kurze Abschnitt legt den Schwerpunkt auf die Verwendung von OSPF zum Erlernen von IPv6-Routen.

Dynamische Routen mit OSPFv3

Auch wenn statische Routen funktionieren, arbeiten die meisten Internetworks mit einem dynamischen Routing-Protokoll, um die IPv6-Routen für alle Subnetze zu lernen, die nicht mit einem lokalen Router verbunden sind. Dieser letzte Hauptabschnitt beschäftigt sich mit dem IPv6-Routing-Protokoll OSPF Version 3 und konzentriert sich auf die Ähnlichkeiten mit OSPF Version 2 und deren Konfiguration.

Dieser Abschnitt beginnt mit den konzeptionellen und theoretischen Details von OSPF Version 3 (OSPFv3). Nachdem wir die Theorie abgehakt haben, gehen Sie die Konfiguration durch, die gleichzeitig einfacher und anders ist als die entsprechende OSPFv2-Konfiguration für IPv4. Dieser Abschnitt endet mit einer ganzen Reihe von Beispielen für **show**-Befehle, um herauszufinden, ob OSPFv3 korrekt funktioniert und IPv6-Routen lernen kann.

OSPF für IPv4 und IPv6 im Vergleich

Wie Sie wahrscheinlich erwartet haben, funktioniert OSPFv3 (die Version, die IPv6 unterstützt) weitgehend wie OSPFv2, das IPv4 unterstützt. Auf den nächsten Seiten wird es um die Terminologie, die Konzepte, Ähnlichkeiten und Unterschiede gehen, wie OSPF für IPv6 im Vergleich zu IPv4 arbeitet.

OSPF-Routing-Protokoll-Versionen und Protokolle

Wenn Techniker von »OSPF« sprechen, beziehen sie sich meistens auf OSPF, wie es mit IPv4 verwendet wird, und hier die Version 2 (OSPFv2). Es gab zwar einmal eine OSPF Version 1, aber Version 2 (OSPFv2) folgte bereits kurz darauf. Als OSPF sich Anfang bis Mitte der 1990er-Jahre als IPv4-Routing-Protokoll durchzusetzen begann, wurde stets OSPFv2 und nicht OSPFv1 eingesetzt. Also gab es schon in der Anfangszeit von OSPF keinen Bedarf zu klären, mit welcher Version man arbeitete, da alle Leute OSPFv2 nahmen und es einfach OSPF nannten.

Als Nächstes betrachten wir die Entwicklung der ursprünglichen IPv6-Protokolle Anfang bis Mitte der 1990er-Jahre. Abgesehen von IPv6 selbst mussten viele andere Protokolle aktualisiert werden, um mit IPv6 arbeiten zu können: ICMP, TCP, UDP usw. einschließlich OSPF. Als eine Arbeitsgruppe OSPF aktualisierte, damit es mit IPv6 arbeiten konnte, wie wurde es wohl genannt? Klar: OSPFv3.

Interessanterweise unterstützt OSPFv3 die Bekanntgabe von IPv6-, aber nicht von IPv4-Routen. Also versucht OSPFv3 nicht, in das vorhandene OSPFv2 den Support für IPv6 aufzunehmen. Auch wenn die Protokolle vieles gemeinsam haben, stellen Sie sich OSPFv2 und OSPFv3 als verschiedene Routing-Protokolle vor: eines für IPv4-Routen (OSPFv2) und eines für IPv6-Routen (OSPFv3).

Weil OSPFv3 IPv6-Routen (und nur IPv6-Routen) bekanntgibt, braucht ein Unternehmensnetzwerk, das mit einer Dual-Stack-Strategie arbeitet, tatsächlich beides: OSPFv2 und OSPFv3 (vorausgesetzt, dass in diesem Netzwerk überhaupt OSPF verwendet wird). Sind die zugrunde liegenden Konzepte sehr ähnlich? Ja. Doch auf jedem Router müssten sowohl ein OSPFv2- als auch ein OSPFv3-Routing-Protokoll-Prozess laufen. Beide müssten Nachbarschaftsbeziehungen bilden und beide Datenbankaktualisierungen senden und Routen berechnen. Also würde eine typische Migration von einem reinen IPv4-Unternehmensnetzwerk, das mit OSPFv2 arbeitet, zu einem

Dual-Stack-Ansatz mit IPv4 und IPv6 (auf allen Hosts und Routern laufen sowohl IPv4 als auch IPv6) folgende Schritte beinhalten:

Schritt 1: Vor IPv6 hat das Unternehmen mit IPv4 und OSPFv2 gearbeitet.

Schritt 2: Für den IPv6-Support ist geplant, Dual-Stack zu nutzen, wodurch auf den Routern im Unternehmensnetzwerk dann das Routing über IPv4 und IPv6 möglich wird.

Schritt 3: Um IPv6-Routing zu unterstützen, ergänzen die Unternehmen die Konfigurationen aller Router mit OSPFv3, aber sie müssen auch die OSPFv2-Konfiguration bewahren, damit IPv4-Pakete weiterhin weitergeleitet werden.

Andere Routing-Protokolle folgten einer ähnlichen Entwicklung, um IPv6 zu unterstützen, allerdings sind die Namen ungewöhnlicher. Im Falle von RIP (Routing Information Protocol) gibt es zwei Versionen, die IPv4 unterstützen, und die tragen wie zu erwarten die Namen RIP Version 1 (RIPv1) und RIP Version 2 (RIPv2). Für den Support von IPv6 hat eine Working Group eine neue RIP-Version entwickelt und sie *RIP next generation (RIPng)* genannt, wobei dieser Name auf die TV-Serie *Star Trek* anspielt. (Stimmt echt!) Das Enhanced Interior Gateway Routing Protocol (EIGRP) als ein proprietäres Cisco-Protokoll wurde immer einfach nur »EIGRP« genannt. Doch um die Diskussion zu vereinfachen, wird in manchen Dokumenten über den EIGRP-Support von IPv4 von »EIGRP« und für IPv6 von »EIGRPv6« gesprochen.

Tabelle Q.4 fasst die Terminologie für diese drei wichtigsten Interior-IP-Routing-Protokolle zusammen.

Tabelle Q.4 Zusammenfassung der Versionsterminologie für Interior-Routing-Protokolle

	RIP	OSPF	EIGRP
Aktuelle Version, die IPv4-Routen unterstützt	RIP Version 2 (RIPv2)	OSPF Version 2 (OSPFv2)	EIGRP
Version, die IPv6-Routen unterstützt	RIP next generation (RIPng)	OSPF Version 3 (OSPFv3)	EIGRP for IPv6 (EIGRPv6)

OSPFv2 und OSPFv3 im Vergleich

Im Rahmen unserer Ausführungen über Theorie und Konzepte von OSPF kann man sagen, dass sich OSPFv3 weitgehend wie OSPFv2 verhält. Beispielsweise verwenden beide die Link-State-Strategie. Beide nutzen die gleiche Metrik. Und die Liste ist noch länger, weil die Protokolle in der Tat viele Gemeinsamkeiten aufweisen. Die folgende Liste enthält viele Merkmale, die bei OSPFv2 und OSPFv3 ähnlich sind:

- Beide sind Link-State-Protokolle.
- Beide arbeiten beim Area-Design mit den gleichen Konzepten und Begriffen.
- Bei beiden muss das Routing-Protokoll auf dem Interface aktiviert werden.
- Nach Aktivierung auf einem Interface müssen beide dann versuchen, die über eine Datenverbindung bzw. ein Interface verbundenen Nachbarn zu finden.

- Beide prüfen bestimmte Einstellungen, bevor ein Router zum Nachbarn eines anderen Routers werden kann. (Die Liste der Überprüfungen ist bei OSPFv2 und OSPFv3 geringfügig anders.)
- Nachdem zwei Router zu Nachbarn geworden sind, fahren sowohl OSPFv2 als auch OSPFv3 damit fort, die Inhalte ihrer LSDB (Link-State Databases) – also die LSA (Link-State Advertisements), die die Netzwerktopologie beschreiben – zwischen den beiden Nachbarn auszutauschen.
- Nachdem alle LSAs ausgetauscht wurden, verwenden OSPFv2 und OSPFv3 den SPF-Algorithmus (Shortest Path First), um die beste Route zu jedem Subnetz zu berechnen.
- Beide arbeiten mit dem gleichen Metrikkonzept, das auf den Kosten für jedes Interface basiert und die gleichen Standardwerte für die Kosten verwendet.
- Beide beschreiben mit LSAs die Topologie, wobei sich die Funktionsweise der LSAs ein wenig unterscheidet.

Der größte Unterschied zwischen OSPFv3 und dem älteren OSPFv2 liegt bei einigen Interna und der Konfiguration. OSPFv3 ändert die Struktur einiger OSPF-LSAs, doch im Rahmen dieses Buchs interessieren diese Unterschiede nicht. OSPFv3 nutzt einen direkteren Ansatz für die Konfiguration und aktiviert OSPFv3 auf jedem Interface anhand eines Interfacesubbefehls.

Für den späteren Vergleich mit der OSPFv3-Konfiguration zeigt Abbildung Q.10 die Struktur der Konfiguration für OSPFv2. Darin erkennen Sie, dass der OSPFv2-Subbefehl **network** (wird im Router-Konfigurationsmodus ausgeführt) sich auf die IPv4-Adresse eines Interface bezieht, der dann das Interface identifiziert, auf dem OSPFv2 aktiviert werden soll. Anders formuliert wird bei der OSPFv2-Konfiguration das Interface nicht direkt erwähnt.

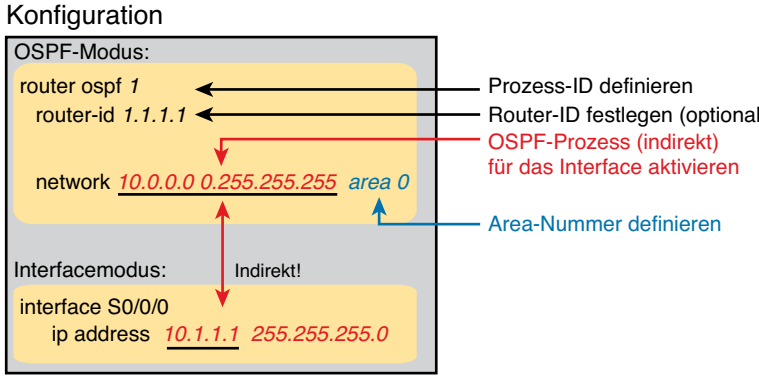

Abbildung Q.10 OSPFv2 aktiviert OSPF auf dem Interface indirekt.

Die OSPFv3-Konfiguration aktiviert direkt OSPF auf dem Interface, indem im Interface-Konfigurationsmodus ein Subbefehl eingefügt wird, um OSPFv3 für dieses Interface zu aktivieren. Tatsächlich kann man bei OSPFv3 gar keinen **network**-Subbefehl im Router-Konfigurationsmodus verwenden. Stattdessen arbeitet OSPFv3 mit dem Interfacesubbefehl **ipv6 ospf** *Prozess-ID* **area** *Area-ID* (siehe Abbildung Q.11). Dieser Befehl aktiviert den aufgelisteten OSPFv3-Prozess für dieses Interface und setzt die OSPFv3-Area.

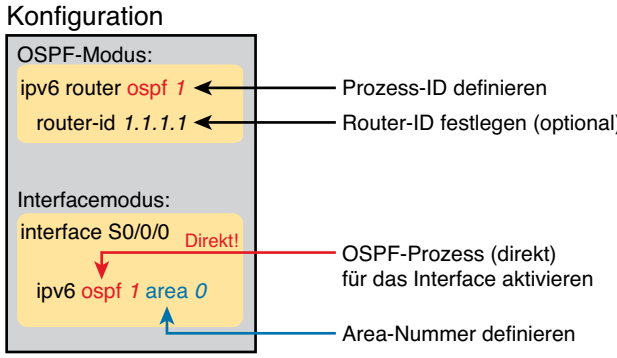

Abbildung Q.11 Die OSPFv3-Konfiguration aktiviert OSPF auf dem Interface direkt.

HINWEIS Das Cisco IOS unterstützt die Konfiguration von OSPFv2 anhand der gleichen Art Befehle, wie sie für OSPFv3 in Abbildung Q.11 gezeigt wurden. Das IOS unterstützt nur den neuen, direkteren Konfigurationsstil für OSPFv3, wie er in dieser Abbildung gezeigt wird.

Nachdem Sie nun eine allgemeine Vorstellung über Gemeinsamkeiten und Unterschiede zwischen OSPFv2 und OSPFv3 haben, zeigen wir im restlichen Abschnitt Beispiele, wie man OSPFv3 konfiguriert und überprüft.

Single-Area-OSPFv3 konfigurieren

Die OSPFv3-Konfiguration erfordert einige grundlegende Schritte: Eine Prozess-ID muss gewählt und konfiguriert werden, der Prozess muss auf jedem Interface aktiviert und der korrekte OSPF-Bereich (Area) für jedes Interface zugewiesen werden. Diese Details sollten in jeder Planungsinformation auflistet werden. Außerdem arbeitet dieses Buch nur mit Single-Area-Designs und somit sollten alle Interfaces der gleichen Area zugewiesen werden.

Das eine potenzielle Konfigurationsproblem ist die Router-ID (RID) von OSPFv3.

Zur Erinnerung: OSPFv2 verwendet eine 32-Bit-RID, die gewählt wird, wenn der OSPF-Prozess initialisiert wird. Wenn also OSPF zuerst konfiguriert wird, oder später, wenn der Router erneut gestartet wird, dann wählt der OSPFv2-Prozess eine Zahl, die er als seine RID verwendet. Der OSPFv2-Prozess wählt seine RID anhand der folgenden Liste aus:

1. Wenn der OSPF-Subbefehl **router-id** *rid* konfiguriert wird, wird dieser Wert benutzt und die Interface-IPv4-Adressen werden ignoriert.
2. Falls die Router-ID nicht mit dem Befehl **router-id** gesetzt wird, müssen alle Loopback-Interfaces geprüft werden, bei denen eine IPv4-Adresse konfiguriert wurde und die den Status *up* haben. Von denen muss die höchste numerische IP-Adresse gewählt werden.
3. Falls durch keinen der ersten beiden Punkte eine Router-ID gefunden werden kann, wählt der Router die höchste numerische IPv4-Adresse unter allen anderen betriebsbereiten Interfaces (d. h. solche mit dem Status *up*) aus. (Anders formuliert wird ein Interface mit dem Status *up/down* durch OSPF eingebunden, wenn dessen Router-ID gewählt wird.)

Interessanterweise verwendet OSPFv3 auch eine 32-Bit-RID anhand genau der gleichen Regeln wie OSPFv2. Die Nummer wird üblicherweise in punktdezimaler Notation (Dotted-Decimal Notation, DDN) aufgeführt. Das heißt, OSPFv3, das IPv6 unterstützt, hat eine Router-ID, die wie eine IPv4-Adresse aussieht.

Wenn man die gleichen RID-Auswahlregeln für OSPFv3 wie bei OSPFv2 verwendet, bleibt leider die Möglichkeit einer Fehlkonfiguration offen: Ein Router, der den OSPFv3-Befehl **router-id** nicht verwendet und für den keine IPv4-Adressen konfiguriert sind, kann keine RID wählen. Wenn der OSPFv3-Prozess keine RID hat, kann er nicht korrekt funktionieren, Nachbarschaftsbeziehungen bilden oder Routen austauschen.

Dieses Problem kann einfach gelöst werden. Bei der Konfiguration von OSPFv3 sollten Sie, wenn der Router keine IPv4-Adressen hat, darauf achten, die RID anhand des Router-Subbefehls **router-id** zu konfigurieren.

Abgesehen von diesem kleinen Problem ist die OSPFv3-Konfiguration relativ simpel. Die folgende Liste fasst die Konfigurationsschritte und -befehle zum Nachschlagen zusammen, die Beispiele folgen:

Schritt 1: Erstellen Sie eine OSPFv3-Prozessnummer und rufen Sie den OSPF-Konfigurationsmodus für diesen Prozess anhand des globalen Befehls **ipv6 router ospf** *Prozess-ID* auf.

Schritt 2: Sorgen Sie dafür, dass der Router eine OSPF-Router-ID hat, und zwar so:

- **a.** Konfigurieren Sie den Router-Subbefehl **router-id***id-wert*.
- **b.** Konfigurieren Sie eine IPv4-Adresse für irgendein Loopback-Interface vor, dessen Line-Status *up* ist – oder
- **c.** Konfigurieren Sie eine IPv4-Adresse für irgendein aktives Interface, dessen Line-Status *up* ist.

Schritt 3: Konfigurieren Sie den Befehl **ipv6 ospf** *Prozess-ID* **area** *Area-Nummer* für jedes Interface, auf dem OSPFv3 aktiviert werden soll. So aktivieren Sie in einem Rutsch das Interface für OSPFv3 und setzen die Area-Nummer.

Single-Area-Konfigurationsbeispiel für OSPFv3

In Abbildung Q.12 sehen Sie die Details eines Internetworks für ein OSPFv3-Konfigurationsbeispiel. Die Abbildung zeigt einen einzelnen Bereich (Area 0). Beachten Sie auch, dass die Router R2 und R3 mit dem gleichen VLAN und IPv6-Präfix verbunden sind.

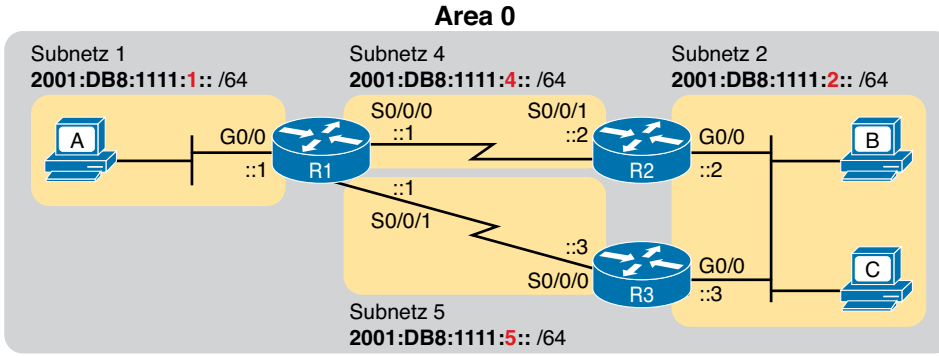

Abbildung Q.12 Single-Area-Design für ein OSPFv3-Konfigurationsbeispiel

Das nächste OSPFv3-Konfigurationsbeispiel arbeitet unter den folgenden Anforderungen:

- Alle Interfaces sollen in Area 0 sein, also beziehen sich alle **ipv6 ospf** *Prozess-ID* **area** *Area-Nummer*-Befehle auf diese Area.
- Jeder Router verwendet hier eine andere OSPF-Prozess-ID-Nummer. Das soll nur den Aspekt betonen, dass die Prozess-IDs nicht zu den benachbarten OSPFv3-Routern passen müssen.
- Jeder Router setzt seine Router-ID direkt mit dem Befehl **router-id** auf eine nahe liegende Zahl (1.1.1.1, 2.2.2.2 und 3.3.3.3 für R1, R2 bzw. R3).
- Die Router arbeiten nicht mit IPv4.

Listing Q.18 zeigt als Einstieg einen Ausschnitt aus dem Befehl **show running-config** für R1, bevor die OSPFv3-Konfiguration mit dem Befehl **ipv6 unicast-routing** eingefügt wurde, plus den Befehl **ipv6 address** für jedes Interface.

Listing Q.18 IPv6-Konfigurationsreferenz für R1 (vor Einfügen der OSPFv3-Konfiguration)

```
ipv6 unicast-routing
!
interface serial0/0/0
 no ip address
 ipv6 address 2001:db8:1111:4::1/64
!
interface serial0/0/1
 no ip address
 ipv6 address 2001:db8:1111:5::1/64
!
interface GigabitEthernet0/0
 no ip address
 ipv6 address 2001:db8:1111:1::1/64
```

Am Anfang von Listing Q.19 steht die OSPFv3-Konfiguration, beginnend auf Router R1. Beachten Sie, dass an diesem Punkt für Router R1 keine IPv4-Adressen konfiguriert sind. Also kann R1 sich nicht für eine OSPFv3-RID entscheiden: Er muss sich auf die Konfiguration des Befehls **router-id** verlassen. In diesem Beispiel finden die folgenden Vorgänge statt:

Schritt 1: Der Techniker fügt den globalen Befehl **ipv6 router ospf 1** ein und erstellt damit den OSPFv3-Prozess.

Schritt 2: Der Router versucht, eine OSPFv3-RID zuzuweisen, was nicht klappt, und deswegen wird eine Fehlermeldung ausgegeben.

Schritt 3: Der Techniker fügt den Befehl **router-id 1.1.1.1** ein, damit der OSPFv3-Prozess von R1 eine RID bekommt.

Schritt 4: Der Techniker fügt auf allen drei Interfaces den Befehl **ipv6 ospf 1 area 0** ein.

Listing Q.19 Zusätzliche Konfiguration auf R1, um OSPFv3 auf drei Interfaces zu aktivieren

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ipv6 router ospf 1
Jan  4 21:03:50.622: %OSPFv3-4-NORTRID: OSPFv3 process 1 could not pick a router-id,
  please configure manually
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)#
R1(config-rtr)# interface serial0/0/0
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# interface serial0/0/1
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# interface GigabitEthernet0/0
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# end
R1#
```

Mit Blick auf die Konfiguration eines einzelnen OSPFv3-Routers können nur zwei andere Arten von Parametern problematisch werden: die OSPF-Prozess-ID und die Area-Nummer. Bei der Überprüfung einer OSPFv3-Konfiguration sollten Sie zuerst die Prozess-ID-Nummern überprüfen und darauf achten, dass alle Werte zum Router passen. Die Prozess-ID des Befehls **ipv6 router ospf** *Prozess-ID* sollte zu allen Interfacesubbefehlen **ipv6 ospf** *Prozess-IDs passen*. Der andere Wert, die Area-Nummer, muss einfach zum Planungsdiagramm passen, aus dem ersichtlich wird, welche Interfaces in welchem Bereich (Area) sein sollen.

Beim Vergleich zweier benachbarter Router müssen einige Parameter passen oder die Router können nicht zu Nachbarn werden. Das Troubleshooting dieser Art von Problemen gehört zum Rahmen der 200-101 ICND2-Prüfung, aber nicht der 100-101 ICND1-Prüfung. Doch der eine große Unterschied zwischen OSPFv2 und OSPFv3 in dieser Checkliste für Nachbarn besteht darin, dass OSPFv3-Nachbarn keine passenden IPv6-Präfixe (Subnetze) oder Präfixlängen haben müssen. Anderenfalls müssen OSPFv3-Nachbarn bestimmte Elemente gemeinsam haben, z. B. dass die Router im gleichen Bereich (Area) sind oder das gleiche Hello- bzw. Dead-Intervall aufweisen.

Listing Q.20 zeigt eine vollständige Konfiguration für den Router R2. In diesem Fall verwendet der Router R2 eine andere OSPF-Prozess-ID als R1; die Prozess-ID für Nachbarn muss bei OSPFv2 oder OSPFv3 nicht übereinstimmen. R2 erstellt seinen OSPFv3-Prozess (2), setzt seine RID (2.2.2.2) und aktiviert OSPFv3 auf allen seinen drei Interfaces mit dem Interfacesubbefehl **ipv6 ospf 2 area 0**.

Listing Q.20 Vollständige IPv6-Konfiguration mit OSPFv3 auf Router R2

```
ipv6 unicast-routing
!
ipv6 router ospf 2
 router-id 2.2.2.2
!
interface serial0/0/1
 ipv6 address 2001:db8:1111:4::2
 ipv6 ospf 2 area 0
!
interface GigabitEthernet0/0
 ipv6 address 2001:db8:1111:2::2
 ipv6 ospf 2 area 0
```

Passive OSPFv3-Interfaces

Wie OSPFv2 kann auch OSPFv3 mit passiven Interfaces konfiguriert werden. Bei manchen IPv6-Subnetzen ist nur ein Router mit dem Subnetz verbunden. In solchen Fällen muss der Router OSPFv3 auf dem Interface aktivieren können, damit der Router das direkt verbundene Subnetz bekanntgeben kann, aber der Router muss nicht unbedingt versuchen, OSPFv3-Nachbarn über das Interface zu entdecken. In solchen Fällen kann der Techniker das Interface als passives OSPF-Interface konfigurieren, indem er den Router wie folgt beauftragt:

- Höre auf, OSPF-Hellos über das Interface zu versenden.
- Ignoriere über das Interface empfangene OSPF-Hellos.
- Bilde über das Interface keine Nachbarschaftsbeziehungen.
- Mache weiterhin alle Subnetze bekannt, die mit dem Interface verbunden sind.

Interessanterweise funktioniert die passive Interface-Konfiguration bei OSPFv2 genauso wie bei OSPFv3. So ist im Konfigurationsbeispiel aus Abbildung Q.12 nur R1 mit dem LAN-Subnetz links in der Abbildung verbunden, also könnte das R1-Interface G0/0 für OSPFv3 passiv gemacht werden. Dafür kann der Techniker den Subbefehl **passive-interface gigabitethernet0/0** im OSPFv3-Konfigurationsmodus auf Router R1 einfügen.

Status und Routen von OSPFv3 überprüfen

Um zu überprüfen, ob OSPFv3 funktioniert, gehen Sie auf zwei unterschiedliche Weisen vor. Sie können am Ende beginnen, indem Sie sich die IPv6-Routen anschauen, die durch OSPFv3 eingefügt wurden. Wenn die korrekten Routen in den korrekten Routing-Tabellen des Routers erscheinen, funktioniert OSPFv3 korrekt. Alternativ können Sie in der gleichen Reihenfolge vorgehen, wie OSPF diese Routen erstellt: Zuerst bestätigen Sie die Konfigurationseinstellungen, sehen sich dann die OSPF-Nachbarn an, dann die OSPF-Datenbank und schließlich die Routen, die OSPF in die Routing-Tabellen eingefügt hat.

Wenn es schnell gehen soll, sehen Sie sich zuerst die Routing-Tabelle an. Doch da es hier ums Lernen geht, ist es hilfreich, die Schritte von Anfang bis zum Ende durchzugehen und sich so durch verschiedene **show**-Befehle von OSPFv3 zu arbeiten. Im verbleibenden Abschnitt machen wir genau das mit verschiedenen OSPFv3-**show**-Befehlen:

- Zuerst werden die Konfigurationseinstellungen überprüft (OSPFv3-Prozess und Interfaces).
- Dann erfolgt die Überprüfung der OSPFv3-Nachbarn.
- Anschließend werden die OSPFv3-LSDB (Link-State Database) und LSAs geprüft.
- Zum Schluss werden OSPFv3-Routen gecheckt.

In diesem Abschnitt kommen verschiedene OSPFv3-**show**-Befehle vor, die Entsprechungen bei den OSPFv2-**show**-Befehlen haben. Tabelle Q.5 fasst diese **show**-Befehle der Übersicht halber zusammen.

Tabelle Q.5 OSPFv2- und passende OSPFv3-**show**-Befehle

Um Details darzustellen über ...	OSPFv2	OSPFv3
OSPF-Prozess	**show ip ospf**	**show ipv6 ospf**
Alle Quellen von Routing-Informationen	**show ip protocols**	**show ipv6 protocols**
Details über Interfaces, auf denen OSPF aktiviert wurde	**show ip ospf interface**	**show ipv6 ospf interface**
Kurzinfos über Interfaces, auf denen OSPF aktiviert wurde	**show ip ospf interface brief**	**show ipv6 ospf interface brief**
Nachbarschaftsliste	**show ip ospf neighbor**	**show ipv6 ospf neighbor**
Zusammenfassung von LSDB	**show ip ospf database**	**show ipv6 ospf database**
Per OSPF gelernte Routen	**show ip route ospf**	**show ipv6 route ospf**

HINWEIS Beachten Sie, dass alle OSPFv3-Befehle genau die gleichen Befehle wie bei IPv4 verwenden, nur kommt hier der Parameter **ipv6** statt **ip** vor.

OSPFv3-Konfigurationseinstellungen überprüfen

Um die OSPFv3-Konfiguration auf einem Router zu überprüfen, reicht ein einfacher **show running-config**-Befehl. Doch in einigen Fällen aus dem realen Leben und bei vielen Simlet-Fragen in den Prüfungen dürfen Sie vielleicht nicht den Enable-Modus aufrufen, um Befehle wie **show running-config** zu verwenden. In solchen Fällen können Sie die OSPFv3-Konfiguration mithilfe verschiedener **show**-Befehle erneut erstellen.

Dafür sehen Sie sich die Ausgabe des Befehls **show ipv6 ospf** an. Dieser Befehl listet Infos über den OSPFv3-Prozess selbst auf. Wie Sie dem Listing Q.21 entnehmen können, erfahren Sie in der ersten Ausgabezeile die folgenden Fakten über die Konfiguration:

- Der Router wurde mit der OSPFv3-Prozess-ID 1 konfiguriert, was bedeutet, dass der Befehl **ipv6 router ospf 1** konfiguriert wurde.
- Der Router verwendet die Router-ID 1.1.1.1, was bedeutet, dass entweder der Befehl **router-id 1.1.1.1** oder der Befehl **ip address 1.1.1.1** *Maske* für ein Interface des Routers konfiguriert wurde.

Listing Q.21 Überprüfung der OSPFv3-Prozesskonfiguration

```
R1# show ipv6 ospf
 Routing Process "ospfv3 1" with ID 1.1.1.1
 Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Graceful restart helper support enabled
 Reference bandwidth unit is 100 mbps
    Area BACKBONE(0)
        Number of interfaces in this area is 3
        SPF algorithm executed 4 times
        Number of LSA 13. Checksum Sum 0x074B38
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

Die Hervorhebungen gegen Ende von Listing Q.21 geben einige Hinweise auf die restliche Konfiguration, aber nicht genug Details, um jeden Teil der verbleibenden OSPFv3-Konfiguration aufzulisten. Die markierten Zeilen führen drei Interfaces auf, die sich »in this area« (in diesem Bereich) befinden, und dass es sich bei der Area um Backbone-Area 0 handelt. Alle diese Infos finden sich unter der Überschriftszeile oben im Beispiel für den OSPFv3-Prozess »1«. Diese Tatsachen zusammengenommen sagen uns, dass dieser Router (R1) die folgende Konfiguration verwendet:

- Interfacesubbefehl **ipv6 ospf 1 area 0**.
- Der Router verwendet diesen Interfacesubbefehl für drei Interfaces.

Doch der Befehl **show ipv6 ospf** identifiziert nicht die Interfaces, auf denen OSPFv3 aktiv ist. Um diese Interfaces zu finden, nehmen Sie einen der beiden Befehle von Listing Q.22. Konzentrieren wir uns auf den Befehl **show ipv6 ospf interface brief**: Dieser listet eine Zeile für jedes Interface auf, auf dem OSPFv3 aktiviert ist. Jede Zeile führt das Interface, die OSPFv3-Prozess-ID (unter der Überschrift »PID«), die dem Interface zugewiesene Area und die Zahl der OSPFv3-Nachbarn (Überschrift »Nbrs«, neighbors) auf, die auf diesem Interface gelernt wurden.

Listing Q.22 Überprüfung der OSPFv3-Interfaces

```
R1# show ipv6 ospf interface brief
Interface    PID   Area            Intf ID    Cost  State Nbrs F/C
Gi0/0        1     0               3          1     DR    0/0
Se0/0/1      1     0               7          64    P2P   1/1
Se0/0/0      1     0               6          64    P2P   1/1

R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 1"
  Interfaces (Area 0):
    GigabitEthernet0/0
    Serial0/0/1
    Serial0/0/0
  Redistribution:
    None
```

HINWEIS So wie der Befehl **show ip ospf interface brief** listet auch **show ipv6 ospf interface brief** sowohl passive als auch nichtpassive OSPFv3-Interfaces auf.

Die zweite Hälfte der Ausgabe von Listing Q.22 listet mit dem Befehl **show ipv6 protocols** Informationen über jede Quelle von IPv6-Routen auf dem Router auf. Dieser Befehl gibt merklich weniger Details über OSPFv3 aus als der Befehl **show ip protocol** über OSPFv2. Doch beide Befehle listen die Interfaces auf, bei denen OSPFv3 aktiviert ist.

Beide Befehle aus Listing Q.22 zeigen genug Informationen, um feststellen zu können, dass dieser Router (R1) unter drei Interfaces den Subbefehl **ipv6 ospf 1 area 0** aufweist: G0/0, S0/0/0 und S0/0/1.

OSPFv3-Nachbarn überprüfen

Für die Überprüfung von OSPFv3-Nachbarn braucht man nur auf einen einzigen Befehl zu sehen: **show ipv6 ospf neighbor**. Dieser listet eine Zeile pro Nachbar auf, in der wesentliche Infos über diesen Nachbarn stehen. Insbesondere sind das die RID des Nachbarn, das lokale Interface des Routers, über den dieser Nachbar erreichbar ist, und der Status der Nachbarschaftsbeziehung.

Im OSPFv3-Beispiel dieses Kapitels soll jeder Router zwei Nachbarschaftsbeziehungen haben. R1 hat zwei serielle Verbindungen, jeweils eine zu den Routern R2 und R3, und somit bildet

R1 eine Nachbarschaftsbeziehung mit jedem dieser Router. R2 und R3 sind mit dem gleichen IPv6-Subnetz über ein LAN verbunden und bilden darüber eine Nachbarschaftsbeziehung. Abbildung Q.13 zeigt die erwarteten OSPFv3-Nachbarschaftsbeziehungen.

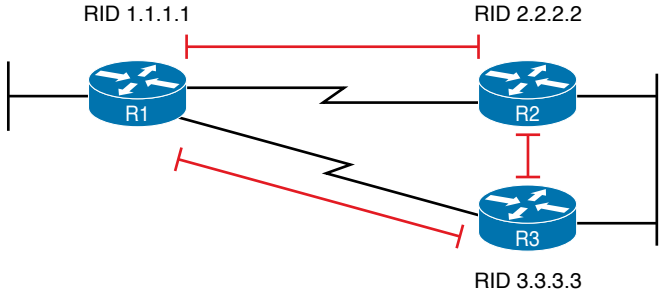

Abbildung Q.13 Erwartete OSPFv3-Nachbarschaftsbeziehungen im Vergleich zu Abbildung Q.12

Listing Q.23 zeigt die OSPFv3-Nachbarschaftsbeziehungen der Router R1 und R2. Der gekennzeichnete Bereich zeigt die Basisinfos, also die RIDs der benachbarten Router, das lokale Interface und den Status. Der Status »FULL« bedeutet, dass die Nachbarschaftsbeziehung funktioniert und dass die Nachbarn die LSAs vollständig ausgetauscht haben.

Listing Q.23 OSPFv3-Nachbarn auf den Routern R1 und R2 überprüfen

```
! Der erste Befehl ist von Router R1 und führt R2 und R3 auf
R1# show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
3.3.3.3           0   FULL/  -        00:00:39    6               Serial0/0/1
2.2.2.2           0   FULL/  -        00:00:31    7               Serial0/0/0
! Der nächste Befehl ist von Router R2 und führt R1 und R3 auf
R2# show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
1.1.1.1           0   FULL/  -        00:00:39    6               Serial0/0/1
3.3.3.3           1   FULL/DR         00:00:33    3               GigabitEthernet0/0
```

Bevor wir zum nächsten Thema übergehen, sehen Sie sich noch einmal die Ausgabe von Listing Q.23 an. Fällt Ihnen etwas auf, das Sie an IPv6 denken lässt statt an IPv4? OSPFv3 verwendet im DDN-Format angegebene 32-Bit-RIDs und gibt in der fünften Ausgabezeile anstelle der vollständigen IPv4-Adresse die IPv6-Interface-ID an.

Die OSPFv3-Datenbank untersuchen

OSPFv3 unterscheidet sich hinsichtlich der Theorie und den Details der OSPF-LSAs ein wenig von OSPFv2. In unserem Buch werden die LSA-Details weitgehend ignoriert, während sie im ICND2-Buch in gewissem Maße aufgegriffen werden. Um die Unterschiede zwischen OSPFv2 und OSPFv3 zu verstehen, müssten die Ausführungen deutlich tiefer gehen.

Allerdings kann man ganz einfach die LSDB in Grundzügen überprüfen, indem man dort nach Typ-1-Router-LSAs sucht. Ein Typ-1-LSA wird unter OSPFv2 und auch OSPFv3 für jeden Router eingesetzt, wobei das LSA den Router selbst beschreibt. Die Kennung des LSA ist mit der RID dieses Routers identisch. Innerhalb einer Area sollte die LSDB einen Typ-1-LSA für jeden Router der Area enthalten. Listing Q.24 zeigt den ersten Teil der Ausgabe des Befehls **show ipv6 ospf database** und verrät Ihnen, ob ein Router die Typ-1-Router-LSAs von den anderen Routern erfahren hat.

Listing Q.24 Überprüfung der OSPFv3-LSDB für R1

```
R2# show ipv6 ospf database

            OSPFv3 Router with ID (2.2.2.2) (Process ID 2)

                Router Link States (Area 0)

ADV Router       Age         Seq#       Fragment ID  Link count  Bits
 1.1.1.1         452         0x80000002  0           2           None
 2.2.2.2         456         0x80000004  0           2           None
 3.3.3.3         457         0x80000005  0           2           None

! Zeilen aus Platzgründen gekürzt
```

Das Beispiel zeigt drei Ausgabezeilen unter der Überschrift im Abschnitt namens »Router Link States«. Dieser Abschnitt zeigt Daten über die Typ-1-Router-LSAs. In diesem Abschnitt bezieht sich die Überschrift »ADV Router« auf den Router, der das LSA bekanntgegeben hat. In diesem Fall kennt R1 (RID 1.1.1.1) sein eigenes Typ-1-LSA plus das von R2 (RID 2.2.2.2) und von R3 (RID 3.3.3.3).

Durch OSPFv3 erlernte IPv6-Routen untersuchen

Schließlich ist der echte Beweis, dass OSPFv3 funktioniert, die Info, ob die Router IPv6-Routen lernen und in die IPv6-Routing-Tabelle aufnehmen. Dieser Abschnitt schließt den Überprüfungsvorgang mit einem Blick auf die von OSPFv3 eingefügten IPv6-Routen ab.

Bei korrekter Funktion lernen OSPFv3-Router genug Informationen, um Routen für alle IPv6-Präfixe (Subnetze) im Internetwork zu erstellen. Ein wesentlicher Unterschied zu OSPFv2 ist, dass über OSPFv3 gelernte Routen mit einer Link-Local-Next-Hop-Adresse arbeiten. Im Internetwork aus Abbildung Q.14, der das gleiche Design wie im OSPFv3-Konfigurationsbeispiel aufweist, fügt R2 beispielsweise eine Route zum Subnetz 1 links hinzu (Subnetz 2001:DB8:1111:1::/64). R2 verwendet die Link-Local-Adresse von R1 als Next-Hop-Adresse (siehe Abbildung).

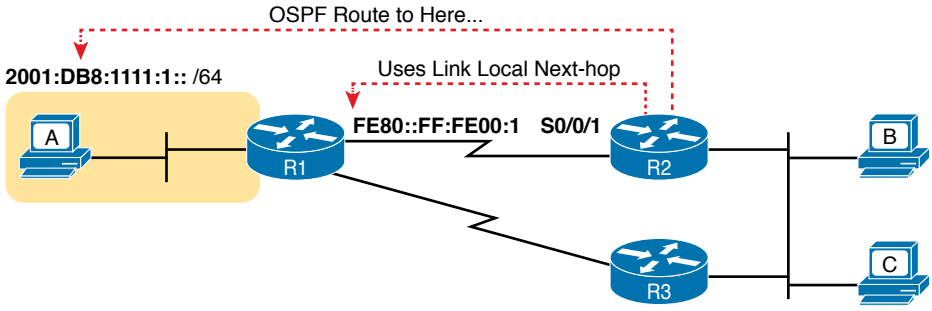

Abbildung Q.14 Wie OSPFv3 Link-Local-Adressen verwendet

Listing Q.25 zeigt die Ausgabe des Befehls show **ipv6 route ospf** für Router R2 für die Route, die in Abbildung Q.14 gekennzeichnet ist. Besonders wichtig ist nun Folgendes:

- Der Code-Buchstabe *O* bedeutet »OSPF«.
- In eckigen Klammern steht 110 als administrative Distanz von OSPF und 65 als Metrik für diese Route.
- Die Tatsache, dass die Route sowohl die Link-Local-Adresse als auch das ausgehende Interface aufführt

Listing Q.25 OSPFv3-Routen auf Router R2

```
R2# show ipv6 route ospf
IPv6 Routing Table - default - 9 entries
! Legende aus Platzgründen weggelassen

O   2001:DB8:1111:1::/64 [110/65]
     via FE80::FF:FE00:1, Serial0/0/1
O   2001:DB8:1111:5::/64 [110/65]
     via FE80::FF:FE00:3, GigabitEthernet0/0
```