Inhalt

OSPF (Open Shortest Path First)

- **OSPFv2**: IPv4
- **OSPFv3**: IPv6
- V2 und V3 sind strikt getrennt

Ablösung für Distant Vector Routing

- **RIP** (Routing Information Protocol)
    - Hop-Anzahl als Angabe für die Effizienz einer Route
    - Schlecht skalierbar

Link-State Routing (OSPF)

- Netzwerk kann in verschiedene „**Areas**“ aufgeteilt werden
- Link kann sein…
    - **Interface** an einem Router
    - **Netzwerksegment**, dass zwei Router verbindet
    - **Endnetzwerk**, das nur mit einem Router verbunden ist
- Link-State beinhält…
    - Netzwerk-Prefix
    - Subnetz-Prefix
    - Kosten

OSPF-Komponenten

OSPF-Pakete (Link-State-Packets)

1. **Hello Paket**:
    - Nachbarschaft mit anderen OSPF-Routern herstellen und halten
    - Beinhält Parameter, die bei benachbarten Routern übereinstimmen müssen, damit sie benachbart sein können
    - Designated-Router und Backup-Designated-Router aushandeln
    - Dead-Intervall: Zeit bis ein Nachbar-Router als down definiert wird und einen Topology-Change auslöst
2. **Database-Description-Paket (DBD)**:
    - Enthält abgekürzte Version der eigenen LSDB
    - LSDB muss überall gleich sein -> Empfänger gleicht DBD mit eigener LSDB ab
3. **Link-State-Request (LSR):** Nach mehr Informationen zu einem DBD fragen
4. **Link-State-Update (LSU):** Antwort auf LSR -> Bekanntmachen von Informationen durch LSAs
5. **Link-State-Acknowledgement (LSAck):** Empfänger bestätigt den Erhalt eines LSU-Pakets

Algorithmus

- Dijkstra Shortest-Path-First Algorithmus
- Errechnet die kumulativen Kosten um ein Ziel zu erreichen
- Erstellt SPF-Baum
    - Setzt jeden Router als Beginn des Baums
    - Errechnet dann den kürzesten Pfad zu jedem anderen Router
- Daraus entstehen die Routen für die Forwarding-Database

Datenstrukturen

- **Adjacency-Datenbank**
  - Benachbarte Router, zu denen es eine bidirektionale Verbindung gibt
  - Einzigartig pro Router
  - Show ip ospf neighbor
- **Link-State-Datenbank**
  - Alle anderen Router im Netzwerk
  - Stellt Topologie dar
  - Identisch bei jedem Router
  - Show ip ospf database
- **Forwarding-Datenbank**
  - Durch Algorithmus generierte Routen auf Basis der LSDB
  - Einzigartig pro Router
  - Show ip route

Ablauf: Link-State Operation

1. **Neighbor-Adjacency einrichten**
   - OSPF-Router müssen prüfen ob es weitere OSPF-Router in der „Nachbarschaft“ gibt
   - Senden „Hello“-Pakete aus allen OSPF-Schnittstellen
2. **Link-State-Advertisements (LSAs) austauschen**
   - Es werden Status und Kosten von jedem angebundenen Link an die Nachbarn ausgetauscht
   - Jeder Router sendet alle LSAs auch an alle anderen Router weiter, bis jeder Router alle LSAs hat
3. **Link-State Datenbank bauen**
   - Aus allen erhaltenen LSAs wird die LSDB gebaut
   - Daraus resultiert die Netzwerktopologie
4. **Ausführen des SPF-Algorithmus**
   - Es wird auf der Basis der LSDB der SPF-Baum erstellt
5. **Beste Route wählen**
   - Die besten Routen aus dem SPF-Baum zu jedem Netzwerk werden in die Routingtabelle aufgenommen, außer es gibt dort bereits eine Route zu diesem Netzwerk mit einer niedrigeren administrativen Distanz (z.B. statische Route)

Single- & Multi-Area

- **Single-Area**: Alle Router sind in derselben Area (Normalerweise: 0)
- **Multi-Area**:
  - Hierarchische Abtrennung von Areas
  - Alle Areas müssen mit dem Backbone (Area: 0) verbunden sein
  - Router zwischen Areas sind „Area Border Routers“ (ARBs)
  - Vorteile:
    - **Kleinere Routingtabellen**: Weniger Einträge, da Netzwerkadressen zusammengefasst werden können
    - **Reduzierter Berechnungsaufwand**: Neuberechnung der Datenbank nach Topology-Change nur für eigene Area
    - **Reduzierte Anzahl von Berechnungen**: LSA-Floods sind geringer, da weniger Router in derselben Area sind

Designated-Router (DR) Notwendigkeit

Problem

- Bei der Initialisierung oder bei Topology-Changes werden jedes Mal wieder LSAs von jedem Router versendet
- Netzwerk kann stark belastet werden
- Durch die Erstellung von Neighbor Ajacencies können viele Nachbarschaften entstehen
  ➔ Unnötig
    - Bei 5 Routern im selben Netz -> 10 Nachbarschaften (20 R -> 190 N): n(n-1) /2

Lösung

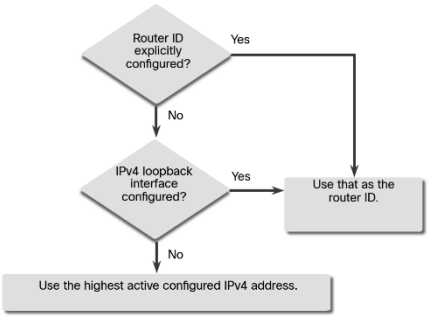
- Es wird ein Designated Router erklärt (Zusätzlich auch ein Backup Designated Router (BDR) für den Fall des Ausfalls des DR)
    - Router mit der höchsten konfigurierten Interface-Priorität wird DR, zweithöchste wird BDR
    - Router die mit der Prio 0 konfiguriert wurden können nicht DR oder BDR werden
    - Wenn keine Prio konfiguriert ist (alle Router Prio 1) wird die Router-ID zur Entscheidung verwendet
    - Kommt ein neuer Router mit einer höheren ID hinzu wird die Wahl nicht wiederholt
- Alle anderen Router werden zu DROthers
- Alle Router senden nur noch an DR und BDR LSAs
- Der aktive DR verteilt dann an alle anderen Router das empfangene LSA
- Beim Ausfall des DR übernimmt der BDR und der DROther mit der höchsten Prio wird zum neuen BDR

OSPF-Router-ID

- Elementarer Bestandteil von OSPF
- 32bit-Wert in Form einer IP-Adresse
- Jeder Router hat eine OSPF-Router-ID
    - Verwendet die ID um…
        - Am Synchronisierungsprozess der OSPF Datenbanken teilzunehmen (Höchste ID beginnt)
        - Um den DR (Höchste ID) und BDR (Zweithöchste ID) zu bestimmen

Bestimmungsprozess

1. Ist die ID manuell festgelegt -> Manuelle ID verwenden
2. Loopback-IP ist konfiguriert -> Loopback-IP als ID verwenden
3. Höchste aktive konfigurierte IP-Adresse verwenden
- Wird die ID im Nachgang verändert muss OSPF neugestartet werden, da bereits mit der vorherigen ID Nachbarschaften bekanntgemacht wurden



Wildcard-Maske

- **Berechnung**: /32-Subnetzmaske – Konfigurierte Subnetzmaske = Wildcard-Maske
    - Umkehrung der Subnetzmaske
- Notwendig bei der Netzwerkkonfiguration für OSPF an einem Router

Punkt-zu-Punkt OSPF-Netzwerke

- Direkte Verbindung zwischen zwei Routern
- Wahl des DR und BDR überflüssig
- Wenig Overhead, da LSAs nur zwischen zwei Routern ausgetauscht werden müssen
- Einfacher Aufbau

Multiaccess OSPF-Netzwerke

- Mehrere Router im selben Netzwerk
- DR und BDR werden gewählt um Netzlast zu reduzieren

Single-Area OSPF Netze anpassen

Kosten

- Kosten einer Route sind abhängig von der Bandbreite der Schnittstelle
    - Je höher die Bandbreite, desto niedriger die Kosten
- Kosten für 100M, 1G und 10G sind gleich, da sie immer auf 1 gerundet werden
    - Um bessere Leitungen zu priorisieren müssen die Kosten manuell konfiguriert werden -> Kosten für 100M und 1G sollten höher sein als für 10G
- **Berechnung**: Referenzbandbreite / Bandbreite = Kosten
- Kosten für eine Route werden aus der Summe der verwendeten Leitungen berechnet

Default-Route

- Wird verwendet um Traffic in andere Netze weiterzuleiten (z.B. Internet)
- **Autonomous-System-Boundary-Router (ASBR):** Router der zwischen einem OSPF- und einem Nicht-OSPF-Netz hängt
- Route muss am ASBR angelegt werden (ip route 0.0.0.0/0 next-hop)
- ASBR muss als Quelle der Route festgelegt werden für OSPF (default-information originate)

Befehle

Befehl	**Funktion**
router ospf *process-id*	OSPF aktivieren
router-id *id*	OSPF-ID festlegen
Network *network-address wildcard-mask* area *area-id* Oder Network *interface-ip* 0.0.0.0 area *area-id* Oder (pro Interface) Ip ospf *process-id* area *area-id*	OSPF Netzwerk an einem Router festlegen
Passive-interface *interface-id*	OSPF-Advertisements deaktivieren auf einem Interface
Show ip ospf interface *interface-id*	OSPF Informationen einer Netzwerkschnittstelle anzeigen
Ip ospf priority *value*	OSPF Priorität auf einer Netzwerkschnittstelle festlegen
Auto-cost reference-bandwidth *bandwidth*	Referenzbandbreite für die Berechnung der Kosten festlegen
Ip ospf cost *value*	Manuelles festlegen der Kosten

Access Control Lists (ACLs)

- Liste aus Access Control Entries
- Access Control Entries: Filter für Netzwerkverkehr (deny/allow)
- Jedes Paket wird mit den auf dem Interface festgelegten ACEs abgeglichen (**Layer 3 & 4**)
- Fördern Sicherheit und/oder Performance
- Können auf eingehenden (**inbound**) und ausgehenden (**outbound**) Verkehr konfiguriert werden
    - Inbound spart dem Router das heraussuchen der passenden Route
- **Ablauf**:
    - Jedes Paket durchläuft die ACL und damit jeden darin enthaltenen ACE nacheinander
    - Sobald ein ACE mit der Quell-IP übereinstimmt wird die ACE Aktion ausgeführt
    - Gibt es keinen Treffer wird das Paket verworfen, da am Ende jeder ACL ein implizites Deny-All existiert
- Bei der Erstellung von ACEs werden ebenfalls Wildcard-Masken verwendet (siehe OSPF)
    - 0.0.0.0 kann durch „host“ ersetzt werden
    - 255.255.255.255 kann durch „any“ ersetzt werden

Standard-ACLs

- Filter nur anhand der Quell-IP-Adresse
- Typischerweise am **Ziel** angewandt
- Können auch mittels „access-class“ auf Terminal-Schnittstellen gelegt werden um den Zugriff auf Geräte zu limitieren

Konfiguration

- Beispiele in grün

Nummerierte Standard-ACL

access-list *access-list-number* {deny | permit | remark *text*} *source* [*source-wildcard*]
access-list 10 permit 192.168.20.0 0.0.0.255
access-list 10 permit host 192.168.20.1

Benannte Standard-ACL

ip access-list standard *access-list-name*
ip access-list standard PERMIT-ACCESS
permit *source* [*source-wildcard*]
permit 192.168.20.0 0.0.0.255
permit host 192.168.10.10

Bearbeiten

Ip access-list standard {access-list-name | access-list-number}
Ip access-list standard PERMIT-ACCESS
Ip access-list standard 10
No sequence-number
No 10
Sequence-number {deny | permit | remark text} source [source-wildcard]
10 permit host 192.168.20.21

Extended-ACLs

- Filter anhand…
  - Quell-IP-Adresse
  - Ziel-IP-Adresse
  - TCP/UDP-Ports
  - Protokoll
- Typischerweise an der **Quelle** angewandt
  - Bandbreite einsparen
- Kann stateful arbeiten: TCP-Replies zulassen aber TCP-Requests nicht (established)

Konfiguration

- Beispiele in grün

Nummerierte Extended-ACL

access-list *access-list-number* {deny | permit | remark *text*} *protocol source source-wildcard* [*operator* {*port*}] *destination destination-wildcard* [*operator* {*port*}] [established]

access-list 10 permit tcp 192.168.10.0 0.0.0.255 any eq 443

access-list 20 permit tcp 192.168.10.0 0.0.0.255 192.168.0.0 0.0.255.255 eq www

Benannte Extended ACL

Ip access-list extended access-list-name

Ip access-list extended SURFING

{deny | permit | remark *text*} *protocol source source-wildcard* [*operator* {*port*}] *destination destination-wildcard* [*operator* {*port*}] [established]

Permit tcp 192.168.10.0 0.0.0.255 any eq 80

Permit tcp 192.168.10.0 0.0.0.255 any eq 443

Bearbeiten

Ip access-list extended {access-*list-name* | *access-list-number*}

Ip access-list extended SURFING

Ip access-list extended 10

No *sequence-number*

No 10

Sequence-number {deny | permit | remark *text*} *protocol source source-wildcard* [*operator* {*port*}] *destination destination-wildcard* [*operator* {*port*}] [established]

10 permit tcp 192.168.10.0 0.0.0.255 any eq www

Allgemeingültige Konfiguration

Anwenden auf Interface

ip access-group {*access-list-number* | *access-list-name*} {in | out}

ip access-group 10 out

ip access-group PERMIT-ACCESS out