

Titel: Labor Single-Area-OSPF

Klasse: 4BHIF

Name: Haiden

Gruppe: 01

Aufgabe: 27.04.2021 **Abgabe:** 04.05.2021

Inhaltsverzeichnis

1	Theorie-Teil.....	1
1.1	Allgemeines	1
1.2	Prinzipielle Funktionsweise	1
1.3	Vor- und Nachteile	2
2	Single Area OSPF.....	3
2.1	DHCP Router Konfig.....	3
2.1.1	Wellington	3
2.1.2	NYC Router	4
2.1.3	Wien Router	5
2.2	Internes Netz Konfiguration	6
2.2.1	NYC Router	6
2.2.2	Wellington Router	6
2.2.3	Wien Router	6
2.3	OSPF Konfiguration.....	7
2.3.1	NYC Router	7
2.3.2	Wellington Router	7
2.3.3	Wien Router	7
2.3.4	Neighbor Information.....	8
2.3.5	OSPF Database Information	8
2.4	Ping Tests über das Netzwerk in verschiedene Netzwerke.....	9
2.4.1	Von Wellington nach NYC.....	9
2.4.2	Von Wellington nach Wien.....	9
2.5	OSPF Timer	10
2.5.1	Verändern der Timer	11
2.6	MD5 Authentication.....	12
2.6.1	New York Router	12
2.6.2	Wellington Router	12
2.6.3	Wien Router	12
2.7	Prioritäten ändern	13
2.7.1	Einstellen der Priorität für Router Wien, damit er DR wird	13
2.7.2	Einstellen der Priorität für Router Wellington, damit BDR wird	13
2.7.3	Einstellen der Priorität für Router NYC, damit DROTHER	13
2.7.4	Verifizieren des Ergebnisses	13
2.8	Schalten sie den DR ab und zeichnen sie den Wahlvorgang auf.....	14
2.8.1	Debugging für OSPF Events einschalten.....	14

2.8.2	Debug Log	14
2.8.3	Beschreibung des Debug Prozesses	15
2.8.4	Wiedereinschalten des Wien Routers	15

1 Theorie-Teil

1.1 Allgemeines

<https://www.ip-insider.de/was-ist-ospf-open-shortest-path-first-a-905626/>

Die Abkürzung OSPF steht für Open Shortest Path First. Es handelt sich um ein Protokoll für dynamisches Routing in IP-Netzen. OSPF wird der Klasse der Link-State-Routingprotokolle zugeordnet. Die Berechnung der Routingtabellen und des besten Weges zu einem Ziel basiert auf dem Shortest-Path-First-Algorithmus (SPF-Algorithmus). OSPF ist eines der am häufigsten verwendeten Interior Gateway Protokolle (IGP) und kommt in größeren Netzwerken zum Einsatz.

Im Gegensatz zu RIP (Routing Information Protocol) skaliert es auch in großen Netzen, garantiert die Schleifenfreiheit, konvergiert bei Routenänderungen sehr schnell und ermöglicht Loadbalancing über gleichwertige Verbindungen. Open Shortest Path First verwendet nicht die Anzahl der Hops zu einem bestimmten Ziel als Entscheidungskriterium für die beste Route, sondern die Pfadkosten basierend auf Metriken wie die Datenrate einzelner Links. Das Protokoll ist in der Version 2 für IPv4 und in der Version 3 für IPv6 verfügbar. OSPF Version 2 ist in RFC 2328 spezifiziert, OSPF Version 3 im RFC 5340.

1.2 Prinzipielle Funktionsweise

Ein zentrales Funktionselement von OSPF ist die Topologie-Datenbank. Sie wird über den Austausch von Routinginformationen mit anderen Routern aufgebaut. Jeder Router übermittelt hierfür seine lokale Sicht des Netzwerks mit seinen über die einzelnen Routerinterfaces erreichbaren Nachbarn an alle anderen Router des Netzwerks. Mithilfe des Shortest Path Algorithmus erstellt jeder Router aus seiner Sicht einen Baum mit dem optimalen Weg zu einem bestimmten Ziel. Aus diesen Informationen kann dann die Routingtabelle erstellt werden. Die Routingtabelle enthält die Informationen, über welchen nächsten Router ein bestimmtes Netz mit welchen "Kosten" erreichbar ist.

Der Austausch der Router bezüglich ihrer jeweiligen lokalen Sicht des Netzwerks findet über so genannte Link State Advertisements (LSAs) statt. Es werden abhängig von den Routertypen und Netzbereichen verschiedene Typen von Link State Advertisements unterschieden, auf die hier nicht im Detail eingegangen werden kann.

Sehr wichtig für den Routingprozess von OSPF ist das Hello-Protokoll. Über regelmäßige Keepalive-Nachrichten erfahren die Nachbarrouter, ob ein Router noch funktioniert und seine Informationen nach wie vor Gültigkeit besitzen. Kommen keine Hello-Pakete eines bestimmten Routers mehr an, können dessen Informationen gestrichen und eine Neuberechnung des Shortest Path gestartet werden. Darüber hinaus kommt das Hello-Protokoll zur Entdeckung neuer Router und zur Wahl eines so genannten Designated Routers (DR) zum Einsatz.

OSPF unterscheidet verschiedene Routertypen wie den Designated Router, den Area Border Router (ABR) oder den Autonomous System Boundary Router (ASBR). Ihnen kommen jeweils spezifische Aufgaben zu. Beispielsweise ist der Designated Router dafür verantwortlich, die Aktualisierungsinformationen innerhalb eines bestimmten Netzbereichs (einer Area) zu verteilen.

Dadurch lässt sich die Menge an den zu übertragenden Link State Advertisements reduzieren. Neben dem DR wird auch ein Backup-Designated Router (BDR) gewählt. Area Border Router übernehmen Routing-Schnittstellenfunktionen zwischen einer bestimmten Area und der Backbone Area, die Autonomous System Boundary Router (ASBR) zwischen Autonomen Systemen (AS).

Grundsätzlich sind OSPF-Router bestimmten Areas zugeordnet. Mithilfe des Area-Konzepts lässt sich eine Hierarchie abbilden. Der Routing Traffic wird durch das Area-Konzept reduziert, da die innere Topologie einer Area den anderen Areas verborgen bleibt. Die Backbone Area verbindet die einzelnen Areas miteinander. Zusätzlich existieren noch weitere besondere Area-Typen wie Stub Areas, Transit Areas, Totally Stubby Areas, Not So Stubby Areas oder Totally Not So Stubby Areas.

1.3 Vor- und Nachteile

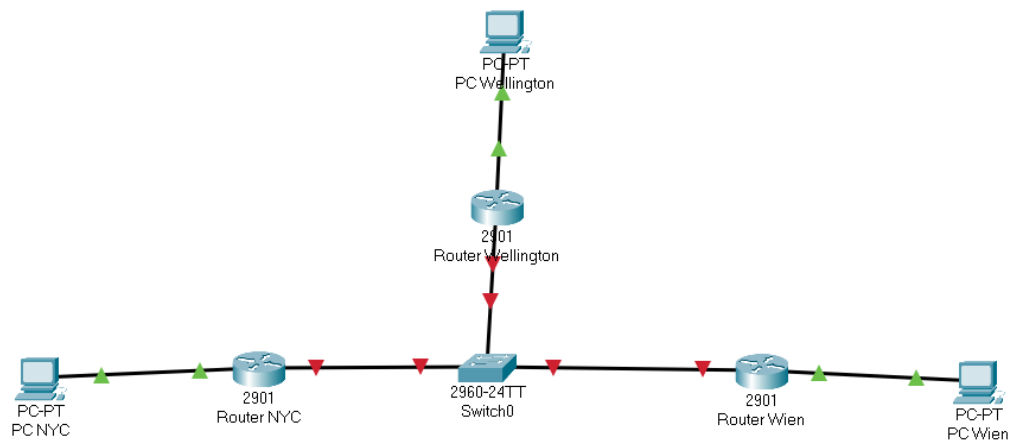
OSPF bietet gegenüber statischem Routing oder einfachen Routingprotokollen wie RIP zahlreiche Vorteile, aber auch einige Nachteile. Die Vorteile sind beispielsweise:

- Unterstützung hierarchischer Netzstrukturen und Unterteilung der Netze in Areas
- geeignet für große Netzwerke
- schnelle Konvergenz bei Veränderungen im Netz
- Sicherstellung der Schleifenfreiheit
- Wahl des besten Weges auf Basis verschiedener Informationen wie Kosten, Auslastung oder Bandbreiten
- dynamische Lastverteilung und Loadbalancing möglich
- Berücksichtigung des Type of Service (TOS) im Routing
- geringer Bandbreitebedarf für den Austausch der Routinginformationen
- Unterstützung der gegenseitigen Authentifizierung der OSPF-Router

Als Nachteile lassen sich aufführen:

- komplexes Protokoll
- tieferes Know-how zur Implementierung notwendig
- leistungsfähigere Router notwendig (Router benötigen eine gewisse Grundperformance und Intelligenz zur Unterstützung von OSPF)

2 Single Area OSPF

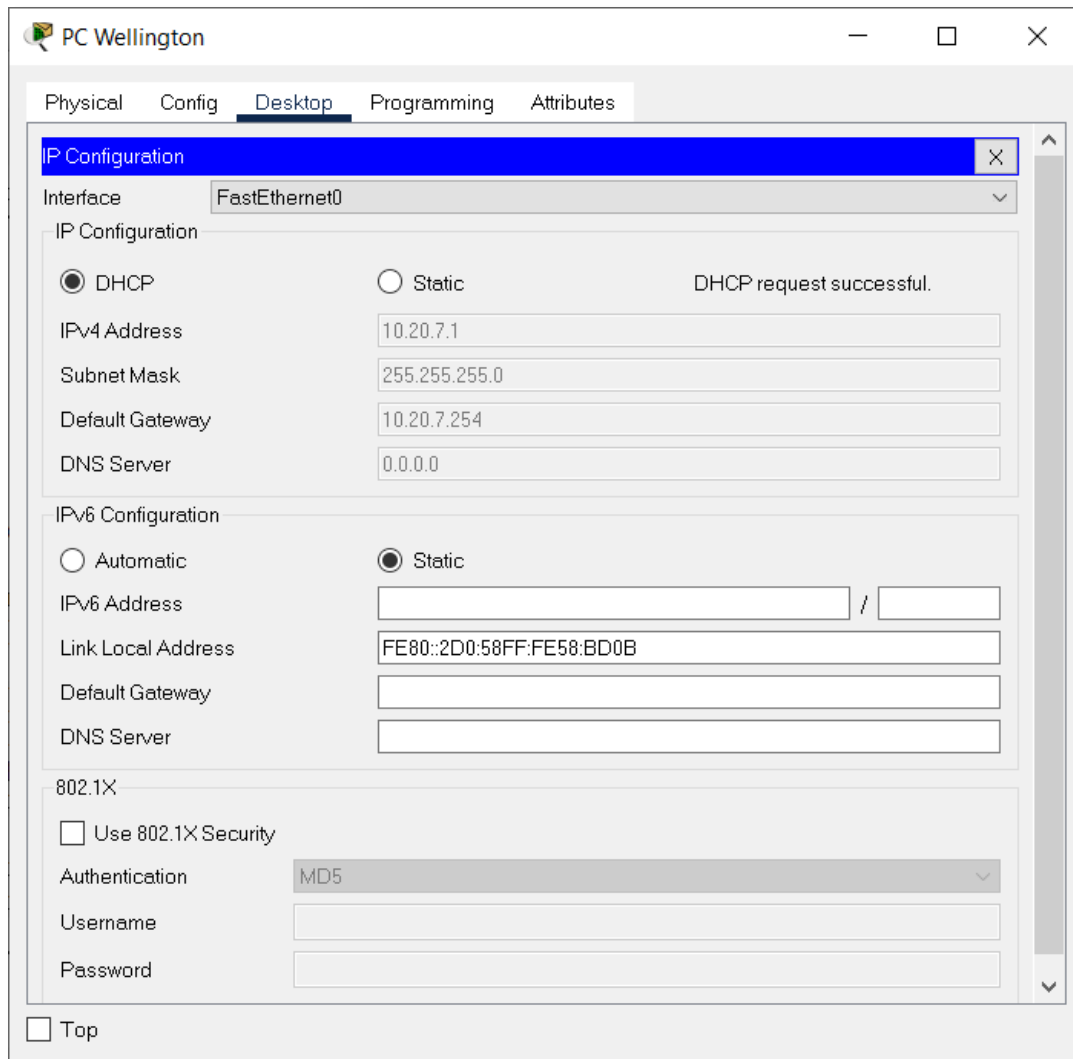


2.1 DHCP Router Konfig

2.1.1 Wellington

```
Wellington(config)#int g0/1
Wellington(config-if)#ip address 10.20.7.254 255.255.255.0
Wellington(config-if)#no shut
Wellington(config-if)#exit
Wellington(config)#ip dhcp pool internalniklas
Wellington(dhcp-config)#network 10.20.7.0 255.255.255.0
Wellington(dhcp-config)#default-router 10.20.7.254
Wellington(dhcp-config)#exit
```

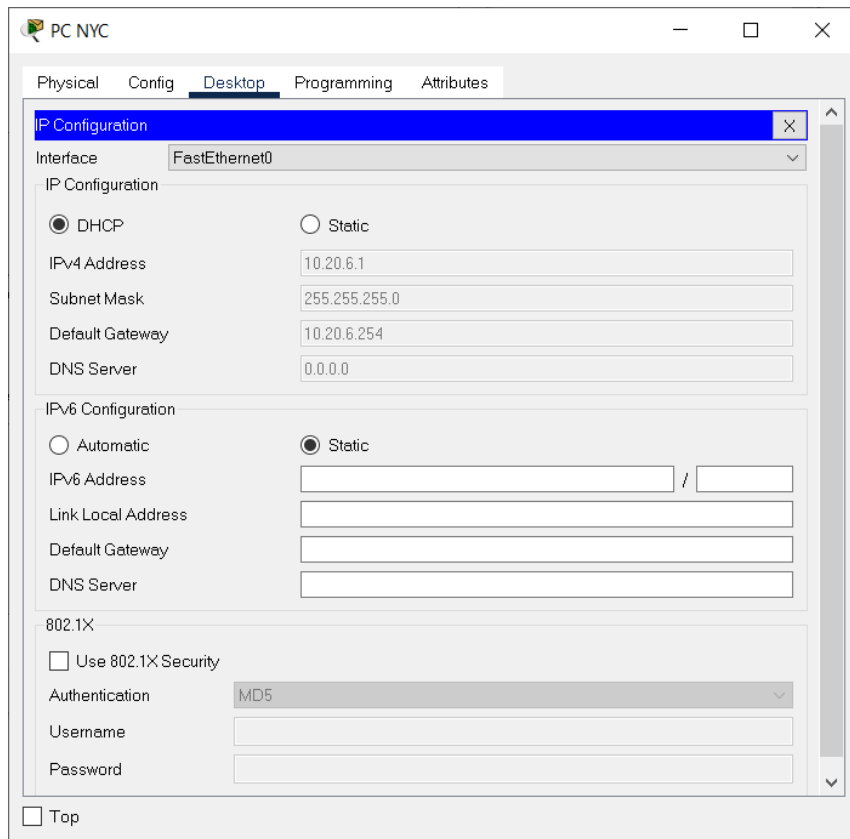
2.1.1.1 Wellington PC DHCP Test



2.1.2 NYC Router

```
NYC(config)#int g0/1
NYC(config-if)#ip address 10.20.6.254 255.255.255.0
NYC(config-if)#no shut
NYC(config-if)#exit
NYC(config)#ip dhcp pool internalniklas
NYC(dhcp-config)#network 10.20.6.0 255.255.255.0
NYC(dhcp-config)#default-router 10.20.6.254
NYC(dhcp-config)#exit
```

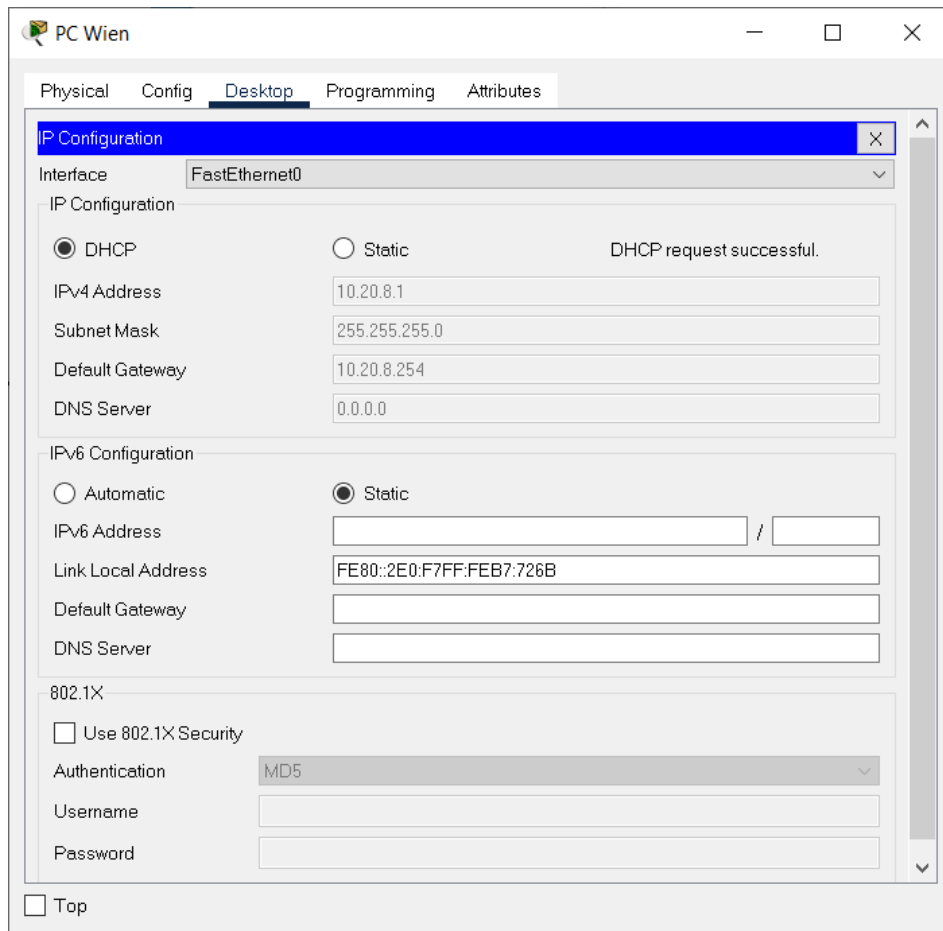
2.1.2.1 NYC PC DHCP Test



2.1.3 Wien Router

```
Wien(config)#int g0/1
Wien(config-if)#ip address 10.20.8.254 255.255.255.0
Wien(config-if)#no shut
Wien(config-if)#exit
Wien(config)#ip dhcp pool internalniklas
Wien(dhcp-config)#network 10.20.8.0 255.255.255.0
Wien(dhcp-config)#default-router 10.20.8.254
```


2.1.3.1 Wien(dhcp-config)#exit Wien PC DHCP Test



2.2 Internes Netz Konfiguration

2.2.1 NYC Router

```
NYC(config)#int g0/0
NYC(config-if)#ip address 192.168.6.1 255.255.255.0
NYC(config-if)#no shut
```

2.2.2 Wellington Router

```
Wellington(config)#int g0/0
Wellington(config-if)#ip address 192.168.6.2 255.255.255.0
Wellington(config-if)#no shut
Wellington(config-if)#exit
```

2.2.3 Wien Router

```
Wien(config)#int g0/0
Wien(config-if)#ip address 192.168.6.3 255.255.255.0
Wien(config-if)#no shut
Wien(config-if)#exit
```

2.3 OSPF Konfiguration

Mit `router ospf <NUM>` gelangt man in den Konfigurationsmodus für OSPF. Dort gibt man zuerst die Netzwerke, die advertised werden sollen an, danach die Router-ID. Am Ende einmal reloaden damit die Router-ID korrekt erkannt wird und überall korrekt angezeigt wird.

2.3.1 NYC Router

```
NYC(config)#router ospf 1
NYC(config-router)#network 192.168.6.0 0.0.0.255 area 0
NYC(config-router)#network 10.20.6.0 0.0.0.255 area 0
NYC(config-router)#router-id 6.6.6.6
NYC(config-router)#exit
NYC(config)#
NYC(config)#exit
NYC#
%SYS-5-CONFIG_I: Configured from console by console
clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

2.3.2 Wellington Router

```
Wellington(config)#router ospf 1
Wellington(config-router)#network 192.168.6.0 0.0.0.255
Wellington(config-router)#network 10.20.7.0 0.0.0.255 area 0
Wellington(config-router)#network 192.168.6.0 0.0.0.255 area 0
Wellington(config-router)#router-id 7.7.7.7
Wellington(config-router)#exit
```

2.3.3 Wien Router

```
Wien(config)#router ospf 1
Wien(config-router)#network 192.168.6.0 0.0.0.255 area 0
Wien(config-router)#network 10.20.8.0 0.0.0.255 area 0
Wien(config-router)#router-id 8.8.8.8
Wien(config-router)#Reload or use "clear ip ospf process" command,
for this to take effect
Wien(config-router)#exit
Wien(config)#exit
Wien#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

2.3.4 Neighbor Information

```
NYC# show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
7.7.7.7 1 FULL/DROTHER 00:00:32 192.168.6.2 GigabitEthernet0/0
8.8.8.8 1 FULL/DR 00:00:37 192.168.6.3 GigabitEthernet0/0
```

2.3.5 OSPF Database Information

```
NYC#show ip ospf database

OSPF Router with ID (6.6.6.6) (Process ID 1)

Router Link States (Area 0)

Link ID ADV Router Age Seq# Checksum Link count
192.168.6.3 192.168.6.3 1400 0x80000004 0x00fe8b 1
6.6.6.6 6.6.6.6 308 0x8000000b 0x00ba48 2
7.7.7.7 7.7.7.7 305 0x8000000a 0x003df2 1
8.8.8.8 8.8.8.8 300 0x80000008 0x000326 1

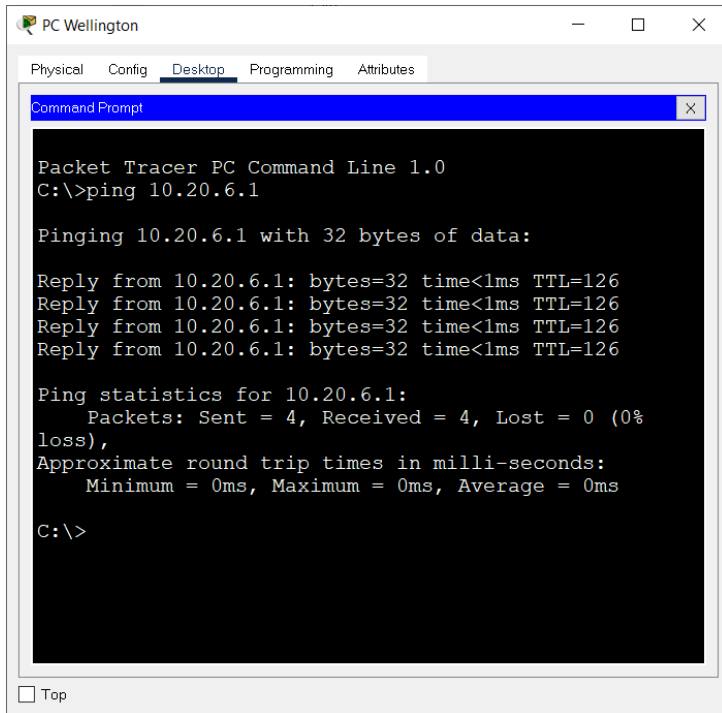
Net Link States (Area 0)

Link ID ADV Router Age Seq# Checksum
192.168.6.3 8.8.8.8 305 0x80000004 0x008be2
```

6.6.6.6 ist der Backup Designated Router, während 8.8.8.8 (Wien) der Designated Router ist.

2.4 Ping Tests über das Netzwerk in verschiedene Netzwerke

2.4.1 Von Wellington nach NYC



```
PC Wellington
Physical Config Desktop Programming Attributes
Command Prompt
Packet Tracer PC Command Line 1.0
C:\>ping 10.20.6.1

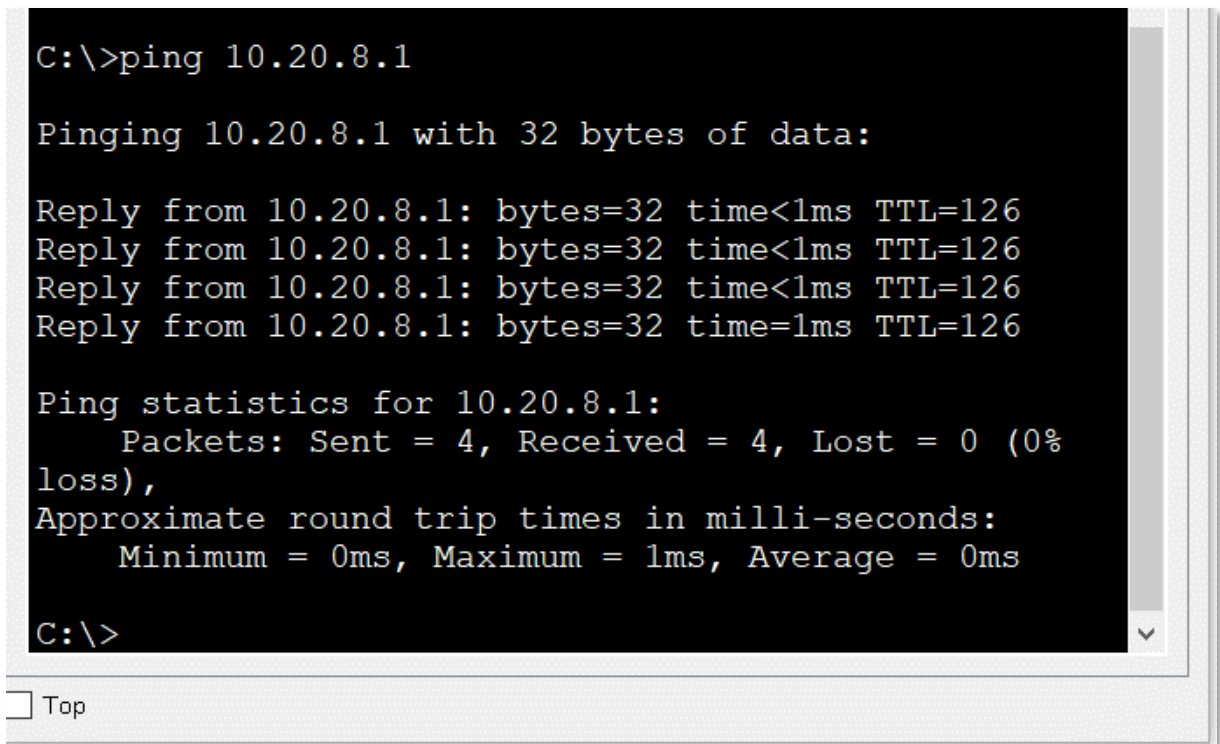
Pinging 10.20.6.1 with 32 bytes of data:

Reply from 10.20.6.1: bytes=32 time<1ms TTL=126
Reply from 10.20.6.1: bytes=32 time<1ms TTL=126
Reply from 10.20.6.1: bytes=32 time<1ms TTL=126
Reply from 10.20.6.1: bytes=32 time<1ms TTL=126

Ping statistics for 10.20.6.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0%
    loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

2.4.2 Von Wellington nach Wien



```
C:\>ping 10.20.8.1

Pinging 10.20.8.1 with 32 bytes of data:

Reply from 10.20.8.1: bytes=32 time<1ms TTL=126
Reply from 10.20.8.1: bytes=32 time<1ms TTL=126
Reply from 10.20.8.1: bytes=32 time<1ms TTL=126
Reply from 10.20.8.1: bytes=32 time=1ms TTL=126

Ping statistics for 10.20.8.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0%
    loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

2.5 OSPF Timer

Quelle: <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13689-17.html#timer>

OSPF hat einen Dead und Hello Timer.

Hello Timer: Hier sendet der Router alle 10 Sekunden ein OSPF Hello Packet. In Broadcast und PTP ist das Default 10 Sekunden, in NBMA Szenarien ist es 30 Sekunden.

Dead Timer: Zeit in Sekunden, die gewartet wird, bis ein Nachbar, der ausgefallen ist, als „tot“ bzw „dead“ erklärt wird. Per Default-Einstellung ist der Dead Timer das 4-fache wie der Hello Timer.

Wait—Timer interval that causes the interface to exit out of the wait period and select a DR on the network. This timer is always equal to the dead timer interval.

Retransmit—Time to wait before retransmitting a database description (DBD) packet when it has not been acknowledged.

Hello Due In—An OSPF hello packet is sent on this interface after this time. In this example, a hello is sent three seconds from the time the **show ip ospf interface** is issued.

Die Timer für ein OSPF Netzwerk sieht man per folgendem Kommando:

```
NYC#show ip ospf interface g0/0
GigabitEthernet0/0 is up, line protocol is up
Internet address is 192.168.6.1/24, Area 0
Process ID 1, Router ID 6.6.6.6, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 8.8.8.8, Interface address 192.168.6.3
Backup Designated Router (ID) 6.6.6.6, Interface address 192.168.6.1
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:01
Index 2/2, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 2, Adjacent neighbor count is 2
Adjacent with neighbor 7.7.7.7
Adjacent with neighbor 8.8.8.8 (Designated Router)
Suppress hello for 0 neighbor(s)
```

2.5.1 Verändern der Timer

Verändern wir wahlweise den Hello Timer auf 1 Sekunde und den Dead-Interval auf 4 Sekunden, d.h. dass die OSPF Teilnehmer jede Sekunde ein Hello Packet schicken müssen. Da die anderen Router dies nicht gewährleisten können, da sie anders eingestellt sind, werden sie detached, siehe folgendes Beispiel:

```
NYC(config)#int g0/0
NYC(config-if)#ip ospf hello-interval 1
NYC(config-if)#ip ospf dead-interval 4
NYC(config-if)#exit
NYC(config)#

01:44:19: %OSPF-5-ADJCHG: Process 1, Nbr 7.7.7.7 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired

01:44:19: %OSPF-5-ADJCHG: Process 1, Nbr 7.7.7.7 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down
or detached

01:44:24: %OSPF-5-ADJCHG: Process 1, Nbr 8.8.8.8 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired

01:44:24: %OSPF-5-ADJCHG: Process 1, Nbr 8.8.8.8 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down
or detached
```

Auf einem anderen Router tritt ebenfalls die selbe Fehlermeldung auf (Router Wien):

```
00:47:19: %OSPF-5-ADJCHG: Process 1, Nbr 6.6.6.6 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired

00:47:19: %OSPF-5-ADJCHG: Process 1, Nbr 6.6.6.6 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down
or detached
```

2.6 MD5 Authentication

Authentication kann genutzt werden, damit keine unautorisierten Router Zugang zum OSPF Gespann erhalten können. Es gibt verschiedene Authentication Methoden, darunter MD5, welches in folgendem Beispiel genommen wird.

2.6.1 New York Router

```
NYC(config)#router ospf 1
NYC(config-router)#area 0 authentication message-digest
NYC(config-router)#exit
NYC(config)#ip ospf message-digest-key 1 md5 niklas
^
% Invalid input detected at '^' marker.
NYC(config)#int g0/0
NYC(config-if)#ip ospf message-digest-key 1 md5 niklas
NYC(config-if)#
```

2.6.2 Wellington Router

```
Wellington(config)#router ospf 1
Wellington(config-router)#area 0 authentication message-digest
Wellington(config-router)#exit
Wellington(config)#int g0/0
Wellington(config-if)#ip ospf message-digest-key 1 md5 niklas
Wellington(config-if)#exit
```

2.6.3 Wien Router

```
Wien(config)#router ospf 1
Wien(config-router)#area 0 authentication message-digest
Wien(config-router)#exit
Wien(config)#int g0/0
Wien(config-if)#ip ospf message-digest-key 1 md5 niklas
Wien(config-if)#exit
```

2.7 Prioritäten ändern

Designated Router und Backup Designated Router werden je nach Priorität des jeweiligen Routers vergeben. Diese kann man im Interface Configuration Mode einstellen, wie folgend gezeigt. Der Router mit der höchsten Priorität wird Designated Router, der mit der zweithöchsten Priorität Backup Designated Router, alle anderen werden Designated Others, d.h. sie haben keine spezielle Rolle.

2.7.1 Einstellen der Priorität für Router Wien, damit er DR wird

Er bekommt die höchste Priorität, da er der DR werden soll.

```
Wien(config)#interface g0/0
Wien(config-if)#ip ospf priority 200
Wien(config-if)#
```

2.7.2 Einstellen der Priorität für Router Wellington, damit BDR wird

Damit Wellington der Backup Designated Router wird, müssen wir ihm die zweithöchste Priorität zuweisen.

```
Wellington(config)#int g0/0
Wellington(config-if)#ip ospf priority 199
Wellington(config-if)#exit
Wellington(config)#exit
```

2.7.3 Einstellen der Priorität für Router NYC, damit DROTHER

Und um zu verhindern, dass NYC BDR oder DR wird, wird seine Priorität auf das niedrigste Gespann im Trio gesetzt, so wird er zu einem DROTHER Router, d.h. er hat keine speziellen Funktionen.

```
NYC(config)#int g0/0
NYC(config-if)#ip ospf priority 198
NYC(config-if)#exit
NYC(config)#exit
```

2.7.4 Verifizieren des Ergebnisses

Nach dem neustarten der OSPF Prozesse auf allen 3 Routern ergibt sich folgendes Bild:

Aus der Sicht von NYC ist Wellington nun der BDR, Wien ist der DR, aufgrund der festgelegten Prioritäten.

```
NYC#show ip ospf neighbor
Neighbor ID Pri State Dead Time Address Interface
7.7.7.7 199 FULL/BDR 00:00:34 192.168.6.2 GigabitEthernet0/0
8.8.8.8 200 FULL/DR 00:00:30 192.168.6.3 GigabitEthernet0/0
```


2.8 Schalten sie den DR ab und zeichnen sie den Wahlvorgang auf.

Wir werden das Ganze aus der Perspektive des NYC Routers betrachten.

2.8.1 Debugging für OSPF Events einschalten

```
NYC#debug ip ospf adj
OSPF adjacency events debugging is on
NYC#
```

2.8.2 Debug Log

```
NYC#
02:37:35: %OSPF-5-ADJCHG: Process 1, Nbr 8.8.8.8 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired

02:37:35: %OSPF-5-ADJCHG: Process 1, Nbr 8.8.8.8 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down
or detached

02:37:35: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000022

02:37:35: OSPF: DR/BDR election on GigabitEthernet0/0

02:37:35: OSPF: Elect BDR 7.7.7.7

02:37:35: OSPF: Elect DR 7.7.7.7

02:37:35: DR: 7.7.7.7 (Id) BDR: 7.7.7.7 (Id)

02:37:35: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000023

02:37:35: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000024

02:37:40: OSPF: DR/BDR election on GigabitEthernet0/0

02:37:40: OSPF: Elect BDR 6.6.6.6
```

```
02:37:40: OSPF: Elect DR 7.7.7.7

02:37:40: OSPF: Elect BDR 6.6.6.6

02:37:40: OSPF: Elect DR 7.7.7.7

02:37:40: DR: 7.7.7.7 (Id) BDR: 6.6.6.6 (Id)

02:37:40: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000024

02:37:40: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000024
```

2.8.3 Beschreibung des Debug Prozesses

- NYC entdeckt beim nächsten Hello Packet schicken, dass Wien nicht mehr online ist.
- NYC baut ein Link State Advertisement Paket zusammen und schickt es aus
- Wellington Router wird als DR und BDR elected
- Wellington Router entdeckt das LSA Packet von Router NYC und es werden die Positionen neu ausgelost
- Aufgrund der vorher gewählten Prioritäten wird Wellington der neue DR und NYC der BDR.

Aus Sicht von NYC:

```
NYC#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
7.7.7.7 199 FULL/DR 00:00:37 192.168.6.2 GigabitEthernet0/0
```

Aus Sicht von Wellington:

```
Wellington#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
6.6.6.6 198 FULL/BDR 00:00:36 192.168.6.1 GigabitEthernet0/0
```

2.8.4 Wiedereinschalten des Wien Routers

```
Wien(config-if)#no shut
```

2.8.4.1 Debug Log von NYC

NYC#

02:44:42: OSPF: DR/BDR election on GigabitEthernet0/0

02:44:42: OSPF: Elect BDR 6.6.6.6

02:44:42: OSPF: Elect DR 7.7.7.7

02:44:42: DR: 7.7.7.7 (Id) BDR: 6.6.6.6 (Id)

02:44:48: OSPF: Rcv DBD from 8.8.8.8 on GigabitEthernet0/0 seq
0x62d4ea10 opt 0x00 flag 0x7 len 32 mtu 1500 state INIT

02:44:48: OSPF: Send DBD to 8.8.8.8 on GigabitEthernet0/0 seq
0x208e52db opt 0x00 flag 0x7 len 32

02:44:48: OSPF: NBR Negotiation Done. We are the SLAVE

02:44:48: OSPF: Send DBD to 8.8.8.8 on GigabitEthernet0/0 seq
0x62d4ea10 opt 0x00 flag 0x2 len 132

02:44:48: OSPF: DR/BDR election on GigabitEthernet0/0

02:44:48: OSPF: Elect BDR 6.6.6.6

02:44:48: OSPF: Elect DR 7.7.7.7

02:44:48: DR: 7.7.7.7 (Id) BDR: 6.6.6.6 (Id)

02:44:48: OSPF: Rcv DBD from 8.8.8.8 on GigabitEthernet0/0 seq
0x62d4ea11 opt 0x00 flag 0x3 len 92 mtu 1500 state EXCHANGE

02:44:48: OSPF: Send DBD to 8.8.8.8 on GigabitEthernet0/0 seq
0x62d4ea11 opt 0x00 flag 0x0 len 32

```
02:44:48: OSPF: Rcv DBD from 8.8.8.8 on GigabitEthernet0/0 seq
0x62d4ea12 opt 0x00 flag 0x1 len 32 mtu 1500 state EXCHANGE
```

```
02:44:48: OSPF: Send DBD to 8.8.8.8 on GigabitEthernet0/0 seq
0x62d4ea12 opt 0x00 flag 0x0 len 32
```

```
02:44:48: Exchange Done with 8.8.8.8 on GigabitEthernet0/0
```

```
02:44:48: Synchronized with with 8.8.8.8 on GigabitEthernet0/0,
state FULL
```

```
02:44:48: %OSPF-5-ADJCHG: Process 1, Nbr 8.8.8.8 on
GigabitEthernet0/0 from LOADING to FULL, Loading Done
```

```
02:44:48: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000024
```

```
02:44:48: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000024
```

```
02:44:48: OSPF: Send DBD to 8.8.8.8 on GigabitEthernet0/0 seq
0x62d4ea12 opt 0x00 flag 0x0 len 32
```

```
02:44:52: OSPF: DR/BDR election on GigabitEthernet0/0
```

```
02:44:52: OSPF: Elect BDR 6.6.6.6
```

```
02:44:52: OSPF: Elect DR 7.7.7.7
```

```
02:44:52: DR: 7.7.7.7 (Id) BDR: 6.6.6.6 (Id)
```

2.8.4.2 Erklärung:

Durch den neuen Router werden die Positionen neu ausgelost. Der NYC Router wird BDR, der Wellington Router wird DR. Diese berücksichtigen allerdings nicht die Priorität des Wien Routers. Zu Beginn ist also folgende Topologie zu beobachten:

```
Wien#show ip ospf neighbor
```

Neighbor	ID	Pri	State	Dead Time	Address	Interface
6.6.6.6	198	FULL/BDR	00:00:38	192.168.6.1	GigabitEthernet0/0	
7.7.7.7	199	FULL/DR	00:00:35	192.168.6.2	GigabitEthernet0/0	

Nach einiger Zeit wird Wien zum DR wieder neugewählt, Wellington wird wieder zum BDR Router gewählt, wie dem Lesen der Debug Messages von NYC zu entnehmen ist:

```
02:47:09: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000028

02:47:09: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000029

02:47:09: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000029

02:47:09: OSPF: DR/BDR election on GigabitEthernet0/0

02:47:09: OSPF: Elect BDR 7.7.7.7

02:47:09: OSPF: Elect DR 8.8.8.8

02:47:09: OSPF: Elect BDR 7.7.7.7

02:47:09: OSPF: Elect DR 8.8.8.8

02:47:09: DR: 8.8.8.8 (Id) BDR: 7.7.7.7 (Id)
```

```
02:47:09: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000029

02:47:09: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000029

02:47:09: OSPF: Build router LSA for area 0, router ID 6.6.6.6, seq
0xffffffff80000029

02:47:19: OSPF: DR/BDR election on GigabitEthernet0/0

02:47:19: OSPF: Elect BDR 7.7.7.7

02:47:19: OSPF: Elect DR 8.8.8.8

02:47:19: DR: 8.8.8.8 (Id) BDR: 7.7.7.7 (Id)
```

Danach ist alles wieder wie vorher.

```
NYC# show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
8.8.8.8 200 FULL/DR 00:00:35 192.168.6.3 GigabitEthernet0/0
7.7.7.7 199 FULL/BDR 00:00:35 192.168.6.2 GigabitEthernet0/0
```