IOS-Kommandoreferenz

Michael Dienert

3. April 2014

Inhaltsverzeichnis

1 Router

Für alle Routingprotokolle gültige Befehle:

```
R1(config)#no ip classless
R1(config)#ip classless     //seit ios 11.3 default
R1#show ip route
```

1.1 CISCO-Routingtabellen

Begriffe bla

Level-1-Route : das *Zielnetz* einer Level-1-Route hat einen kleineren oder gleichen Präfix wie der Präfix des Classful-Netzes (8, 16 oder 24) zu dem das Zielnetz gehört. **Beispiele:**

Default-Route : 0.0.0.0/0

Supernet-Route : 192.168.0.0/22

Klasse-C-Netz-Route : 192.168.1.0/24

Parent Route : eine Level-1-Route **ohne** GW / Exit-I/F; also eigentlich nur ein Zielnetz; hat so eine Art Header-Funktion in den Cisco-Routing-Tabs.

> Achtung!!! Der Präfix, der hinter der Parent-Route steht, ist in manchen Fällen der Präfix ihrer Child-Routes, manchmal aber der Classful-Präfix der Parent-Route!!! **Bravo!! Noch idiotischer hätte Cisco das nicht machen können**:

```
//Beispiel mit 24 = Praefix der Child-Route
172.16.0.0/24 is subnetted, 1 subnets
C        172.16.3.0 is directly connected, FastEthernet0/0

//Beispiel mit 16 = Class-B-Praefix der Parent-Route
172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C        172.16.1.4/30 is directly connected, Serial0/0/0
C        172.16.1.8/30 is directly connected, Serial0/0/1
C        172.16.3.0/24 is directly connected, FastEthernet0/0
```

Ultimate-Level-1-Route : eine Level-1-Route **mit** GW / Exit-I/F

Child Route = Level-2-Routes : Das sind Ultimate Routes, da Child-Routes immer ein GW / Exit-I/F haben.

Route aus Routingtabelle heraussuchen

1. Ziel passt auf Level-1-Ultimate-Route ⇒ forward
2. Ziel passt auf Level-1-Parent-Route ⇒ Punkt 3
3. Ziel passt auf Level-2-Child-Route ⇒ forward
4. Ziel passt auf keine Child-Route ⇒ Punkt 5
5. - Router-Config = no ip classless ⇒ drop

- Router-Config = ip classless ⇒ suche in Level-1-Routes nach Supernet / Default-Route ⇒ forward

6. Ziel passt nirgends ⇒ drop

1.2 Statisches Routing

```
R1(config)# ip route <IP/Network> <NETMASK> <NEXT HOP ROUTER/Interface>
```

1.3 Routingprotokolle

1.3.1 Distanzvektor-Protokolle mit periodischen Updates

Diverse Timer und Regeln zur Verhinderung von Routing-Loops:

Count to Infinity : in einer Routing Loop schaukelt sich die Metrik hoch: der maximale Metrik-Wert wird auf 16 begrenzt. 16 bedeutet **unerreichbar**.

Triggered Update : Triggered Update wird sofort versendet, wenn:

- ein I/F auf dem Router wechselt von up nach down oder umgekehrt
- eine Route wird als unerreichbar markiert oder von unerreichbar wieder nach ereichbar gesetzt
- eine (statische oder dynamisch erlernte) Route wird in der Routing-Tabelle eingetragen

Im Prinzip lösen Triggered Updates das Routing-Loop-Problem vollständig, sie können aber verloren gehen und verbreiten sich nicht schnell genug in einem grossen Netz. Ausserdem bleibt das Problem mit dem Wackelkontakt am Netzwerkstecker.

Update Timer : RIP arbeitet mit **periodischen Updates**: alle **30s** versendet ein RIP-Router die gesamte Routingtabelle an seine Nachbarn.

Invalid Timer : **180s**, trifft für eine Route innerhalb dieser Zeit kein Update ein, wird sie durch *Setzen der Metrik auf 16* als *unerreichbar markiert* und nach Ablauf des Flush-Timers (also 60s später) komplett gelöscht.

Flush Timer : **240s**, Invalid und Flush Timer werden gleichzeitig gestartet.

Holddown Timer : **240s**; Ablauf:

- Router empfängt Triggered Update über ausgefallene Route
- Router startet Holddown Timer und merkt sich die Route als *eventuell unerreichbar*, *belässt sie aber in der Routingtabelle* (mit ihrer ursprünglichen Metrik) und stellt auch Pakete dorthin zu
- während der Holddown-Zeit **ignoriert** der Router alle Updates die eine **gleiche oder schlechtere** Metrik wie die markierte Route haben. Kommt ein Update mit **besserer** Metrik herein, wird die markierte Route wieder aus dem Holddown-Zustand entfernt und auf den aktuellen Stand gebracht.

1.3.2 RIPv1

```
R1(config)#router rip
R1(config-router)#network 141.31.147.0
R1(config-router)#network 192.168.42.0

R1(config-router)#default-information originate  //statisch eingetragenes default-gw propagieren
```

RIPv1 arbeitet ohne Netzmasken → nur 'directly connected' Klasse A,B oder C - Netzadressen dürfen angegeben werden!

1.3.3 RIPv2

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 141.31.147.0    //klasse a,b oder c - netzadresse
R1(config-router)#network 192.168.42.0    //klasse a,b oder c - netzadresse

R2(config)#ip route 192.168.0.0 255.255.0.0 Null0  //alles fuer dieses netz ins 'nichts' routen

R1(config-router)#default-information originate  //statisch eingetragenes default-gw propagieren
```

1.3.4 EIGRP

Eigenschaften

- EIGRP-Packets werden an die Multicast-Adresse 224.0.0.10 versandt
- Anstelle von TCP oder UDP wird RTP, Reliable Transport Protocol verwendet
- Hello-Packets alle 5s
- Hold-Time = 15s; entspricht DeadInterval bei OSPF: wenn 15s lang keine Hello-Packets kommen, gilt die Route als 'down'
- EIGRP verwendet **keine periodischen Updates** sondern Bounded-Partial-Updates.
- EIGRP arbeitet mit dem Diffusing Update Algorithm (DUAL); DUAL berechnet Routen anhand einer **Topologie**- und **Nachbartabelle**.

  Successor : der Nachbarrouter, mit der besten Route ins Ziel; also der 'Next-Hop'; in den Routingtabellen nach Schlüsselwort angegeben **via**

  Feasible Distance : beste Metrik ins Ziel

  $$FD = (10^7/BW + T/10) \cdot 256$$

  BW: kleinste Bandbreite des Pfads in kbit/s

  T: Summer aller Laufzeiten des Pfads in μs

  Reported Distance : die von einem Nachbarouter gemeldete Metrik. Also die FD, die der Nachbar ins Ziel hat und mir meldet.

  Feasible Successor : ein Router, der eine alternative Route ins Ziel hat und der die **Feasibility Condition** erfüllt. Der FS kann sofort, ohne Neuberechnung durch den DUAL-Algo. eine ausgefallene Route ersetzen.

Feasibility Condition : ein Nachbar ist FS, wenn seine RD kleiner als die eigene FD ist ⇒ die Metrik des Nachbarn ins Ziel muss **besser** (kleinerer Wert) sein, als meine eigene Metrik ins Ziel!

EIGRP-Kommandos

```
R1(config)#ip classless
R1(config)#router eigrp 1  //1 ist die AS-nummer
                           // AS-nr muss auf allen routern eines AS gleich sein
R1(config-router)#no auto-summary
R1(config-router)#eigrp log-neighbor-changes
R1(config-router)#metric weights tos k1 k2 k3 k4 k5  //siehe unten
R1(config-router)#redistribute static
R1(config-router)#passive-interface fa0/0  //keine routigupdates ueber dieses I/F versenden

R1(config-router)#network 172.16.0.0    //klasse a,b oder c - netzadresse
R1(config-router)#network 192.168.10.0  //klasse a,b oder c - netzadresse

R1(config-router)#network 172.16.0.0  0.0.0.63   // subnetz .0 --- .64
R1(config-router)#network 192.168.10.8  0.0.0.3  // subnetz .8 --- .11

R1(config-if)#bandwidth 64  //bandbreite fuer metrik in kbit/s
                            //wichtig fuer serielle verbindungen

R1(config)#interface serial 0/0/0  //mehrere Subnetze an R1 als Supernetz über
                                   //dieses I/F propagieren
R1(config-if)#ip summary-address eigrp 10 192.168.0.0 255.255.252.0  //10 = AS-nummer

R1(config-if)#ip hello-interval eigrp 10 60   //60s
R1(config-if)#ip hold-time eigrp 10 180      //180s hold-time > hello-interval !!!!


R1#show ip eigrp neighbors
R1#show ip eigrp neighbors 10         //10 = AS-nummer

R1#show ip eigrp topology
P 172.18.0.0/16, 1 successors, FD is 2172416
         via 209.165.202.134 (2172416/28160), Serial0/0/1
P=passive, keine                FD     RD
  Pfadberechnung aktiv


R1#show ip eigrp topology  all-links
R1#show ip eigrp topology  10         //10 = AS-nummer
R1#show ip eigrp topology  10.1.20.0  //netz-ip

R1#show ip eigrp interfaces 10        //10 = AS-nummer
```

tos = 0 (fest, historisch), k1 = 1 (Bandwidth), k2 = 0 (Load), k3 = 1 (Delay), k4 = 0 (Reliability), k5 = 0 (Reliability)

1.3.5 OSPF

Funktion

- jeder OSPF-Router erzeugt eine **Nachbar-Tabelle**, in der alle seine direkten

Nachbarn eingetragen sind.

- allen Nachbarn werden alle Verbindungen (Links-States) zu den anderen Nachbarn mitgeteilt (mit **Link-State-Packages, LSP**, die Nachricht heisst **Link State Advertisement LSA**); diese werden in den **Link-State-Datenbanken** auf jedem Router gespeichert und an die übernächsten Nachbarn weitergeschickt. Diese speichern wiederum in der LS-DB und schicken das LSP weiter, ...
  Nach einer Weile haben alle Router eine identische Link-State-Datenbank, die die **Topologie des Netzes** beschreibt.
- Aus der Link-State-DB wird mit dem SPF-Algorithmus auf jedem Router zu jedem Ziel im Netz die beste Route berechnet. Dabei werden die Pfadkosten summiert und der Pfad mit den geringsten Kosten als Route gewählt.
- Multicast-Adressen:

  Hello-Packets : 224.0.0.5

  Desginated Router sendet LSA an DROthers : 224.0.0.5

  DROthers senden LSAs an DR und BDR : 224.0.0.6

Intervalle

Hello-Intervall : Takt, in dem Hello-Pakete gesendet werden; 10s bei Ethernet + Serial; 30s bei FrameRelay

Dead-Intervall : Bleiben Hello-Pakete länger als diese Zeit aus, wird der Nachbar als 'down' betrachtet; bei Cisco gilt: $DI = 4 \cdot HI$; also 40s bzw. 120s bei FrameRelay

Cost Berechnung der Link-Kosten aus der Bandbreite in kBit/s (auf Cisco-Routern):

$$\begin{aligned} cost &= 10^8/BW \\ \text{z.B.: } cost &= 10^8/64000 = 1562 \end{aligned}$$

Referenzwert ist hier 100Mbit/s $= 10^8$; dieser kann verändert werden (siehe unten).
Die Link-Kosten können auch direkt gesetzt werden.

Adjacencies Damit zwei Router Nachbarn werden können, muss gelten:

- die Subnetzmasken + Netzadressen der jeweiligen Verbindungen zu den Nachbarn müssen gleich sein
- die Hello-Intervalle müssen gleich sein
- die Dead-Intervalle müssen gleich sein
- die Network-Types müssen gleich sein (Broadcast Multi-Access/ NBMA)
- die Area-IDs müssen gleich sein

network-Kommando

- Beim ersten network-Kommando wir Router-ID festgelegt.
- Jedes Interface, das zur IP im network-Kommando passt, wird freigeschaltet, OSPF-Pakete zu senden und zu empfangen.
- das mit dem network-Kommando angegebene Netz ist in den OSPF-Routing-Updates enthalten.

Auswahl der Router-ID:

1. wird mit `router-id <a.b.c.d>` explizit gesetzt
2. wenn das Kommando `router-id` nicht abgesetzt wird, wählt der Router die **höchste** IP eines seiner Loopback-Interfaces
3. gibt es keine Loopback-Interfaces, wählt der Router die höchste IP seiner konfigurierten Interfaces. Diese müssen up and up sein, aber nicht am OSPF-Routing teilnehmen.

Die Router-ID wird beim Absetzen des ersten `network`-Kommandos festgelegt. Um Änderungen anschliessend wirksam werden zu lassen, muss man `clear ip ospf process` ausführen.
Gibt es doppelt vorkommende Router-IDs, funktioniert OSPF nicht!

Desginated Router und BDR Wahl des DR und BDR:

1. DR wird der Router mit der höchsten Interface-Priority
2. BDR wird der Router mit der zweithöchsten I/F-Priority
3. sind die I/F-Priorities gleich, werden die Router-IDs verglichen

Der default-Wert der Interface-Priority ist 1. Wird der Wert auf 0 gesetzt, kann der Router weder zum DR noch BDR gewählt werden.

OSPF-Kommandobeispiele:

```
R1(config)#interface loopback 0
R1(config-if)#ip address 10.1.1.1 255.255.255.255

R1(config)#router ospf 10  //10 ist prozess-id, 16bit, unabhaengig von AS-nummer!
R1(config-router)#network 10.1.20.32 0.0.0.31 area 0  //area-nr hat 32 bit

R1(config-router)#default-information originate
R1(config-router)#end
R1#show ip route   //die routingtabelle zeigt die default-GWs an:
                   // O*E1 bedeutet externe route ohne kumulierung
                   // O*E2 bedeutet externe route mit kumulierung

R1(config-router)#router-id 10.10.10.10
R1(config-router)#end
R1#clear ip ospf process

R1#(config-if)#bandwidth 64                          //cost = 10^8 / 64000 = 1562
R1#(config-if)#auto-cost reference-bandwidth 1000  //cost-referenz auf 10^9
R1#(config-if)#ip ospf cost 1562                   //linkkosten direkt setzen

R1(config-if)#ip ospf hello-interval 5  //intervall in sekunden
R1(config-if)#ip ospf dead-interval 20  //intervall in sekunden

R1(config-if)#ip ospf priority 10  //default = 1;

R1#show ip protocols              //sehr uebersichtliche ausgabe;
R1#show ip ospf
R1#show ip ospf neighbor
R1#show ip ospf neighbor details
R1#show ip ospf neighbor serial0/0
R1#show ip ospf interface serial 0/0/0
```

1.4 Sicherheit

1.4.1 Passwörter

- CISCO-eigene Verschlüsselung: Type 7
  einschalten mit `service password encryption`
  wirkt sich aus auf:

  enable password

  username *name* password *pwd*

  line und console

- MD5-Verschlüsselung: Type 5
  einschalten, indem man statt `password secret` verwendet. wirkt sich aus auf:
  Alle Passwörter?

```
R1(config)#service password-encryption
R1(config)#security passwords min-length 10
R1(config)#username micha password geheim
```

```
R1(config)#enable password cisco     //niemals enable-passwort mit Type 7 verschluesseln!
//entsprechend mit MD5-Verschluesselug
R1(config)#username micha secret geheim
R1(config)#enable secret cisco
```

1.4.2 Logins verbieten

`no password` + `login` ist die Standardeinstellung der vty-Zugänge. **aux** und **tty** (console) sind jedoch offen.

```
R1(config)#line aux 0
R1(config-line)#no password
R1(config-line)#login
```

1.4.3 vty - Zugriffsprotokolle konfigurieren

```
R1(config)#line vty 0 4
R1(config-line)#no transport input   //erstmal ALLE protokolle verbieten
R1(config-line)#transport input telnet ssh  //ssh und telnet zulassen
```

1.4.4 vty - timeout konfigurieren

```
R1(config)#line vty 0 4
R1(config-line)#exec-timeout 3  //nach 3 minuten aufhaengen
R1(config-line)#exit
R1(config)#service tcp-keepalives-in
```

1.5 ssh

Auf Cisco Routern ist ein ssh-Server und ein ssh-Client installiert.

1.5.1 Konfigurationsschritte

```
Router(config)#hostname R1
R1(config)#ip domain-name wara.de       //zwingend fuer ssh
R1(config)#crypto key generate rsa      //schluesselpaar generieren
                                        //modul-laenge in bit eingeben, z.b. 1024
R1(config)#username micha secret geheim //username generieren
R1(config)#line vty 0 4
R1(config-line)#transport input ssh     //an vty nur ssh erlauben
R1(config-line)#login local             //local -> lokaler benutzer
R1(config-line)#exit
R1(config)#ip ssh time-out 15           //15 sekunden
R1(config)#ip ssh authentication-retries 2
```

1.6 Gefährliche Dienste ausschalten

snmp Version 1 und 2 verwendet Klartext-Strings (evtl. Passwörter) . → ausschliesslich Version 3 verwenden!!

1.6.1 Konfigurationsschritte

```
R1(config)#no cdp run
R1(config)#no ip source-route
R1(config)#no ip classless
R1(config)#no service tcp-small-servers
R1(config)#no service udp-small-servers
R1(config)#no ip finger
R1(config)#no service finger
R1(config)#no ip bootp server
R1(config)#no ip http server
R1(config)#no ip name-server
R1(config)#no snmp-server
R1(config)#no ip directed-broadcast   //gegen smurf-attacken
R1(config)#no ip proxy-arp
R1(config)#no service config
R1(config)#no ip domain-lookup

oder am schnellsten mit

R1(config)#auto secure
```

1.7 Routingprotokolle absichern

1.7.1 Routingupdates auf bestimmte Interfaces beschränken

```
R1(config)#router rip
R1(config-router)#passive-interface default           //keine routing-updates auf
                                                      //irgendwelchen i/f versenden
R1(config-router)#no passive-interface serial 0/0/0   //hier und nur hier wieder erlauben
```

1.7.2 RIPv2 Routingupdates signieren

Signaturen gibt es nur bei RIPv2! RIPv1 kann das nicht!!!

```
R1(config)#key chain RIP_KEY     //es heisst tatsaechlich key chain
                                 //und nicht key-chain oder keychain
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string cisco

R1(config)#interface serial 0/0/0
R1(config-if)#ip rip authentication mode md5
R1(config-if)#ip rip authentication key-chain RIP_KEY
```

1.7.3 EIGRP Routingupdates signieren

Fast identisch zu RIP-Signatur. ABER NUR FAST!

```
R1(config)#key chain EIGRP_KEY
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string cisco

R1(config)#interface serial 0/0/0
R1(config-if)#ip authentication mode eigrp 10 md5   //10 ist die AS-nummer
R1(config-if)#ip authentication key-chain eigrp 10 EIGRP_KEY
```

1.7.4 OSPF Routingupdates signieren

```
R1(config)#interface serial 0/0/0
R1(config-if)#ip ospf message-digest-key 10 md5 cisco  //10 ist die key-id
R1(config-if)#ip ospf authentication message-digest
R1(config-if)#exit
R1(config)#router ospf 15                              //15 ist die ospf-prozess-id
R1(config-router)#network 10.1.1.0 0.0.0.255 area 20
R1(config-router)#area 20 authentication message-digest //
```

1.8 Security Device Manager konfigurieren

Zugang zum SDM = Webanwendung konfigurieren

```
R1(config)#ip http server           //sehr einsichtig! oben haben wir http ausgeschaltet!
R1(config)#ip http secure-server    //https?
R1(config)#ip http authentication local
R1(config)#username micha privilege 15 secret geheim  //geheim wird md5 verscluesselt
                                                      //privilege level 15 = enable-privilegien
R1(config)#line vty 0 4
R1(config-line)#privilege level 15
R1(config-line)#login local
R1(config-line)#transport input telnet ssh
```

1.9 Router Dateisystem

IOS hat so was aehnliches wie Laufwerkskennungen. Es gibt folgende Laufwerke:

ftfp:

flash:

system: entspricht dem RAM

nvram:

Allgemeine Syntax zum copy-Befehl:

```
R1#copy source-url dest-url
```

Beispiele:

```
copy running-config startup-config
copy system:running-config nvram:startup-config

copy nvram:startup-config tftp:

copy flash: tftp:   ios auf tftp-server laden

copy tftp: flash:   ios von tftp-server ins flash laden
```

Hinweis zu tftp: wird keine url (ip des tftp-Servers + Pfad) angegeben, fragt ios interaktiv nach.

1.10 Debugging Kommandos

```
R1(config)#service timestamps debug datetime msec  //zeitstempel fuer die log-eintraege

R1#show processes      //aehnlich ps aux

R1#no debug all  //ich dachte, das heisst undebug all

R1#terminal monitor   //debug output auf aktuelle vty-konsole ausgeben
```

1.11 Passwort wiederherstellen

```
R1#show version  //wert des config-reg. merken

//reboot mit gedrueckter BREAK-taste

rommon1>confreg 0x2142
rommon2>reset
.... //alle fragen mit no beantworten
router>enable
router#copy startup-config running-config  //nicht andersrum!!!!
...  //passwoerter neu konfigurieren
router(config)#config-register 0x2102
router(config)#^Z  // 'end' geht auch
```

1.12 Allgemeines zu ACLs

- jeder Eintrag (= Zeile) in einer ACL enthält eine Bedingung:
  - bei Standard ACLs: **Source**-IP (einzelne IP oder Bereich)

    ⇒ Standard ACLs so nah als möglich Richtung Ziel platzieren
  - bei Extended ACLs: `{ICMP | TCP | UDP | IP}` und Source-IP + Port und Destination-IP + Port

    ⇒ Extende ACLs so nah als möglich Richtung Quelle des zu blockierenden Verkehrs platzieren
- sobald die Quell-IP-Adresse bzw. Quell-/Ziel-IP+Port eines Pakets auf die Bedingung eines Eintrags zutrifft, wird das Paket akzeptiert oder verworfen und die weiteren Bedingungen werden **nicht mehr** untersucht!!!!

1.13 ACL einem Interface hinzufügen

Allgemeine Syntax:

```
R1(config-if)#[no] {ip|ipx} access-group {acl-nr.|acl-name} {in|out}
```

1.14 Standard ACL konfigurieren

Allgemeine Syntax:

```
R1(config)#access-list acl-nr. {deny|permit|remark}
{any | [host] src-ip [src-wildcard]} [log]
```

Das Schlüsselwort `any` steht für 0.0.0.0/0
`any` ersetzt also:
source-ip = 0.0.0.0
source-wildcard = 255.255.255.255

Das Schlüsselwort `host` darf **vor** der Source-IP stehen und ersetzt dann die Source-Wildcard 0.0.0.0
`host` *source-ip* ersetzt also:
source-wildcard = 0.0.0.0

Mit `remark` kann man Kommentare einfügen

1.14.1 Standard ACL (mit Nummer) für IP-Interface

```
R1(config)#no access-list 1   //alles alte loeschen
R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#interface serial 0/0/0
R1(config-if)#ip access-group 1 out
```

1.14.2 Standard ACL für VTY-Zugang

Damit kann man telnet-Zugriff auf Router auf ganz bestimmte Clients festlegen.

```
R1(config)#access-list 21 permit 192.168.10.0 0.0.0.255
R1(config)#access-list 21 permit 192.168.11.0 0.0.0.255
R1(config)#access-list 21 deny any
R1(config)#line vty 0 4
R1(config-line)#login
R1(config-line)#passord secret
R1(config-line)#access-class 21 in
```

1.15 Erweiterte ACL konfigurieren

Allgemeine Syntax:

```
R1(config)#access-list acl-nr. {deny|permit|remark}
{icmp|ip|tcp|udp}
src-ip [src-wildcard] [{lt|gt|eq|neq} port-nr./port-name]
dest-ip [dest-wildcard] [{lt|gt|eq|neq} port-nr./port-name]
[established] [log]
```

Das Schlüsselwort `any` steht für 0.0.0.0/0
`any` ersetzt also:
source-ip = 0.0.0.0
source-wildcard = 255.255.255.255

Das Schlüsselwort `host` darf **vor** der Source-IP stehen und ersetzt dann die Source-Wildcard 0.0.0.0
`host` *source-ip* ersetzt also:
source-wildcard = 0.0.0.0

Mit `remark` kann man Kommentare einfügen

Beispiele Deny FTP:

```
access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 21
access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 20
access-list 101 permit ip any any

interface fa0/0
   ip access-group 101 in
```

Permit HTTP/HTTPS

```
access-list 101 permit tcp 192.168.11.0 0.0.0.255 any eq 80        //any:80 = ziel
access-list 102 permit tcp any 192.168.11.0 0.0.0.255 established  //any  = quelle

interface fa0/0   //fa0/0 internes interface
   ip access-group 101 in
interface fa0/1   //fa0/1 externes interface
   ip access-group 102 in
```

1.15.1 Benannte ACLs

Allgemeine Syntax:

```
R1(config)#ip access-list [standard|extended] name

R1(config-std-nacl)#[deny|permit|remark] src-ip [src-wildcard] [log]  //
    beispiel fuer named-standard-acl

R1(config-if)#ip access-group name [in/out]
```

1.15.2 Beispiel für benannte Standard-ACL

```
R1(config)#ip access-list standard IOSISTBULLSHIT
R1(config-std-nacl)#deny host 192.168.10.5
R1(config-std-nacl)#permit 192.168.11.0 0.0.0.255
R1(config-std-nacl)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip access-group IOSISTBULLSHIT out
```

1.15.3 Dynamische ACL

```
R1(config)#username micha password 0 neu

R1(config)#access-list 101 permit tcp any host 10.2.2.2 eq telnet
R1(config)#access-list 101 dynamic testlist timeout 15
                           permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255

R1(config)#interface serial 0/0/1
R1(config-if)#ip access-group 101 in
```

```
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#autocommand access-enable host timeout 5
```

1.16 DHCP

Server-Port: #67

Client-Port: #68

Dynamic : zeitbegrenzte **Lease**

Manual : Kopplung der IP an die MAC-Adresse des Clients

Automatic : zeitunbegrenzte Lease

Ablauf : client:DHCPDISCOVER
server:DHCPOFFER
client:DHCPREQEST
server:DHCPACK

1.16.1 dhcp-server

Pro Interface, über das Clients Leases beziehen, muss ein DHCP-Pool konfiguriert werden!
Das Interface wird über das dhcp-config-network-Kommando (s.u.) bestimmt: es wird das I/F genommen, dessen IP in diesem Netz

```
R1(dhcp-config)\#network 192.168.10.0 255.255.255.0
```

liegt. Das network-Kommando legt auch die Maske des Pools fest.
Cisco-Routers als dhcp-server konfigurieren:

```
R1(config)#ip dhcp excluded-address 192.168.10.1 192.168.10.9    //von ... bis ausschliessen
R1(config)#ip dhcp excluded-address 192.168.10.254               //genau eine ausschliessen
R1(config)#ip dhcp pool WARAPOOL
R1(dhcp-config)#network 192.168.10.0 255.255.255.0       //offenbar ist hier auch 192.168.10.0 /24
                                                         //leerzeichen vor /24??????
R1(dhcp-config)#default-router 192.168.10.254
R1(dhcp-config)#domain-name wara.de

R1#show ip dhcp binding
R1#show ip dhcp server statistics
R1#show ip dhcp conflict

//tcpdump auf router; access-list dient als filter; funktioniert nicht nur bei dhcp
R1(config)#access-list 100 permit ip host 0.0.0.0 host 255.255.255.255
R1#debug ip packet detail 100

R1#debug ip dhcp server packet
R1#debug ip dhcp server events
```

1.16.2 dhcp-client

Interface eines Cisco-Routers als dhcp-client konfigurieren:

```
R1(config)#interface fa0/0
R1(config-if)#ip address dhcp
R1(config-if)#no shutdown
```

1.16.3 dhcp-relay

Durch Eintragen einer helper-address wird ein Router zum DHCP-Relay-Agent:

> Das Interface, das dem Client am nächsten liegt, bekommt eine helper-address zugewiesen.

Das Relaying gilt somit für folgende Ports:

37 - time

49 - TACACS

53 - dns

67 - dhcp-server

68 - dhcp-client

69 - tftp

137 - NetBios-nameservice

138 - NetBios-datagramservice

Mit `ip forward-protocol` kann man zusätlich weitere Ports, für die Broadcast-pakete weitergeleitet werden sollen, exakt festlegen.

1.16.4 Konfigurationsschritte

```
R1(config)#interface fa0/0
R1(config)#ip helper-address 192.168.11.5
```

1.17 NAT

Begriffe:

inside local address - private IP-Adresse im LAN; das ist **nicht** die IP-Adresse des Routers ins private LAN, sondern die eines beliebigen Hosts im privaten LAN.

inside global address - öffentliche IP-Adresse des Verbindungsrouters ins Internet

outside global address - eine öffentliche IP-Adresse im Internet; z.B. die eines Webservers.

outside local address - öffentliche IP-Adresse eines Hosts im Internet; in den meisten Fällen identisch mit *outside global address*.

Static NAT : 1:1 Übersetzung von **inside-local-adr.** : **inside-global-adr.** → es existiert nur **eine** *inside global address*

Dynamic NAT : *inside global address* wird aus einem Pool genommen

Overloading / PAT : mehrere Clients nutzen die Ports einer *inside global address*; nur bis 4000 Verbindungen sinnvoll

Dynamic NAT und Overloading : Dyn. NAT und PAT kombiniert

1.17.1 statisches NAT konfigurieren

Allgemeine Syntax:

```
R1(config)#ip nat inside source static insideLocal_ip insideGlobal_ip
```

```
R1(config)#ip nat inside source static 192.168.2.101 141.31.147.117
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface serial 0/0/1
R1(config-if)#ip nat outside
```

1.17.2 dynamisches NAT konfigurieren

Allgemeine Syntax:

```
R1(config)#ip nat pool poolName start-ip end-ip netmask maske
R1(config)#ip nat inside source list acl-nr. pool poolName
```

```
R1(config)#ip nat pool poolName WARAPOOL 141.31.147.10 141.31.147.20 netmask 255.255.255.224

R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255   //definieren, welche privaten
                                                        //adressen uebersetzt werden duerfen
R1(config)#ip nat inside source list 1 pool WARAPOOL
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface serial 0/0/1
R1(config-if)#ip nat outside
```

1.17.3 NAT overload / PAT konfigurieren mit einer öffentlichen Adresse

Allgemeine Syntax:

```
R1(config)#ip nat inside source list acl-nr. interface if overload
```

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255   //definieren, welche privaten
                                                         //adressen uebersetzt werden duerfen
R1(config)#ip nat inside source list 1 interface serial 0/0/1 overload
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface serial 0/0/1
R1(config-if)#ip nat outside
```

1.17.4 NAT overload / PAT konfigurieren mit mehreren öffentlichen Adressen

Diese Konfiguration scheint bis auf das zusätzliche Schlüsselwort `overload` identisch zu sein mit der von 'Dynamisches NAT'.

```
R1(config)#ip nat pool poolName WARAPOOL 141.31.147.10 141.31.147.20 netmask 255.255.255.224

R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255   //definieren, welche privaten
                                                         //adressen uebersetzt werden duerfen
R1(config)#ip nat inside source list 1 pool WARAPOOL overload
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface serial 0/0/1
R1(config-if)#ip nat outside
```

1.17.5 NAT Fehlersuche

Allgemeine Syntax:

```
R1(config)#clear ip nat translation inside global-ip local-ip
           [outside local-ip gobal-ip]

R1(config)#clear ip nat translation protokoll
            inside global-ip global-port local-ip local-port
           [outside local-ip local-port gobal-ip global-port]
```

```
R1#show ip nat translations
R1#show ip nat translations verbose
R1#show ip nat statistics
R1#clear ip nat translation *

R1#debug ip nat   //zeigt die uebersetzungsvorgaenge an
```

1.18 IPv6

EUI-64 Standard

1.18.1 IPv6-Forwarding einschalten und Adressen zuweisen

```
R1(config)#ipv6 unicast-routing   //ipv6 forwarding einschalten
```

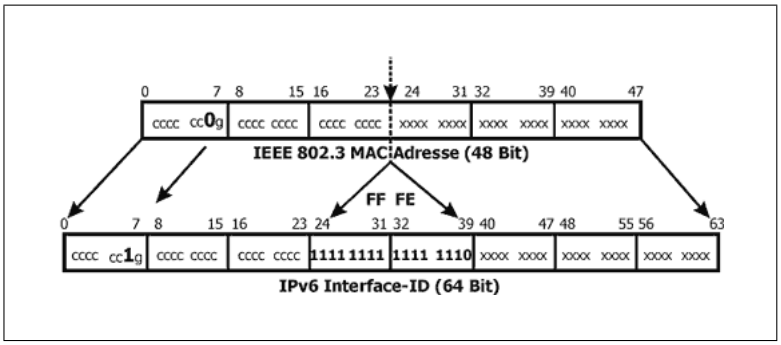


Abbildung 1: Umrechnung einr MAC- in eine EUI-64-Adresse

```
R1(config)#interface fa 0/0
R1(config-if)#ipv6 address 2001:db8:2222:7272::72/64
R1(config-if)#ipv6 address 2001:db8:2222:7272::/64 eui-64  //die letzten 64 bit werden aus
                                                           //der MAC-Adresse gebildet
```

1.18.2 IPv6 im Dual-Stack-Betrieb

```
R1(config)#interface fa 0/0
R1(config-if)#ip address 192.168.99.1 255.255.255.0
R1(config-if)#ipv6 address 3ffe:b00:c18:1::3/127          //ein netz mit nur 2 adressen?
```

1.18.3 IPv6 Namensauflösung

Bei statischer Namenszuordnung können bis zu 4 IPv6-Adressen mit einem Hostnamen verknüpft werden.

Oder man gibt bis zu 6 DNS-Server an.

Allgemeine Syntax (statische Namenszuordnung):

```
R1(config)#ipv6 host name [port] ipv6adr [{ipv6adr}]
```

```
R1(config)#ipv6 host alfred 3ffe:b00:ffff:b::1

//oder mit dns-server

R1(config)#ip name-server 3ffe:b00:ffff:1::10
```

1.18.4 RIPng konfigurieren

Hier gibt es kein `network`-Kommando mehr. Stattdessen wird RIP für jedes Interface, das in den Routing-Updates enthalten sein soll, extra eingeschaltet.

```
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router rip RIPPCHEN  //RIPPCHEN ist der names des ripng-prozesses

R1(config)#interface fa0/0
```

```
R1(config-if)#ipv6 address 2001:db8:1:1::/64 eui-64
R1(config-if)#ipv6 rip RIPPCHEN enable   //ripng fuer dieses i/f einschalten
```

1.18.5 RIPng, diverse show- und debug-befehle

```
R1#show ipv6 interface
R1#show ipv6 interface brief
R1#show ipv6 neighbors
R1#show ipv6 protocols
R1#show ipv6 rip
R1#show ipv6 route
R1#show ipv6 route summary
R1#show ipv6 routers
R1#show ipv6 static
R1#show ipv6 static 2001:db8:5555:0/16
R1#show ipv6 static interface s0/0/0
R1#show ipv6 static detail

R1#clear ipv6 rip
R1#clear ipv6 route *
R1#clear ipv6 route 2001:db8:c18:3::/64
R1#clear ipv6 traffic
R1#debug ipv6 packet
R1#debug ipv6 rip
R1#debug ipv6 routing
```

2 Switching

2.1 Grundbegriffe

2.1.1 Hierarchical Network

Access : Switches mit VLAN-Support, PoE, Port Security; Stellt Switchports für Hosts bereit → Interface to End Devices.

> Die Access-Schicht verbindet **End**-Geräte mit dem Netzwerk und kontrolliert, welche Geräte miteinander über das Netz kommunizieren dürfen.

Distribution : Switches mit Layer3-Support, ACLs

- VLANS, Subnetze, Inter-VLAN-Routing
- Redundante Verbindungen
- ACLs

Core : Zusammenfassen des Datenverkehrs vom Distribution-Layer: schnelle Switches (very high forwarding rate); Core-Layer = High-Speed-Backbone, nur schnelles Switching, kein Routing, keine ACLs

2.1.2 Converged Network

Data, Voice & Video

2.1.3 Kollisions- und Broadcastdomänen

Collision Domain :

> **jeder beschaltete** Switchport stellt eine Kollisionsdomäne dar.

Broadcast Domain :

> Switches filtern Broadcasts **nicht**; Eine Ansammlung untereinander verbundener Switches stellt eine Broadcastdomände dar. Router trennen BC-Domains ⇒ VLANs und Subnetze bilden jeweils eine BC-Domain.

2.2 Grundeinstellungen

2.2.1 Start eines Switchs

Power on Self-Test und System-LED

Syst-LED blinkt : `wärend` des POST

Syst-LED leuchtet grün : POST erfolgreich

Syst-LED leuchtet gelb : POST durchgefallen

2.2.2 Konsolenmeldungen synchronisieren

```
S1(config)#line console 0
S1(config-line)#logging synchronous
S1(config-line)#end
```

2.2.3 IOS History Einstellungen

```
S1#terminal history size 20 // size 20 ist optional, default = 10
S1#terminal no history      // history ausschalten
S1#show history
```

2.2.4 Port Einstellungen

```
S1(config)#interface fa0/1
S1(config-if)#mdix auto
S1(config-if)#duplex auto
S1(config-if)#speed auto
S1(config-if)#end
```

2.2.5 Management-VLAN, Management IP zuweisen

Für Fernwartung benötigt der Switch eine IP-Adresse, eine Maske und ein Default-GW. Diese werden einem **virtuellen Interface** zugewiesen. Das virtuelle Interface ist wiederum ein VLAN, **dem ein passender Switch-Port zugewiesen werden muss**!

> Bei einem 24-Port-Switch kann man das VLAN-Interface wie einen **25. logischen Port** betrachten. Dieser liegt immer im Management-VLAN.

VLAN-Interface :

- auf einem Layer2-Switch gibt es nur **genau ein** aktives Layer3-VLAN-Interface
- nur das Management-VLAN hat ein Layer3-VLAN-Interface

Default-Einstellung : in der Werkseinstellung ist VLAN 1 das Management-VLAN
⇒ unbedingt ändern

```
S1(config)#interface vlan 99
S1(config-if)#ip address 192.168.10.20 255.255.255.240
S1(config-if)#no shutdown
S1(config-if)#exit
S1(config)#interface fa 0/18
S1(config-if)#switchport mode access
S1(config-if)#switchport access vlan 99
S1(config-if)#end
S1#show ip interface brief
S1#configure terminal
S1(config)#ip default-gateway 10.0.0.1  //default-gw konfigurieren
```

2.2.6 httpd auf Switch konfigurieren

```
S1(config)#ip http authentication [enable | local | tacacs ]
S1(config)#ip http server
```

2.2.7 ssh und telnet auf Switch aktivieren

```
Switch(config)#hostname S1
S1(config)#ip domain-name wara.de       //zwingend fuer ssh
S1(config)#crypto key generate rsa      //schluesselpaar generieren
                                        //modul-laenge in bit eingeben, z.b. 1024
S1(config)#ip ssh version 2
S1(config)#username micha secret geheim //username generieren
S1(config)#line vty 0 4
S1(config-line)#transport input ssh     //an vty nur ssh erlauben
S1(config-line)#login local             //local -> lokaler benutzer
S1(config-line)#exit
S1(config)#ip ssh time-out 15           //15 sekunden
S1(config)#ip ssh authentication-retries 2


S1(config)#line vty 0 4
S1(config-line)#transport input telnet  //falls man telnet moechte

S1(config)#crypto key zeroize rsa       //rsa-schluessel wieder loeschen
```

2.2.8 Statischer Eintrag in MAC-Adress-Tabelle

```
S1(config)#mac-address-table static 0030.656a.971a vlan 10 interface fa0/7
S1(config)#end
S1#show mac-address-table
```

2.3 Port-Security konfigurieren

Verschiedene Typen sicherer MAC-Adressen

static secure MAC-Addr. : manuell konfigurieren (s.u.)

dynamic secure MAC-Addr. : werden dynamisch gelernt und in Adresstabelle des Switches gespeichert; sind nach Neustart weg

sticky secure MAC-Addr. : mit `switchport port-security mac-address sticky` wird das **Sticky Learning** eingeschaltet:

- Wenn SL eingeschaltet wird, werden auch vorher gelernte dynamic secure MAC-Adr. und alle die ab jetzt gelernt werden in der **running-config** gespeichert; sichern mit `copy run start`
- Wird SL wieder ausgeschaltet, werden die Sticky MAC-Adr. wieder aus der running-config entfernt, bleiben aber als dyn. secure MAC-Adr. in der MAC-Adress-Tabelle des Switchs.

Port Security kann nur auf Static Access Ports konfiguriert werden! Nicht auf Dynamic Access Ports oder Trunk Ports!

2.3.1 Violation Settings

Shutdown	Restrict	Protect
Grundeinstellung. Port fährt selbstständig runter	Port blockiert, Eintrag in Logdatei, Violation-Zähler erhöhen	Port blockiert lediglich

2.3.2 Static Secure MAC-Address

```
S1(config)#interface fa0/7
S1(config-if)#switchport mode access
S1(config-if)#switchport port-security          //port-security generell einschalten
S1(config-if)#switchport port-security mac-address 0030.656a.971a
S1(config-if)#switchport port-security violation shutdown //default
S1(config-if)#switchport port-security violation restrict //blockieren + syslog + counter
S1(config-if)#switchport port-security violation protect  //nur blockieren
S1(config-if)#end
S1#show port-security address
```

2.3.3 Dynamic Secure MAC-Address

```
S1(config)#interface fa0/7
S1(config-if)#switchport mode access
S1(config-if)#switchport port-security             //port-security generell einschalten
S1(config-if)#switchport port-security maximum 5   //5 adressen erlauben
```

Static und Dynamic kann kombiniert werden: z.b. maximum auf 10 setzen und 5 Adressen statisch vergeben, der Rest wird dann dynamisch gelernt.

2.3.4 Sticky Secure MAC-Address

```
S1(config)#interface fa0/7
S1(config-if)#switchport mode access
S1(config-if)#switchport port-security             //port-security generell einschalten
S1(config-if)#switchport port-security maximum 5   //5 adressen erlauben
S1(config-if)#switchport port-security mac-address sticky   //dyn. adressen in sticky wandeln
```

```
S1(config-if)#end
S1#show port-security address
```

2.4 VLANs

Def. von CISCO : Ein VLAN ist ein logisch abgeteiltes IP-Subnetz.

Def. von Dienert : Ein VLAN ist eine Broadcastdomäne

Cisco unterscheidet zwischen folgenden VLAN-Typen: **Data, Default, Management, Native**

Data : Im Data-VLAN wird nur Datenverkehr transportiert, der von den Endgeräten der Bemutzer erzeugt wurde. Dazu gehört kein Datenverkehr von VoIP-Geräten! Dieser wird in einem extra VLAN verteilt.

Default VLAN : fest unveränderlich; bei einem unkonfigurierten Switch sind alle Ports im VLAN-1; **CDP und STP Kontroll-Verkehr wird immer im VLAN-1** übertragen.
Cisco empfiehlt, ein anderes VLAN anstelle von VLAN-1 als Default-VLAN zu konfigurieren. Das bedingt, dass alle Ports, die im VLAN-1 sind umkonfiguriert werden. Der Begriff **Default-VLAN** bezieht sich dann auf ein anderes VLAN als VLAN-1. Die Rolle von VLAN-1 beschränkt sich in diesem Fall auf den Transport von CDP-, STP-, usw. -Frames.

Management VLAN : IP und Maske zuweisen; **nicht VLAN-1 nehmen (das ist die Werkseinstellung)**, sondern z.B. VLAN-99

Native VLAN : für Abwärtskompatibilität mit Nicht-dot1q-Geräten; wird einem 802.1Q-Trunk-Port zugeordnet. Der Trunk-Port platziert dann alle untagged-Frames im Native VLAN → der fehlende Tag stellt einen virtuellen Tag im Native VLAN dar.
Merksatz aus einem HP-Handbuch (gilt der auch für cisco?):

> A port can be a member of one untagged VLAN. All other VLAN assignments for that port must be tagged

Voice-VLAN : z.B. VLAN-150, spezielles VLAN, Verkehr in diesem VLAN wird bevorzugt (QOS). Am Switchport muss ein Cisco-IP-Phone angeschlossen werden. Das enthält einen eingebauten 3-Port-Switch, der Voice- und Data getrennt weiterleitet.

Trunk : Trunks übertragen die Frames mehrerer VLANs über eine **einzelne** physikalische Verbindung

VLAN-id :

VLAN-id = 1 : Default-VLAN s.u.

VLAN-id = 2 . . . 1001 : Normaler Bereich

VLAN-id = 1002 . . . 1005 : token-ring, fddi

2.4.1 Einen Untagged-Port konfigurieren (Access-Port)

```
S1(config)#interface fa0/1
S1(config-if)#switchport mode access     // ohne wenn und aber in access-mode setzen
S1(config-if)#switchport access vlan 55  // dieses I/F dem vlan 55 zuordnen
S1(config-if)#switchport access vlan 10  // so kann man den port direkt umbuchen
S1(config-if)#end
S1#show vlan brief                       //zeigt schoene vlan-uebersicht
```

2.4.2 Einen Access-Port mit VoIP konfigurieren

```
S1(config)#interface fa0/1
S1(config-if)#mls qos trust cos
S1(config-if)#switchport voice vlan 150  //immer 150 nehmen
S1(config-if)#switchport mode access
S1(config-if)#switchport access vlan 55
S1(config-if)#end
```

2.4.3 Einen Tagged-Port konfigurieren (Trunk-Port)

```
S1(config)#interface fa0/1
S1(config-if)#switchport mode trunk     // ohne wenn und aber in trunk-mode setzen
S1(config-if)#switchport trunk allowed vlan 10,20,30  //OHNE ADD!!!!
S1(config-if)#switchport trunk allowed vlan add 40 //noch eins hinzufuegen
S1(config-if)#switchport trunk allowed vlan all  // alle erlauben! ohne add!!!!
S1(config-if)#switchport trunk native vlan 99    //sehr wichtig
S1(config-if)#end
S1#show interfaces trunk                //zeigt alle I/Fs, die im trunk-mode sind
S1#show interfaces fa0/7 switchport     //detaillierte ausgabe
```

2.4.4 Weitere Switchport-Modi

```
S1(config)#interface fa0/1
S1(config-if)#switchport mode dynamic auto      //s.u.
S1(config-if)#switchport mode dynamic desirable //s.u.
S1(config-if)#switchport nonegotiate            //DTP an diesem switchport abschalten
```

Welche Switchport-Modi werden durch DTP (Dynamic Trunking Protocol) eingestellt:

Dynamic Auto : Port sendet nur aus, dass er einen Trunk bilden **kann**, er sendet aber keine Requests, dass er einen Trunk bilden **möchte**.

Dynamic Desirable : Port sendet aus, dass er eine Trunk bilden kann und bittet den entfernten Port, in den Trunking State zu wechseln. Desirable ⇒ Port wünscht sich Trunking State

	dyn. Auto	dyn. Desi.	Trunk	Access
dyn. Auto	**access**	trunk	trunk	access
dyn. Desi.	trunk	**trunk**	trunk	access
Trunk	trunk	trunk	trunk	vermeiden!
Access	access	access	vermeiden!	access

2.4.5 Einen Trunk wieder entfernen

```
S1(config)#interface fa0/1
S1(config-if)#no switchport trunk allowed vlan   //alle vlans entfernen
S1(config-if)#no switchport trunk native vlan    //ANSCHLIESSEND port ins vlan 1 bringen
S1(config-if)#end
```

2.4.6 Einen Trunk-Port in den Access-Mode bringen

```
S1(config)#interface fa0/1
S1(config-if)#switchport mode access            //port von trunk auf access umkonfigurieren
S1(config-if)#end
```

2.5 VLAN-Fehlersuche

Typische Fehler, von oben nach unten zu überprüfen:

Native VLAN mismatch : die Trunk-Ports sind mit verschiedenen native VLANs konfiguriert. Abhilfe, z.b. von Native Vlan 100 nach 99 bringen:

```
S1(config-if)#switchport mode trunk native vlan 99
```

Trunk Mode mismatch : z.B. ein Port Trunk Mode 'off', anderer Port Trunk Mode 'on' ⇒ Tabelle in Kap. 2.4.4 beachten

VLANs and IP-Subnets : korrekte IP-Adressierung überprüfen: ein VLAN ist ein logisch separiertes IP-Subnetz

Allowed VLANs on trunks : nicht alle benötigten VLANs sind auf dem Trunk erlaubt ⇒ fehlende erlauben:

```
S1(config-if)switchport trunk allowed vlan 10,20,99
```

2.6 VLAN Trunking-Protocol

VTP-Domain : vebundene Switches, die **denselben VTP-Domain-Name haben**; begrenzt durch Router/Layer-3-Switch

VTP-Domain-Name : wird vom Server an Clients verteilt, wenn diese noch nicht einer anderen Domäne angehören

VTP : arbeitet mit 802.1Q Ethernet-**Multicast**-Frames
⇒ VTP ist ein Layer-2-Protokoll
⇒ VTP arbeitet nur auf Trunks !!!
die VTP-Version muss auf allen Switches der Domäne gleich sein! Default-Version = Version 1

Revision-Number : 32 bit, Default = 0, wird pro hinzugefügtes oder entferntes VLAN und bei VLAN-Namensänderung um 1 erhöht;

VTP-Pruning : restricts flooded traffic to those trunks, that are necessary to reach the destination

VTP-Modes :

- Server
  - Server-Mode ist die Default-Einstellung
  - Es sind mehrere Server pro Domäne möglich, aber ein Server kann nur in *einer* Domäne sein!
  - Server kann VLANs erzeugen, löschen und umbenennen
  - VLAN-Info für ganze Domäne wird im NVRAM = Startup-Config gespeichert
- Client
  - Switch muss *extra* als Client konfiguriert werden!
  - Client kann keine VLANs erzeugen, löschen, umbenennen
  - VLAN-Info steht nur im RAM = Running-Config
- Transparent
  - leitet nur VTP-Advertisments weiter
  - nimmt nicht teil
  - erzeugen, löschen, umbenennen von VLANs nur *lokal*

2.6.1 VTP konfigurieren

Zu beachten:

- alle Switches in Default-Zustand zurücksetzen; Default-Zustand überprüfen!

Grund: trifft ein VTP-Advertisment mit einer höheren Revision-Number als die eigene auf einem Switch ein, übernimmt der Switch die VLAN-Informationen aus diesem Advertisment. Das betrifft vor allem Switches, die zuvor schon in einer anderen VTP-Domäne betrieben wurden.

  Zurücksetzen der Rev.Nr. durch hin und her Ändern des VTP-Domain-Namens:

  ```
  S1(config)#vtp domain foo
  S1(config)#vtp domain cisco1
  ```

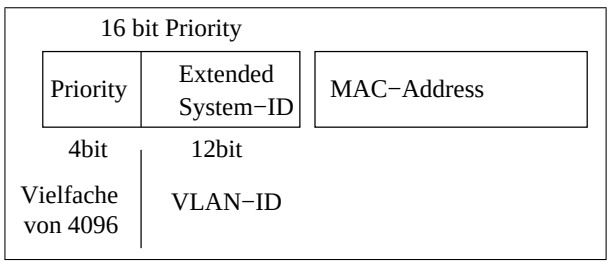
- mindestens 2 Switches als VTP-Server konfigurieren; überall gleiche VTP-Vers.-Nr.!
  - VTP-Domain auf dem ersten Server einrichten
  - VLANs und Trunk-Ports konfigurieren
- restliche Switches in Client-Mode bringen
  - Trunks konfigurieren (VTP arbeitet nur über Trunks!)
  - mit VTP-Server verbinden
  - VTP-Status überprüfen und anschliessend Access-Ports konfigurieren

```
S1(config)#vtp version 1       // auch version 2 moeglich
S1(config)#vtp domain XYZ      //domain-name auf XYZ setzen
S1(config)#vtp password xxyyzz //passwort auf xxyyzz setzen
S1(config)#vtp mode server     //das ist default
S1(config)#vtp mode client
S1(config)#vtp mode transparent
S1(config)#vtp pruning  //pruning (beschneiden) in der domaene erlauben, nur auf server absetzen
S1(config)#end
S1#show vtp status
```

2.7 Spanning Tree

BPDU : Bridge Protocol Data Unit; werden alle 2s versandt (Hello-Timer)

Bridge-ID :



Die Bridge mit der **kleinsten** Bridge-ID wird Root-Bridge.

Network-Diameter : Da STP in der Default-Einstellung mit einem Network-Diameter von 7 “Hops” arbeitet, ist nach 14 s die Root-Bridge bestimmt.

Max-Age : empfängt ein Switch 20s lang keine BPDUs (also 10 aufeinanderfolgende BPDUs), wird ein neuer Root-Election-Prozess gestartet.

Forward-Delay : Zeit, die im Listening und Learning Status verbracht wird

Port-States :

STP	RSTP
Blocking (20s)	
Listening (15s)	Discarding (Schleifen auflösen)
Learning (15s)	Learning (MAC-Adr. Tabelle füllen
Forwarding	Forwarding
Disabled	Discarding

Pfadkosten

Strecke	Kosten
10Gbit/s	2
1Gbit/s	4
100MBit/s	19
10 MBit/s	100

Port-Rollen :

Root-Ports :

- Root-Ports sind der Root-Bridge am nächsten, d.h. haben kleinste Pfadkosten Richtung Root-Bridge
- Nur ein Port pro Switch kann Root-Port sein; haben mehrere Ports gleiche Pfadkosten, entscheidet die Port-Priority
- Die Root-Bridge selbst hat nur Designated-Ports

Designated und Non-Designated Ports :

- Jedes Segement = Verbindung Switch-Switch hat genau einen **Designated Port**,der andere Port wird **Non-Designated-Port**, wenn die Verbindung nicht Richtung Root-Bridge weist (also sozusagen zwei Äste des Baums **quer** verbindet).
- Dabei wird wieder über die Wegekosten entschieden: Der Port mit den geringsten Wegkosten Richtung Root-Bridge wird Designated Port.
- Bei gleichen Wegkosten wird der Port auf dem Switch mit der kleineren Bridge-ID zum Designated Port.

Edge-Ports :

- entspricht Portfast; Edge-Ports gehen sofort in den Forwarding-State
- Edge-Ports werden zu normalen ST-Ports, sobald sie ein BPDU empfangen

Alternate-Ports (nur bei RSTP) :

- Entspricht Non-Designated-Port einer "Querverbindung" (s.o.) bei STP; im Normalfall im Discarding-State
- Wird zu einem Designated-Port und wechselt in den Forwarding-State, wenn der Pfad über den Designated-Port ausfällt.

Backup-Port (nur bei RSTP) :

- Nur bei redundanten (parallelen) Verbindungen; Normalszustand: Discarding
- Wechselt nach Forwarding, wenn paralleler Pfad ausfällt

Link-Types :

- Point-to-Point : Full-Duplex; nur Point-to-Point Designated Ports können bei RSTP den Schnellübergang Discarding $\rightarrow$ Forwarding machen.
- Shared Link : Half-Duplex, z.B. Verbindung zu einem Hub.

2.7.1 Bridge Priority setzen

```
S1(config)#spanning-tree vlan 10 priority 24576 //muss vielfaches von 4096 sein
S1(config)#spanning-tree vlan 10 root primary  //24576 oder kleiner
```

```
S1(config)#spanning-tree vlan 10 root secondary  //28672 oder kleiner
S1(config)#end
S1#show spanning-tree
```

2.7.2 RPVST+ einschalten

```
S1(config)#spanning-tree mode rapid-pvst
S1(config)#end
S1#clear spanning-tree detected-protocols
```

2.7.3 Edge-Port konfigurieren

```
S1(config)#interface fa0/7
S1(config-if)#spanning-tree portfast
```

2.7.4 Link-Types konfigurieren

```
S1(config)#interface fa0/7
S1(config-if)#spanning-tree link-type point-to-point  //full-duplex
S1(config-if)#spanning-tree link-type shared  //half-duplex, d.h. am port haengt ein z.b. hub
```

2.7.5 Port-Priority setzen

```
S1(config)#interface fa0/7
S1(config-if)#spanning-tree port-priority 64  // 0-240, 16er schritte; KLEINER IST BESSER!
```

2.8 Inter VLAN-Routing

2.8.1 Subinterfaces

Subinterfaces entsprechen virtuellen Interfaces unter Linux.
Beispiel für eine Konfiguration von Subinterfaces auf dem Router R1. Die im Beispiel verwendeten VLANs müssen vorher eingerichtet worden sein:

```
R1(config)#interface fa0/1.10            // .10 = bezeichner des sub-I/F, frei waehlbar
R1(config-subif)#encapsulation dot1q 10  // 10 = VLAN-ID
R1(config-subif)#ip address 172.17.10.1 255.255.255.0
R1(config-subif)#exit
R1(config)#interface fa0/1.20
R1(config-subif)#encapsulation dot1q 20  // 20 = VLAN-ID
R1(config-subif)#ip address 172.17.20.1 255.255.255.0
R1(config-subif)#exit
R1(config)#interface fa0/1.99
R1(config-subif)#encapsulation dot1q 99 native // 99 = VLAN-ID des Native VLAN
R1(config-subif)#ip address 172.17.99.1 255.255.255.0
R1(config-subif)#exit
R1(config)#interface fa0/1
R1(config-if)#no shutdown                //unbedingt beachten: das phy. I/F hochfahren!!
```

3 Wireless LANs

3.1 802.11 WLAN-Uebertragungsstandards

Protokoll	veröffent-licht	Frequenz	Durchsatz (netto)	Datenrate (brutto)	Multiplex-verfahren	Reichweite (im Haus, abhängig von Wänden)	Reichweite (Radius im Freien, inkl. einer Wand)
802.11	1997	2.4 GHz	0.9 Mbit/s	2 Mbit/s		ca. 20m	ca. 100m
802.11a	1999	5 GHz	23 Mbit/s	54 Mbit/s	OFDM	ca. 35m	ca. 120m
802.11b	1999	2.4 GHz	4.3 Mbit/s	11 Mbit/s	DSSS	ca. 38m	ca. 140m
802.11g	2003	2.4 GHz	19 Mbit/s	54 Mbit/s	OFDM	ca. 38m	ca. 140m
802.11n	Juni 2009 (geschätzt)	2.4 GHz ... 5 GHz	74 Mbit/s	248 Mbit/s	MIMO	ca. 70m	ca. 250m
802.11y	Juni 2008 (geschätzt)	3.7 GHz	23 Mbit/s	54 Mbit/s		ca. 50m	ca. 5000m

OFDM Orthogonal Frequency Division Multiplex

DSSS Direct Sequence Spread Spectrum

MIMO Multiple Input Multiple Output : mehrere Sender, Empfänger, Antennen gleichzeitig, Mehrwegeausbreitung

4 WAN-Protokolle

4.1 HDLC

4.1.1 HDLC einschalten

```
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation HDLC
```

4.2 PPP

4.2.1 PPP einschalten

```
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation ppp
```

4.2.2 PPP Authenticaton konfigurieren

Optionen:

`chap`

`pap`

`chap pap` - beide einschalten, zuerst chap ausführen

`pap chap` - beide einschalten, zuerst pap ausführen

`if-needed` - siehe TACACS/XTACACS

***list-name* — `default`** - siehe AAA/TACACS+

`callin` - Authentication nur bei eintreffenden Anrufen

AAA/TACACS+ AAA/TACACS is a dedicated server used to authenticate users.
aaa= authentication, authorisation, accounting

TACACS/XTACACS wird nicht mehr verwendet

Nachfolger: TACACS+ und RADIUS

```
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation ppp
R1(config)#ppp authentication {chap | pap | chap pap | pap chap}
               [if-needed] [<list-name> | default] [callin]
```

4.2.3 PPP Komprimierung einschalten

Optionen:

`predictor` - Prädiktions-(Vorhersage-)Komprimierung verwenden

`stac` - Stacker-Algorithmus verwenden → LZS = Lempel-Ziv patentiert durch Fa. Stac; Programmname: *Stacker*

```
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation ppp
R1(config-if)#compress [predictor | stac]
```

4.2.4 PPP Link Quality Monitoring / Error Detection

Die Link-Quality-Schwelle in Prozent festlegen:

```
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp quality 80  (Use the no ppp quality command to disable LQM.)
```

4.2.5 PPP Multi-Link Load-Balancing

Multi-Link PPP (MPPP) einschalten.

```
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp multilink (Use the no ppp multilink command to disable MPPP)
```

4.2.6 PPP Callback

```
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp callback [accept | request]
```

4.2.7 Fehlersuche PPP

```
R1#debug ppp
R1#undebug all
```

4.3 Diagnose

4.3.1 Status eines Serial -I/F überprüfen

```
R1#show interfaces serial 0/0/0
R1#show controllers serial 0/0/0
R1#debug ppp { packet | negotiation | error | authentication | compression | cbcp }
R1# no debug ppp { ... }
R1#undebug all
```

4.4 Frame Relay

4.4.1 Grundkonfiguration

```
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#encapsulation frame-relay [ciso | ietf]
R1(config-if)#bandwidth 64      //bandbreite in kbit/s; fuer eigrp/ospf -> metrik
R1(config-if)#no shutdown
```

4.4.2 Dynamic Address-Mapping mit Inverse ARP

```
R1#show frame-relay map
```

4.4.3 Static Address-Mapping

Allgemeine Syntax:

```
R1(config)#frame-relay map {ip|ipx|appletalk} layer3Addr dlci
            [broadcast] [ietf]|[cisco]
```

```
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#encapsulation frame-relay
R1(config-if)#bandwidth 64
R1(config-if)#no frame-relay inverse-arp
R1(config-if)#frame-relay map ip 10.1.1.1 102 broadcast  //broadcast: erleichtert OSPF-config
R1(config-if)#no shutdown
```

4.4.4 Local Management Interface LMI

LMI-Typen

1. cisco
2. ansi
3. q933a

Hauptfunktionen von LMI:

Keepalives : Hauptaufgabe von LMI ist die Aufrechterhaltung der Verbindung (PVC) mit Keepalives

Statusermittlung : Austausch von Statusinformationen mit allen DLCIs, die der Switch/Router (Switch in der Wolke, lokal: der Router) kennt; d.h. es werden die Betriebszustände der verschiedenen PVCs (die der Router kennt) ermittelt.

Verfügbare PVCs : LMI teilt dem Router mit, welche PVCs verfügbar sind

Flow Control : optionale Funktion; XON/XOFF - Flusskontrolle

Multicast-Adressierung : optionale Funktion; erlaubt die Verwendung einiger DLCIs

Globale DLCIs : optionale Funktion von LMI als Multicast-Adressen

```
R1#(config-if)#frame-relay lmi-type [cisco | ansi | q933a]
R1#(config-if)#keepalive <interval>
R1#show frame-relay lmi
```

4.4.5 Subinterfaces

Allgemeine Syntax:

```
R1(config-if)#interface ifNummer.subifNummer {point-to-point |
    multipoint}
...
R1(config-subif)#frame-relay interface-dlci dlciNr
```

> Die Subinterface-Nummber darf im Bereich 1 bis $2^{32} - 1$ liegen. Sie sollte der Einfachheit wegen aber den gleichen Wert haben wie die DLCI-Nummer.

```
R1(config)#interface serial 0/0/0
R1(config-if)#no ip address
R1(config-if)#encapsulation frame-relay
R1(config-if)#no shutdown
R1(config-if)#interface serial 0/0/0.103 point-to-point
R1(config-subif)#ip address 10.1.1.1 255.255.255.252
R1(config-subif)#bandwidth 64
R1(config-subif)#frame-relay interface-dlci 103
```

4.4.6 Fehlersuche / Überprüfung

```
R1#show interface serial 0/0/1
R1#show frame-relay lmi
R1#show frame-relay pvc 102
R1#show frame-relay map
R1#clear frame-relay inarp
R1#debug frame-relay lmi
```