

1 VLAN Trunking Protocol VTP

1.1 Nutzen & Begriffe

- Zentralisierung des VLAN Management
- Schnelles Einfügen eines zusätzlichen VLANs
- Konsistenz über ganzes Netz
- VTP-Domain: Gruppe von Switches, die VTP untereinander ausführen
S(config)#vtp domain DOMAIN_NAME.
- VTP Advertisements: VTP-Infos werden über trunk Ports verteilt (Layer-2)
- VTP Modi (def. server): *S(config)#vtp mode [server | client | transparent]*
- VTP Pruning: sorgt dafür, dass Advertisements nicht auf jedes IF geflutet werden.

1.2 VTP Funktionen

Praxis: Erst alle Switches konfigurieren, die NICHT VTP-Server sein sollen

1.2.1 VTP Domain

- Verteilung durch Server
- **Achtung:** Verteilte Namen können nicht mehr einfach so überschrieben werden!
- Vorsicht beim Einbau eines neuen Switches (Verbreitung von falschen Infos)

1.2.2 VTP advertisements

3 Arten von Advertisements:

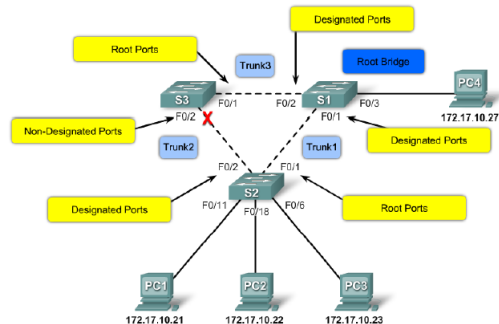
- Summary Advertisements: alle 5 Minuten mit aktueller Config
- Subset Advertisements: Änderungen von VLANs werden bekannt gegeben
- Request Advertisements: Client an Server, danach Summary Advertisement

1.2.3 VTP Konfiguration

Server	Client
<i>S#sh vtp status</i> <i>S(config)#vtp domain myName</i> <i>S(config)#vtp version 2</i>	<i>S#sh vtp status</i> <i>S(config)#vtp mode client</i> <i>S(config)#vtp version 2</i> <i>S(config)#vtp password myPW</i>

2 Spanning Tree Protocol STP

2.1 Der ST Algorithmus



Ablauf:

- 1.) Bestimmung Root Bridge (RB) (tiefste Bridge ID [Standard: MAC Adresse]).
- 2.) Bestimmung der "root ports" (kürzester Weg zur Root Bridge [Link Metrik]).
- 3.) Auf jedem LAN-Segment: Bestimmung des "designated ports".
- 4.) "non-designated ports" (weder root noch designated ports) werden blockiert.

2.2 Bridge ID

- Bridge Priority (2 Bytes) + MAC Address (6 Bytes)
- Bridge Priority (4 Bits) + Extended System ID (12 bits) + MAC Address (48 Bits)
- Bridge Priority: Kann nur in Schritten von 4096 verändert werden
- Extended System ID: gibt an, zu welchem VLAN der Rahmen gehört
- Somit hat ein Switch so viele Bridge IDs wie VLANs

2.3 Port-Rollen & STP Zustände

- Ermittlung des Root Ports (nächster zur Root Bridge [Link-Kosten])
Kosten manipulieren: *S(config-if)#spanning-tree cost 25*
- Ermittlung des Designated Ports ("nächster" Port zur Root) — Falls "unentschieden":
 - tiefere BID
 - tiefere Port ID
- Beeinflussbar durch höhere Priority:
S(config-if)#spanning-tree port-priority 112
- Zustände: Disabled → Blocking → Listening → Learning → Forwarding
- Nachrichten: Topology Change Notification (TCN), Topology Change Acknowledgement (TCA), Topology Change (TC)
- Topologieänderung: Switch [TCN] → RB — RB [TCA] → Switch — RB [TC] → All Switches
- Zustandsänderung: [TCN] → Blocking → Listening → Learning → Forwarding

3 Point-to-Point Protocol PPP

3.1 Serielle Punkt-zu-Punkt Verbindungen

- Einsatz: Layer-2-Protocol eingesetzt, um zwei Knoten miteinander zu verbinden
- über möglichst alle Medien laufen (Kupferkabel, Glasfaser, etc.)
- Unterstützung verschiedener gleichzeitig laufender Layer-3-Protokolle
- Data Terminating Equipment (DTE): LAN-Abschlussgerät (z.B. Router)
- Data Communication Equipment (DCE): WAN-Abschlussgerät (Telecom → NTU → definiert Takt)
- High Level Data Link Control (HDLC): PPP baut auf HDLC auf
- HDLC: Receive + Send Sequence Number → kann fehlerhafte Rahmen wiederholen lassen
- HDLC-Konfiguration: *R(config-if)#encapsulation hdlc*

3.2 Konzepte & Konfiguration von PPP

3.2.1 PPP enthält 3 Unterprotokolle:

- Rahmenbildung (ähnlich wie HDLC)
- Link Control Protocol (LCP): Hinauffahren, Konfiguration und Test des Links
- Network Control Protocol (NCP): Um versch. Layer-3-Protokolle zu konfigurieren

3.2.2 Aufbau einer PPP Verbindung

1. Phase: Link Establishment, Aushandlung von Optionen
2. Phase (optional): Kontrolle der Linkqualität (Fehlerrate)
3. Phase: Aushandlung der Layer-3-Optionen

3.2.3 PPP Konfiguration

```
R(config-if)#encapsulation ppp
R(config-if)#compress [stack | predictor]
R(config-if)#ppp quality 80
R(config-if)#ppp multilink
```

3.2.4 PPP Authentication

Password Authentication Protocol (PAP) und Challenge Handshake Authentication Protocol (CHAP)
(PAP NICHT mehr sicher, da Passwort Klartext gesendet wird)

```
R1(config)#username R3 password cisco
R(config-if)#ppp authentication chap
```

4 Frame Relay

FR erledigt:

- Rahmenbildung
- Zugang zum Netz
- "Bitticht" Rahmen ans Ziel
- gibt Rahmen in richtiger Reihenfolge ab
- Fehler werden erkannt, aber nicht korrigiert



Abbildung 3.5: Hub and Spoke Topologie (logisch)

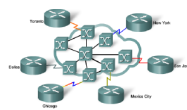


Abbildung 3.6: Hub and Spoke Topologie (physikalisch)

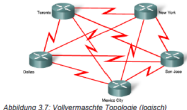


Abbildung 3.7: Vollvermaschte Topologie (logisch)



Abbildung 3.8: Vollvermaschte Topologie (physikalisch)

4.1 Zuordnung FR VC zu IP

Local Management Interface LMI umfasst:

- Virtual Circuit (VC) status message: sollte VC gelöscht werden wird Frame Relay Access Device (FRAD), welches beim Kunden steht, informiert.
- Multicasting: Rahmen wird an eine Gruppe von Zielen gesendet
- Global Addressing: Gibt Data Link Connection Identifiers (DLCIs) [da FR verbindungsorientiert ist], die globale Bedeutung haben
- Simple flow control: Xon/Xoff Mechanismus für Protokoll-Suiten ohne Flusskontrollen

4.2 Konfiguration

Inverses ARP ausschalten:

```
R(config-if)#encapsulation frame-relay ietf
R(config-if)#no frame-relay inverse-arp
```

Statisches Mapping:

```
R(config-if)#frame-relay map ip XXX.XXX.XXX.XXX 102 broadcast
R(config-if)#int s0/0/1.112 point-to-point
R(config-subif)#encapsulation frame-relay ietf
R(config-subif)#ip address [IP] [SUBNET]
R(config-subif)#frame-relay interface-dlci 115
```

5 Access Control Lists ACLs

TBD!!!

6 IP Adressierungsdienste (DHCP, NAT, IPv6)

6.1 DHCP

6.1.1 Einführung

Folgendes wird für Kommunikation benötigt:

- MAC-Adresse (Schicht 2)
- eine Schicht-3-Adresse
- eine Subnetz-Maske
- ein Default Gateway und
- die Adresse eines DNS Servers

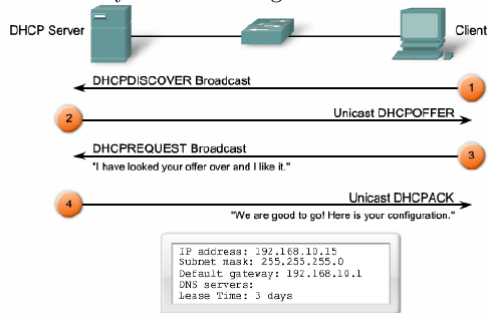
DHCP Server hat Aufgabe, Arbeitsstationen beim Aufstarten mit nötigen Parametern zu versorgen. Er verwaltet die IP-Adressen.

6.1.2 Funktion von DHCP

Drei verschiedene Mechanismen:

- Manuelle Vergabe: Admin vergibt Host gezielt IP-Adresse, DHCP teilt Adresse dem Client mit.
- Automatische Vergabe: DHCP vergibt Host statische IP aus Pool. Es wird keine Lease-Time vereinbart! → permanente Vergabe
- Dynamische Vergabe: DHCP vergibt IPs aus Pool dynamisch. Lease-Time wird von DHCP-Server bestimmt. Wird IP nicht mehr gebraucht, teilt dies der Client am Server mit.

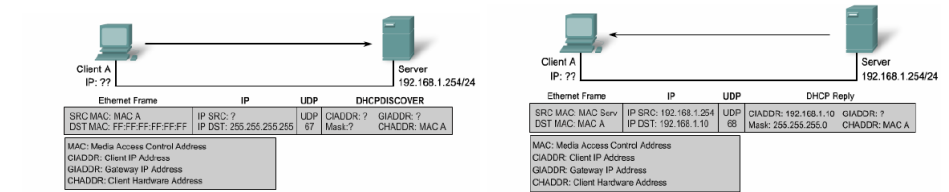
Ablauf der dynamischen Vergabe:



6.1.3 BOOTP & DHCP

BOOTP	DHCP
Static mappings	Dynamic mappings
Permanent assignment	Lease
Only supports four configuration parameters	Supports over 20 configuration parameters

8	16	24	32
OP Code (1)	Hardware type (1)	Hardware address length (1)	Hops (1)
Transaction Identifier			
Seconds – 2 bytes		Flags – 2 bytes	
Client IP Address (CIADDR) – 4 bytes			
Your IP Address (YIADDR) – 4 bytes			
Server IP Address (SIADDR) – 4 bytes			
Gateway IP Address (GIADDR) – 4 bytes			
Client Hardware Address (CHADDR) – 16 bytes			
Server name (SNAME) – 64 bytes			
Filename – 128 bytes			
DHCP Options – variable			



Danach folgt ein DHCP Acknowledge Paket.

6.1.4 Konfiguration DHCP (1./2. = Server; 3. = Client)

- 1.) Reserv. Adressbereich: $R1(config)\#ip\ dhcp\ exclude\text{-}address\ [LOWADDR]\ [HIGHADDR]$
- 2.) Pool: $R1(config)\#ip\ dhcp\ pool\ [POOLNAME]$
 $R1(config)\#network\ [ADDR]\ [MASK]$
 $R1(config)\#default\text{-}router\ [ADDR]$
 $R1(config)\#dns\text{-}server\ [ADDR]$
 $R1(config)\#domain\text{-}name\ [NAME]$
 $R1(config)\#netbios\text{-}name\text{-}server\ [ADDR]$
- 3.) $S1(config\text{-}if)\#ip\ address\ dhcp$

6.1.5 DHCP Relay

Dient dazu Anfragen von zwei Netzen weiterzuleiten (Bsp. 192.168.10.X nach 192.168.11.5).

$R1(config)\#int\ fa0/0 \Rightarrow (192.168.10.X\text{-}Netz)$

$R1(config\text{-}if)\#ip\ helper\text{-}address\ 192.168.11.5$

6.2 NAT (Network Address Translation)

6.2.1 Allgemein

NAT dient dazu private (interne) in öffentliche Adressen zu übersetzen. **Zwecke:** spart IPv4 Adressen; verbirgt interne Adressen vor Kommunikationspartner.

NAT-PAT: NAT \Rightarrow 1:1 Übersetzung von priv. in öff. Adr.; PAT $\Rightarrow n$ priv. in m öff. Adr., wobei $n > m$, wobei meistens $m = 1$

6.2.2 Vor-/Nachteile von NAT

Vorteile:

- Adressierungsschema mit öff. IPv4-Adr. kann beibehalten werden, Adressknappheit entschärft, Firmen mit priv. Adr. in ihren Netzen
 - Wechsel ISP ohne Neunummerierung der Rechner
 - NAT gibt Sicherheit, ersetzt aber nicht Firewall
- Nachteile:
- Performance
 - verstößt gegen grundl. Internet-Prinzip: Veränderung der Adr. \Rightarrow Entschärfung durch stat. Adr.
 - Traceability schwierig
 - Tunneling-Protokolle können beeinträchtigt werden
 - Anwendungen, die TCP-Verbindung von aussen verlangen, funktionieren meist nicht.

6.2.3 Konfiguration von statischem NAT

- 1.) statisches Mapping: $R1(config)\#ip\ nat\ inside\ source\ static\ [LOCAL\ IP]\ [GLOBAL\ IP]$
- 2.) inside/outside IF: $R1(config)\#int\ [fa/s]\ [NUM] \Rightarrow R1(config\text{-}if)\#ip\ nat\ [inside|outside]$

6.3 Konfiguration von dynamischem NAT

- 1.) Pool v. Adr. nach aussen: $R1(config)\#ip\ nat\ pool\ [poolname]\ [START\ IP]\ [END\ IP]\ netmask\ [SUBNET] \Rightarrow$ Bspw: 209.165.200.226 (Start), 209.165.200.24 (End) mit 255.255.255.224

- 2.) Zu übersetzende Adr.: `R1(config)#ip access-list [NUM] [permit|deny] [iprange] [InvertedSubnet] ⇒ Bsp: 1 (NUM), 192.168.0.0 mit 0.0.255.255`
- 3.) Übersetzung: `R1(config)#ip nat inside source list [NUM] pool [POOLNAME]`
- 4.) Auf IFs anwenden: `R1(config)#int [fa|s] [NUM] ⇒ R1(config-if)#ip nat [inside|outside]`

6.3.1 Konfiguration von NAT overload

- a) Nur *eine*, ev. unbekannte, öffentliche Adresse
 - 1.) ACL für internes Netz (siehe Konfiguration dynamisches NAT Punkt 2.))
 - 2.) Definiere PAT: `R1(config)#ip nat inside source list [NUM] int [zB s0/1/0] overload`
 - 3.) IFs konfigurieren (siehe Konfiguration dynamisches NAT Punkt 4.))
- b) Pool von öffentlichen Adressen
 - 1.) ACL und Pool (siehe Konfiguration dynamisches NAT Punkte 2.) und 3.))
 - 2.) Übersetzung: `R1(config)#ip nat inside source list [NUM] pool [POOLNAME] overload`
 - 3.) IFs konfigurieren (siehe Konfiguration dynamisches NAT Punkt 4.))

6.4 IPv6

6.4.1 Allgemein

Verbesserungen gegenüber IPv4

- einfacherer Header
- keine Broadcast Adressen, stattdessen Multicast
- Wegfall der Checksumme im IP Header
- Keine Fragmentierung von IPv6 Paketen, ausser durch den Absender selbst
- IPv6 Header kann durch Optionen erweitert werden
- Autokonfiguration von IPv6 Adressen
- Security (IPsec) ist Teil der Basisspezifikation
- Mobility (Mobile IP) kann besser unterstützt werden.

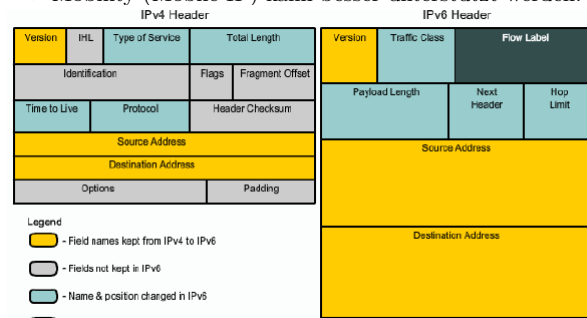


Abbildung 7.19: Gegenüberstellung IPv4- und IPv6-Header

6.4.2 IPv6 Adressierung

- acht Gruppen an je vier Hex-Zahlen, getrennt durch Doppelpunkt getrennt
- führende Nullen können weggelassen werden
- aufeinanderfolgende Nullen können EINMAL durch zwei Doppelpunkte "::" ersetzt werden

Prefixes:

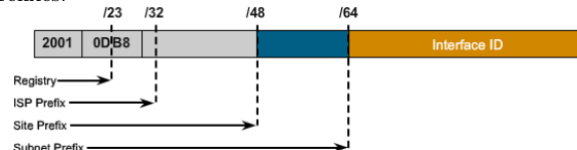


Abbildung 7.21: Form einer globalen IPv6 unicast Adresse

- Private Adressen: FC00:: /7 Unique local addresses
- Link-lokale Adressen: FE80:: /10, gelten nur für einen Link, werden von Router nicht weitergeleitet
- Spezielle Adressen: **Loopback** 0:0:0:0:0:0:1 oder ::1, **Unspecified** 0:0:0:0:0:0:0:0 oder ::

Zuteilung IPv6 Adressen:

- Manuell auf Router IF: `R1(config-if)#ipv6 address [ipv6-addr]/[prefix-length]`
- EUI-64-Zuweisung: `R1(config-if)#ipv6 address [ipv6-addr]/[prefix-length] eui-64`
Nur Prefix wird von Hand konfiguriert. 8-Byte IF-ID wird aus 6-Byte MAC-Adresse generiert

Stateless Autoconfiguration: Einfaches Anschliessen an Netz ("Plug&Play" für Adressvergabe)

DHCPv6: Ergänzend oder konkurrierend zu Stateless Autoconfiguration

6.4.3 IPv6 Übergangsstrategien

- Dual Stack: auf einem Rechner laufen IPv4- und IPv6-Stack parallel
- Manuelle Tunnels: IPv6-Paket wird in IPv4-Paket eingepackt
- 6to4-Tunnel: wie manueller Tunnel, aber automatisch aufgesetzt
- 6rd: Basierend auf 6to4, aber mit IPv6 Adressbereich und IPv4 Tunnelendpunkt
- ISATAP-Tunnel: weiteres automatisches Tunnelverfahren
- Teredo-Tunnel: Automatisches Aufsetzen von End- zu Endgerät (Voraus.: Dual Stack Hosts)
- NAT: Übersetzung von IPv6 zu IPv6 und umgekehrt

6.4.4 Konfiguration von IPv6 Adressen

`R1(config)#ipv6 unicast-routing`

`R1(config)#int [fa|s] [NUM]`

`R1(config-if)#ipv6 address [ipv6Prefix]/[prefixLength] [(eui-64)]`

Achtung: Wenn Router-IF mit EUI-64 konfiguriert, ist es schwierig dieses zu pingen.

Namensauflösung konfigurieren:

- Entweder: `R1(config)#ipv6 host [name] [(port)] [ipv6addr] [(ipv6addr...)]`
- Oder: `R1(config)#ip name-server [addr]`

6.4.5 Konfiguration von RIPng mit IPv6

`R1(config)#ipv6 router rip [NAME]`

`R1(config-if)#ipv6 rip [NAME] enable`

Wobei [NAME] eine ID für den Routing-Prozess ist.

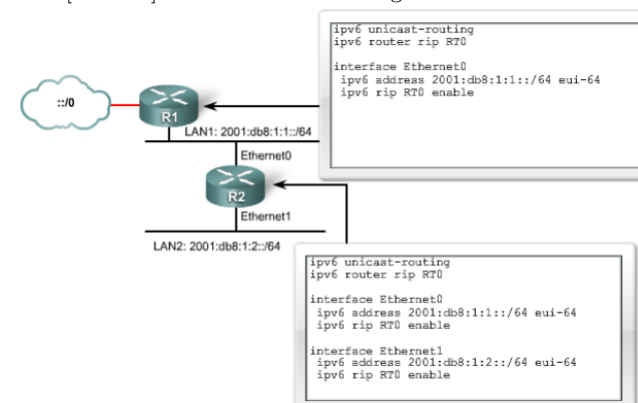


Abbildung 7.27: Konfiguration von RIPng auf einem Router