Administration physikalischer und virtueller Switches

Catalyst 3500 XL & VDE Switch

Serdar Ayalp
Universität Koblenz
serdarayalp@uni-koblenz.de

Suat Algin
Universität Koblenz
salgin@uni-koblenz.de

27. August 2009

Diplomarbeit am Institut für Informatik,
Arbeitsgruppe Steigner,
Universität Koblenz-Landau

Betreuer :

Prof. Dr. Christoph Steigner

Dipl. Inf. Frank Bohdanowicz

Erklärung

Hiermit versichern wir, Suat Algin & Serdar Ayalp, gemäß der Diplomprüfungsordnung Informatik / Computervisualistik der Universität Koblenz-Landau, Campus Koblenz, in der Fassung vom 27.08.2009, dass wir diese Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt haben. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsausschuss vorgelegen.

Koblenz, den 27. August 2009

Inhaltsverzeichnis

1 Einführung

Suat Algin

Ein Switch (engl. Schalter; auch Weiche) ist eine Netzwerk-Komponente zur Verbindung mehrerer Computer bzw. Netz-Segmente in einem lokalen Netzwerk (LAN). Da Switches den Netzwerkverkehr analysieren und logische Entscheidungen treffen, werden sie auch als intelligente Hubs bezeichnet.
Die Funktionsweise eines Switches ist der einer Bridge sehr ähnlich, daher wurde anfangs auch der Begriff Multi-Port-Bridge genutzt [1]. Ziel der Diplomarbeit ist es, eine Dokumentation auf die Beine zu stellen, der interessierten Studenten der Informationstechnologie die Möglichkeit bietet, einerseits anhand von physikalischen Switches Administrationsaufgaben nachzuempfinden und andererseits anhand von virtuellen Switches größere Netzwerktopologien aufzubauen. Diese Virtualisierung wird durch das von Virtual Square entwickelte Tool VDE erreicht. Die physikalischen Switches bestehen aus vier Catalyst 3500 XL. Im Laufe dieser Arbeit wird sowohl die Bedienung der einzelnen Systeme untereinander, wie auch die Vernetzung der virtuellen Switches mit den physikalischen Switches erläutert. In diesem Zusammenhang wird auch auf Protokolle und Methoden wie das Spanning Tree Protokoll oder die Virtualisierung eines Netzes durch VLANs eingegangen.
Zum Schluss kann der Leser das gelernte in einigen praktischen Aufgaben anwenden.

[1] http://de.wikipedia.org/wiki/Switch_(Computertechnik)

2 Speicherarten eines Switches

Serdar Ayalp

Der Catalyst 3500 XL Switch besitzt verschiedene Speicherarten für unterschiedliche Aufgaben.[6]

- **EEPROM (Electric Erasable Programmable Read Only Memory) :** Bekannt auch als *Flash-Memory*. Das Betriebsystem eines Switches(IOS)[2] wird im Flash-Memory gespeichert und ggf. durch eine neue IOS-Version, welche meistens durch einen TFTP-Server[3] zur Verfügung gestellt und hochgeladen wird, ersetzt.
- **DRAM (Dynamic Random Access Memory) :** In diesem Speicher wird die aktuelle Konfiguration gespeichert. Der Speicherinhalt ist flüchtig (engl. *volatile*). Dies bedeutet, dass die gespeicherten Informationen bzw. Konfigurationen bei fehlender Betriebsspannung verloren gehen.
- **NVRAM (*Non-Volatile* (Nichtflüchtiges) RAM) :** In diesem Speicher wird die Start-Konfiguration gespeichert. Nach einem Stromausfall wird diese vom IOS-Betriebsystem ausgelesen und der Switch wird mit dieser Konfiguration gestartet.
- **Boot ROM (Read Only Memory) :** In diesem wird der sogenannte *Boot-Loader*[4] gespeichert.

Nach dem Einschalten des Switches sucht der Boot-Loader im Flash-Memory nach dem IOS-Betriebsystem und lädt es in den Arbeitsspeicher des Switches. Die aktuelle Start-Konfiguration wird auch vom NVRAM ausgelesen und in den Arbeitsspeicher geladen.

[2] **IOS** : *Internetwork Operating System*, das Betriebssystem von Cisco-Routern und Switches[11]
[3] **TFTP** :Trivial File Transfer Protocol [14]
[4] **Zitat** :Ein *Boot-Loader* ist eine spezielle Software, die gewöhnlich durch die Firmware (z.B. BIOS) eines Rechners von einem bootfähigen Medium geladen und anschließend ausgeführt wird. Der Boot-Loader lädt dann weitere Teile des Betriebssystems. [9]

3 Verbindungsaufbau per CLI

Serdar Ayalp

Jeder Cisco Catalyst Switch enthält ein sogenanntes *Command-Line-Interface*[5], mit dessen Hilfe der Switch konfiguriert werden kann. Man kann das CLI entweder durch den Konsolen-Port (engl. *console port*) oder per Telnet über einen Ethernet- bzw. Fast-Ethernet-Port erreichen.[7]
Für eine erstmalige Konfiguration ist eine Verbindung durch den Konsolen-Port notwendig. Erst nach der Zuordnung einer IP-Adresse ist der Switch per Telnet über seine Ports erreichbar. Man verbindet den Konsolen-Port am Gerät mit der COM-Schnittstelle des Rechners. Für die Kommunikation ist auch ein Terminal-Programm wie z.B. Hyperterminal unter XP (siehe Abb. 1) oder TuTTY unter Vista (siehe Abb. 2) notwendig.

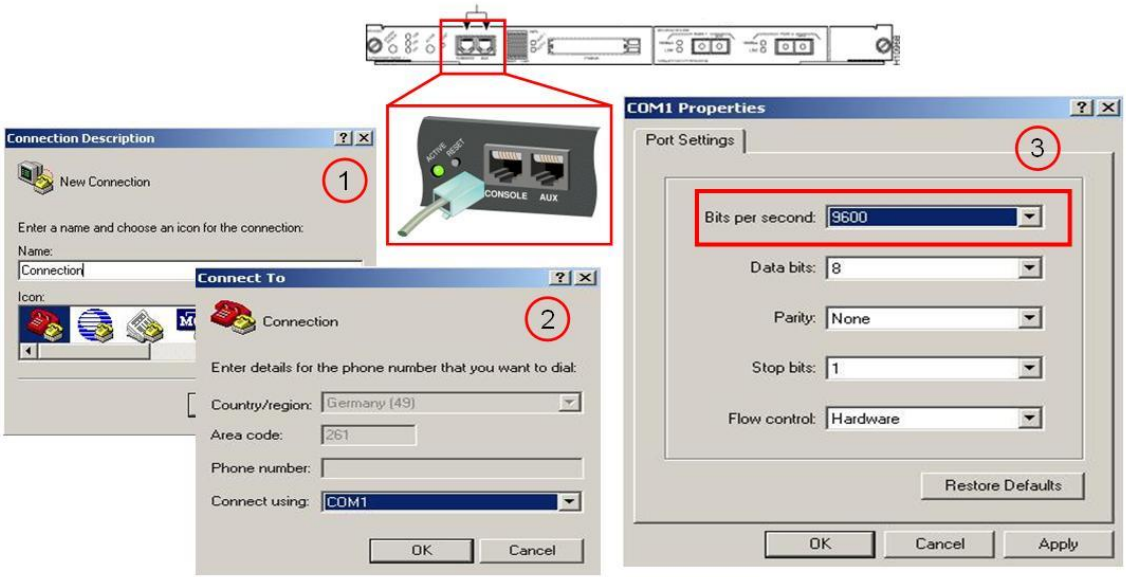

Abbildung 1: Verbindungsaufbau unter XP per Hyperterminal über den Konsolen-Port

[5]CLI, siehe Kapitel 4.1

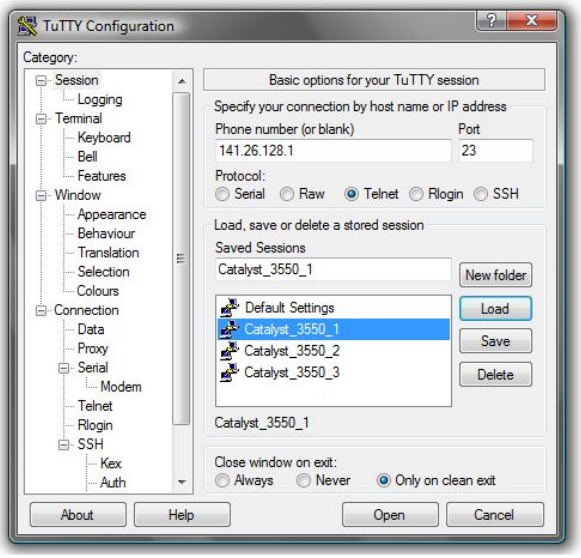

Abbildung 2: Verbindungsaufbau unter Vista per TuTTY über Telnet

Nach einem erstmaligen Start befindet sich der Switch in einem interaktiven Modus, in dem Angaben wie Name des Switches, Passwörter etc. vom Benutzer bezüglich der Konfiguration erwartet werden. Besonders wichtig sind die Passwörter, weil damit verhindert wird, dass jeder auf den Switch zugreift. Verbindungen über den Konsolen-Port oder über Telnet müssen also durch Passwörter geschützt werden.
Ein Zugriff über den Konsolen-Port wird mit einem sogenannten ***enable secret***-Passwort geschützt. Einem Switch wird ein *enable secret*-Passwort im sogenannten *Global Configuration*-Mode (siehe Kapitel 4.1) folgendermaßen zugeordnet :

```
Switch_Name(config)#enable secret xxx
Switch_Name(config)#
```

Abgesehen davon existiert noch ein Befehl, mit dem man das Zugriffspasswort bestimmen bzw. ändern kann- das sogenannte ***enable***-Passwort. Der einzige Unterschied zwischen beiden ist die Speicherung des Passworts in der Konfigurationsdatei *config.text*[6], die im Flash-Speicher des Switches gehalten wird. Wenn man ein *enable secret*-Passwort verwendet, wird es verschlüsselt bevor es in die *config.text*-Datei geschrieben wird. Im Gegensatz dazu wird ein ***enable***-Passwort einfach in die Konfigurationsdatei geschrieben, wo es von jedem gelesen werden kann.

```
Switch_Name#dir
Directory of flash:/

  2  -rwx     1811552   Mar 01 1993 c3500xl-c3h2s-mz.120-5.WC17.bin
```

[6]siehe Unten die Ausgabe des Befehls *dir*.

```
  3  -rwx        94375   Apr 30 2001 c3500XL-diag-mz-120-5.3.WC.1
  4  drwx          768   Mar 01 1993 html
 16  -rwx          109   Mar 01 1993 info
165  -rwx          720   Mar 01 1993 vlan.dat
166  -rwx           74   Mar 01 1993 placement.txt
 19  -rwx         2053   Mar 01 1993 config.text
 17  -rwx          109   Mar 01 1993 info.ver
 18  -rwx          315   Mar 01 1993 env_vars

3612672 bytes total (581120 bytes free)
Switch_Name#
```

Natürlich ist ein zugeordnetes *enable secret*-Passwort durch einen sogenannten Negations-Befehl (engl. ***no***-Command) (siehe Kapitel 4.1) zu entfernen.

```
Switch_Name(config)#no enable secret
Switch_Name(config)#
```

Auch die Zugriffe über Telnet lassen sich kontrollieren (siehe Kapitel 4.1). Ein Cisco-Catalyst-Switch erlaubt 16 gleichzeitige Telnet-Sitzungen (engl. *Sessions*). Um alle Telnet-Verbindungen durch ein Passwort zu schützen, sind folgende Zeilen im *Global Configuration*-Modus einzugeben.

```
Switch_Name(config)#line vty 0 15
Switch_Name(config-line)#password koblenz
Switch_Name(config-line)#end
```

Die Zahlen 0 und 15 bedeuten, dass man alle 16 Sitzungen konfigurieren will[7]. Um das Ergebnis zu testen, geht man wieder auf den *privileged*-Mode (siehe Kapitel 4.1) zurück und lässt die aktuell laufende Konfigurationsdatei durch den Befehl *show running-config* ausgeben. Ungefähr am Ende der Ausgabe sollten folgende Zeilen stehen.

```
Switch_Name#show running-config
.....................
line vty 0 4
 password koblenz
 login
line vty 5 15
 password koblenz
 login
```

Mit folgendem Befehl wird der Switch jedes Mal mit der momentan laufenden Konfiguration gestartet.

```
Switch_Name#copy running-config startup-config
```

[7] **VTY** : Terminal-Leitungen, auch *Virtual Terminal Lines* oder VTYs genannt.

4 Managementoptionen eines Switches

Serdar Ayalp

Ein Cisco-Switch der Familie 3500 XL mit dem Betriebsystem *Cisco IOS Release 12.0(5)* stellt folgende Managementoptionen für die Konfiguration zur Verfügung.

- **Command-Line Interface (CLI, Befehlszeile) :** Ermöglicht die Eingabe umfangreicher Cisco-Befehle zur Konfiguration eines Switches.
- **Cluster Management Suite (CMS) :** Eine HTML-basierte grafische Benutzeroberfläche, die mit irgendeinem Web-Browser, der JavaScript und Java unterstützt, genutzt werden kann. Durch die Visualisierung einzelner Schritte werden Operationen mit Switches(z.B. Cluster-Management)[8]enorm erleichtert.

Alle CMS-Funktionen werden durch einen internen, im Flash-Speicher gespeicherten, Web-Server zur Verfügung gestellt. Das Web-basierte Management nutzt das HTTP-Protokoll zur Konfiguration eines Switches, mit dem eine Verbindung über einen seiner Ethernet-Ports eingerichtet wurde. Daher wird es nicht empfohlen, diesen Port direkt oder durch eine falsche Konfiguration zu deaktivieren, da man über diesen mit dem Switch kommuniziert.

4.1 Cisco IOS Command-Line Interface (CLI, Befehlszeile)

Serdar Ayalp

Durch die unterschiedlichen zur Verfügung gestellten Konfigurationsmodi lässt sich auf unterschiedlichen Ebenen mit einem Switch arbeiten. Jeder Modus verfügt über eine Reihe von spezifischen Befehlen. Beispielsweise lässt sich der Befehl *interface* nur im sogenannten *global configuration mode* anwenden. Die von einem Cisco 3500 XL Switch unterstützten Konfigurationsmodi sind:

- **User EXEC:** Der Switch befindet sich immer in diesem Modus, wenn eine neue Sitzung (engl. *Session*) eröffnet wird.

  ```
  Switch_Name>
  ```

  In diesem Modus lassen sich System-Informationen anzeigen. Man darf aber keine Konfigurationsänderung vornehmen. Um diesen Modus zu verlassen, sind die Befehle ***logout*** oder ***quit*** einzugeben.

- **Privileged EXEC:** Diesen Modus erreicht man, wenn man im *User EXEC*-Modus den Befehl ***enable*** eingibt.

[8]Für weitere Informationen siehe Kapitel 4.2

```
Switch_Name>enable
Switch_Name#
```

In diesem Modus lassen sich Konfigurationsänderungen vornehmen. Deshalb ist die Nutzung mit einem Passwort einzuschränken. Diesen Modus verlässt man mit einem ***disable***-Befehl.

- **VLAN Database:** Diesen Modus erreicht man, wenn man sich im *Privileged EXEC*-Modus befindet und den Befehl ***vlan database*** eingibt.

```
Switch_Name#vlan database
Switch_Name(vlan)#
```

VLAN-spezifische Befehle können in diesem Modus verwendet werden, z.B. VLANs erzeugen, löschen etc.. Mit einem ***exit***-Befehl kehrt man wieder in den *Privileged EXEC*-Modus zurück.

- **Global Configuration :** Mit einem ***configure*** *[terminal]*-Befehl im *privileged EXEC*-Modus tritt man in diesen Modus ein. Man wechselt in diesen Modus, wenn man Konfigurationsänderungen für den ganzen Switch vornehmen möchte. Um diesen Modus zu verlassen, sind die Befehle ***exit*** oder ***end*** bzw. die Tasten-Kombination ***Ctrl-Z*** einzugeben.

```
Switch_Name#configure terminal
Switch_Name(config)#
```

- **Interface Configuration :** Wenn man spezifische Einstellungen für Switch Interfaces bzw. Ports vornehmen möchte, gibt man im *Global Configuration*-Modus den Befehl ***interface*** mit einem speziellen Port- bzw. Interface-Bezeichner ein. z.B. Konfiguration des ersten Ports des Switches.

```
Switch_Name(config)#
Switch_Name(config)#interface fa0/1
Switch_Name(config-if)#
```

Mit dem ***exit***-Befehl kehrt man wieder in den *Global Configuration*-Modus zurück.

- **Line Configuration :** *Interface Configuration*-Modus beinhaltet weder die Konfiguration des Konsolen-Ports noch die der sogenannten *Virtual Terminals* (Telnet-Ports). Um diese zu konfigurieren, ist der Befehl ***line*** zu benutzen. Zugriff auf die *Line Configuration* für den Konsolen-Port gibt uns die Möglichkeit, Einstellungen wie z.B. Passwort-Zuweisungen für den Konsolen-Port vorzunehmen. Diesen Modus erreicht man, wenn man im *Global Configuration*-Modus ist und den Befehl ***line*** eingibt. Je nach Bedarf ist als Parameter entweder *console* oder *vty*(für Telnet-Verbindungen) einzugeben [3].

```
141.26.125.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.125.1(config)#line console 0
141.26.125.1(config-line)#
```

Telnet-Ports sind als *Virtual Terminals* bekannt. Verschiedene Switches ermöglichen je nach IOS-Version eine unterschiedliche Anzahl gleichzeitiger Telnet-Sitzungen (engl. *Sessions*). In unserem Fall, ermöglicht das *Cisco IOS Release 12.0(5)* 16 simultane Telnet-Sitzungen[9]. Um diesen Modus zu verlassen, ist ein ***exit*** einzugeben. Hiermit kommt man wieder auf den Modus *Global Configuration* zurück.

In Abbildung 3 werden alle Übergänge zwischen den Konfigurationsmodi noch einmal ausführlich dargestellt.

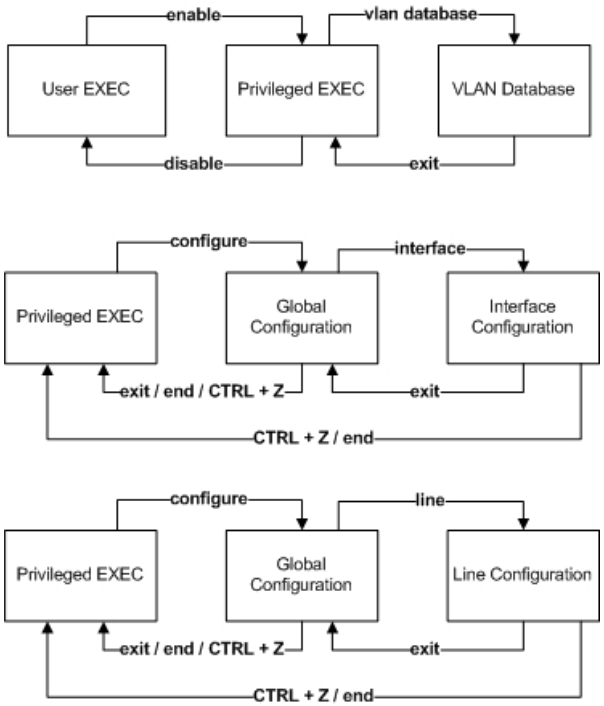

Abbildung 3: Übergänge zwischen den Konfigurationsmodi

In jedem IOS existieren einige Zeichen, die bei der Befehlseingabe sehr hilfreich sind. Zum Beispiel lässt sich mit einem Fragezeichen (**?**) (ohne *Return* zu drücken) alle verfügbaren Befehle des jeweiligen Modus anzeigen.

[9]Für ein Beispiel, siehe Kapitel 3

```
Switch_Name> ?
Exec commands:
  <1-99>            Session number to resume
  access-enable     Create a temporary Access-List entry
  clear             Reset functions
  connect           Open a terminal connection
  disable           Turn off privileged commands
  disconnect        Disconnect an existing network connection
  enable            Turn on privileged commands
  exit              Exit from the EXEC
  help              Description of the interactive help system
  lock              Lock the terminal
  login             Log in as a particular user
  logout            Exit from the EXEC
  name-connection   Name an existing network connection
  ping              Send echo messages
  rcommand          Run command on remote switch
  resume            Resume an active network connection
  set               Set system parameter (not config)
  show              Show running system information
  systat            Display information about terminal lines
  telnet            Open a telnet connection
  terminal          Set terminal line parameters
  traceroute        Trace route to destination
  tunnel            Open a tunnel connection
  where             List active connections
```

Bzw. mit einem Fragezeichen nach einem Befehl sind alle verfügbaren Variablen bzw. Parameter im jeweiligen Modus zu sehen.

```
Switch_Name> show ?
  cgmp                Display CGMP information
  class-map           Show QoS Class Map
  clock               Display the system clock
  diags               Show runtime diagnostic info
  errdisable          Error disable
  etherchannel        EtherChannel information
  exception           exception informations
  flash:              display information about flash: file system
  history             Display the session command history
  hosts               IP domain-name, lookup style, nameservers,
                      and host table
  html                HTML helper commands
  location            Display the system location
  mac-address-table   MAC forwarding table
  policy-map          Show QoS Policy Map
  port                Show switch port configuration
  queue               Show queue contents
  queueing            Show queueing configuration
  rmon                rmon statistics
```

```
  rps                 Show the Redundant Power System (RPS) status
  sessions            Information about Telnet connections
  snmp                snmp statistics
  spanning-tree       Spanning tree topology
  tacacs              Shows tacacs+ server statistics
  terminal            Display terminal configuration parameters
  udld                UDLD information
  users               Display information about terminal lines
  version             System hardware and software status
  vlan                VTP VLAN status
  vmps                VMPS version information
  vtp                 VTP information
```

Mit der *Tabulator*-Taste ist es auch möglich, einen Befehl zu vervollständigen.

```
Switch_Name> configure ter <tab>
Switch_Name> configure terminal
```

Eine erstellte Konfiguration kann durch einen *Negations*-Befehl (engl. *no*-command) zurückgesetzt werden. Im folgenden Beispiel wird erst ein VLAN erzeugt und danach mit einem *Negations*-Befehl wieder gelöscht.

```
Switch_Name(vlan)#vlan 234
VLAN 234 added:
    Name: VLAN0234
Switch_Name(vlan)#no vlan 234
Deleting VLAN 234...
Switch_Name(vlan)#
```

4.2 Konfiguration durch die Cluster Management Suite (CMS)

Serdar Ayalp

Die *Cluster Management Suite* besteht aus vier zusammenhängenden Applikationen (*Cluster Builder, Cluster View, Cluster Manager, Visual Switch Manager(VSM)*), die für die folgenden Aufgaben zuständig sind:

- Erzeugung eines Clusters zur zentralen Konfiguration aller Switches über einen sogenannten ***Command***-Switch.
- Überwachen der Switches und ihrer Ports
- Visualisierung von Links- und Leistungsinformationen.

Die CMS-Applikationen unterstützen mit einer graphischen Benutzungsoberfläche die Überwachung und Konfiguration aller Cluster eines Switches.

4.2.1 Zugriff auf die CMS

Serdar Ayalp

Man kann auf ein erzeugtes Cluster (*Erzeugen und Verwalten der Cluster*, siehe Kapitel 5.1) durch die IP-Adresse des ***Command-Switches*** oder eines normalen *Standalone-Switches* zugreifen. Durch die Eingabe der IP-Adresse landet man auf einer web-basierten Oberfläche (siehe Abb. 4), wo man viele Aktionen ausführen kann.

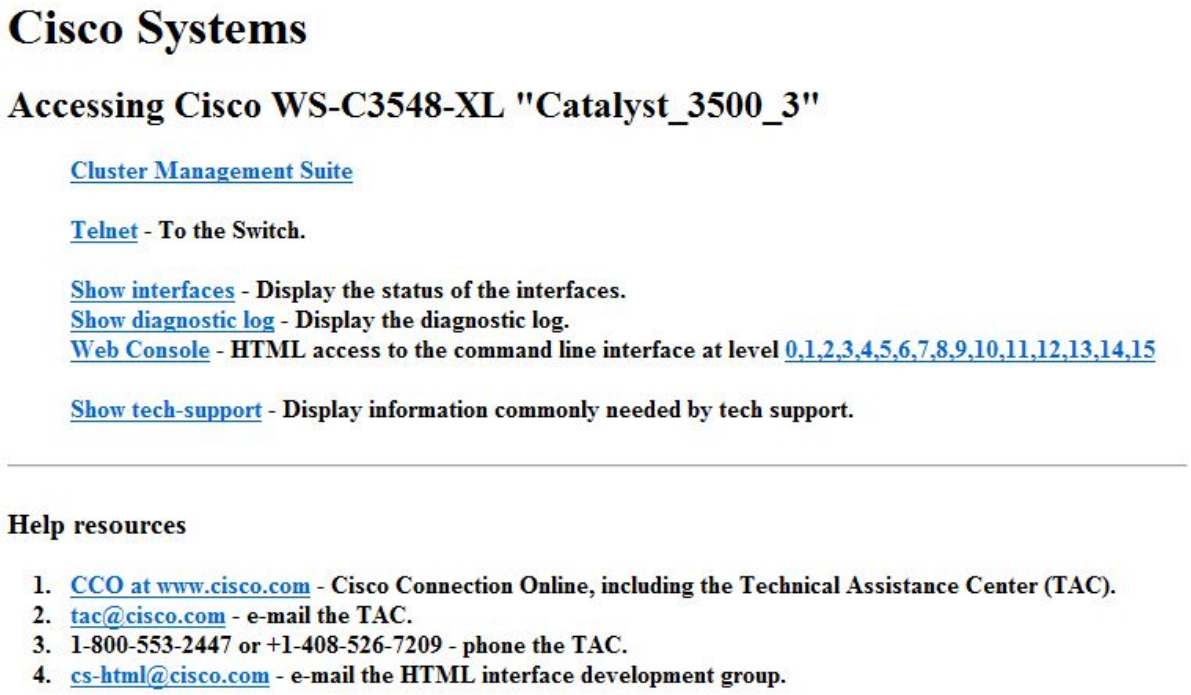

Abbildung 4: Mögliche Aktionen auf der web-basierten Oberfläche

Wie oben in der Abbildung zu sehen ist, werden viele nützliche Dienste, die man mit einem einfachen Browser vornehmen kann, zur Verfügung gestellt. Diese sind :

- **Cluster Management Suite** : Graphische Oberfläche, auf der man sämtliche Einstellungen über einen Switch ändern kann.
- **Telnet** : Telnet-Verbindung zum Switch herstellen.

```
Direkte URL-Eingabe :
telnet://141.26.128.3
```

- **Show Interfaces** : Alle Ports werden mit sämtlichen Eigenschaften aufgelistet. Zum Beispiel:
  - Port *down* oder *up*, (z.B. *FastEthernet0/1 is down*)
  - Typ des Ports, (z.B. *Hardware is Fast Ethernet*)
  - Interne Hardware-Adresse des Ports, (z.B. *0005.dd3e.85c1*)
  - Maximum Transmission Unit (MTU)[10], (z.B. *MTU 1500 bytes*)

  ```
  Direkte URL-Eingabe :
  http://141.26.128.3/exec/show/interfaces/CR
  ```

- **Show diagnostic log** : Die Log-Datei des Switches wird ausgegeben.

- **Web Console** : Mit diesem Eintrag ist es möglich, das *Command-Line-Interface (CLI)* per HTTP zu erreichen. Wegen der Geschwindigkeit ist es aber nicht zu empfehlen.

  ```
  Direkte URL-Eingabe :
  http://141.26.128.3/level/15/exec/-
  ```

- **Show tech-support** : Auflistung der Konfigurationsdatei und der technischen Informationen des Switches.

  ```
  Direkte URL-Eingabe :
  http://141.26.128.3/exec/show/tech-support/cr
  ```

Mit direkten URL-Eingaben im Browser lässt sich auch eine direkte Verbindung zum zuständigen Interface herstellen.

[10] ***Zitat*** : Die Maximum Transmission Unit (MTU) beschreibt die maximale Größe einer Nutzlast in Bytes, welche auf der Sicherungsschicht (Schicht 2) verwendet und dabei als ganzes Stück auf einmal übertragen werden kann. [12]

4.2.2 Cluster Management Suite

Serdar Ayalp

Die Cluster Management Suite (CMS) von Cisco erlaubt einem Benutzer mit einem Standard-Webbrowser an mehreren Catalyst-Switches gleichzeitig Konfigurationen durchzuführen.

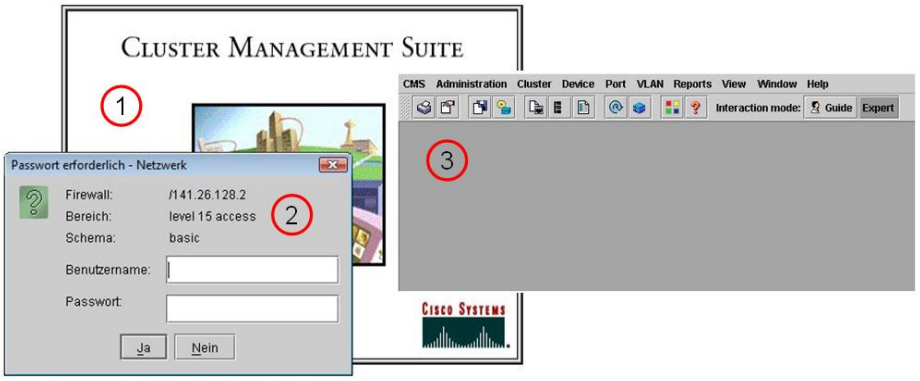

Abbildung 5: Cluster Management Suite

Für die Visualisierung ist unbedingt ein Browser notwendig, der Java unterstützt. Es ist aber leider nicht möglich, die CMS mit den neuen Java-Versionen zu betreiben. Daher haben wir uns in unserer Arbeit für Java RE 1.4.2 Update 18 entschieden [11]. Je nach Switch-Model und Java-Version kann die CMS ein unterschiedliches Aussehen haben[12]. Mit den Menü-Einträgen oben werden viele Aktionen bereitgestellt. Die wichtigsten sind unter anderem **Cluster**, **VLAN** und **Port**. Beispielweise hat man unter dem Eintrag *Port* die Möglichkeit, auf einzelne Porteinstellungen zuzugreifen. Diese sind z.B. Portgeschwindigkeit, Portstatus, Duplex-Mode etc.

[11] Besuche die Sun's Webseite, um Java herunterzuladen, http://www.sun.com
[12] siehe Abb. 5 für Java RE 1.4.2 Update 18 und Switch-Model Cisco Catalyst 3548-XL

5 Cisco Catalyst 3500 Management

Suat Algin

Dieser Abschnitt handelt von den zwei verschiedenen Konfigurationsmöglichkeiten der Catalyst 3500 XL Switch-Familie. Zum Einen besteht die Möglichkeit die Switches, anhand der Cluster Management Suite(CMS) über ein Webinterface zu konfigurieren. Zum Anderen existiert natürlich auch die herkömmliche, unkomfortable Methode die Switches über das Command Line Interface(CLI) zu administrieren. Wie man in den nächsten Kapiteln sehen kann, lassen sich einige Konfigurierungen nur über dieses CLI realisieren. Die Konfiguration anhand der CMS kann sowohl bei Standalone Switches, wie auch bei einer Gruppierung von vielen Switches zu einem Cluster erfolgen. Bei einem Cluster kann die gesamte Konfigurierung über den sogennanten Command-Switch erfolgen.

5.1 Clusterisierung

Serdar Ayalp

Zur Administration eines Netzwerks mit mehr als einem Switch über eine einzige Schnittstelle benötigt man ein Cluster. Ein Cluster ist eine zu einer Gruppe zusammengefasste Menge von Switches mit einem „Command Switch“. Der „Command-Switch“ wird zur zentralen Konfigurierung genutzt. Also werden mehrere physikalische Switches zu einer logischen Gruppe zusammengefasst. Diese Gruppe von Switches präsentiert sich im Netzwerk nun als ein logischer Switch. Voraussetzung dafür ist der Einsatz der Cluster Software auf jedem einzelnen Switch und die softwareseitige Erstellung eines Clusters. Dabei können sich die Switches an der gleichen oder an unterschiedlichen Stellen im Netzwerk befinden. Der größte Vorteil eines Clusters ist, dass man alle Switches durch nur eine einzige IP-Adresse ansprechen kann. Es lassen sich bis zu 16 Switches einem Cluster, mit folgenden Einschränkungen, hinzufügen:

- 1 sogenannter ***Command-Switch***
- 15 ***Member-Switches***

Der Command-Switch ist der einzige *Access-Point* zum Cluster und wird für das Verwalten, die Konfiguration und das Monitoring anderer Member-Switches verwendet.

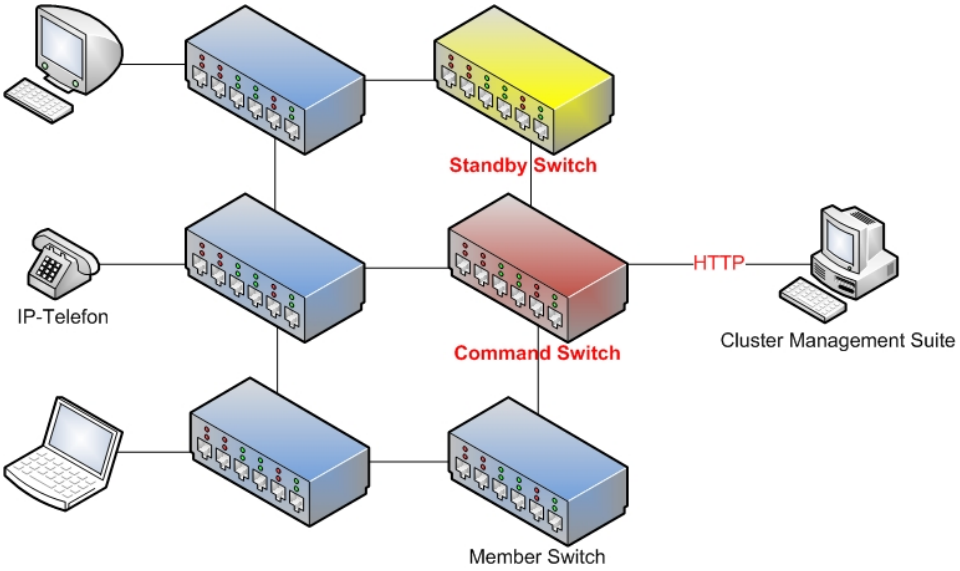

Abbildung 6: Ein Cluster mit allen möglichen Komponenten

Bei der Erzeugung eines Clusters wird der erzeugende Switch zum Command-Switch. Dieser muss aber folgende Kriterien erfüllen, um als „Command-Switch“ agieren zu können:

- Minimum IOS Release 12.0(5)XU [13]
- Zuweisung einer IP-Adresse
- Aktivierung der Cisco Discovery Protocol Version 2
- Der Switch gehört zu keinem anderen Cluster
- Die VLAN-Zugehörigkeit stimmt mit allen anderen Switches überein
- Es dürfen keine Access-Lists konfiguriert sein [14]

Sind diese Kriterien erfüllt, kann man über den Menüpunkt „Cluster“ / “Create Cluster“, dass Cluster erzeugen und den Switch damit zum „Command-Switch“ machen.

[13] Weitere Informationen zur Kompatibilität ihres IOS unter: http://www.cisco.com [2]
[14] Ausgenommen der Access Class 199, welche durch die Konfigurierung des Command-Switch erzeugt wird

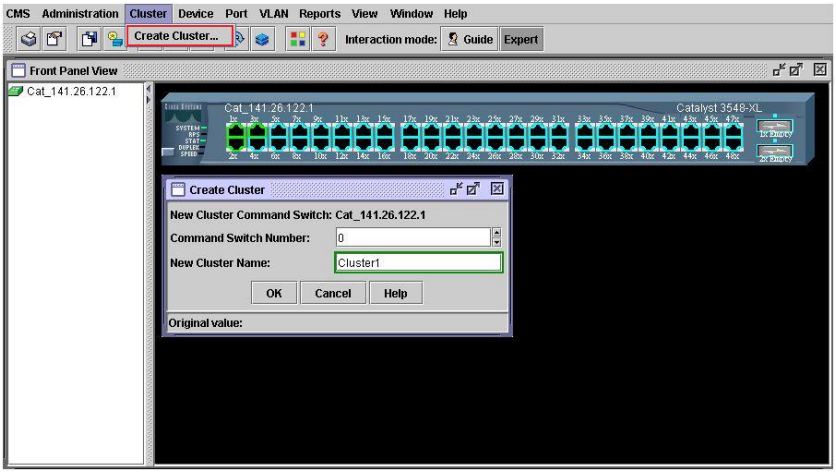

Abbildung 7: Cluster Erzeugung(CMS)

Für die Konfiguration und das Management in einem Cluster wird das ***Hot Standby Router Protocol (HSRP)*** verwendet, welches eigentlich für Cisco-Router entwickelt wurde. Genauso wie bei den Switches werden physikalische Router zur einer logischen Gruppe zusammengefasst und diese Gruppe von Routern wird im Netzwerk als ein logischer Router präsentiert.

> Hierzu wird dem logischen Router eine virtuelle IP-Adresse und eine virtuelle MAC-Adresse zugeordnet. Einer der Router innerhalb der Gruppe wird als der ***Primary-Router*** definiert, dieser bindet darauf die virtuelle MAC und die virtuelle IP an seine Netzwerkschnittstelle und informiert in regelmäßigen Abständen (*Default*: 3 Sekunden) die anderen Router der Gruppe, die als Secondary-Router agieren, mittels einer Multicast-Nachricht an die Zieladresse 224.0.0.2, Port UDP 1985. Fällt der Primary-Router aus und bleiben die Multicast-Pakete für länger als die Zeit des „**Holddown timers**“ (*Default*: 10 Sekunden) aus, so werden die virtuelle IP-Adresse und die virtuelle MAC-Adresse sofort auf einen der ***Secondary-Router*** übertragen, der damit zum neuen Primary-Router wird. Die MAC- und die IP-Adresse werden übertragen, damit die betroffenen Hosts nicht ihren ARP-Cache aktualisieren müssen.(**Zitat :**[10])

Einige IOS-Versionen unterstützen nicht die Erzeugung von Clustern (engl. *Clustering*) oder bieten nicht sämtliche Funktionalitäten an. In diesem Fall ist ein IOS-Upgrade (empfohlen ***IOS 12.0(5)XU***) notwendig, damit alle Switches miteinander auf dem selben Niveau kommunizieren können. Genauso wie bei den Routern ist es auch in einem Cluster von Switches möglich, einen sogenannten *Secondary-Switch* bzw. ***Standby-Switch*** zu definieren, der aktiviert wird, wenn der Command-Switch ausfällt.

Nach der Erzeugung eines Clusters können alle Switches (*Kandidaten-Switches*), die folgende Bedingungen erfüllen, zum Cluster hinzugefügt werden.

- Auf dem Switch läuft ein Cluster-fähiges IOS-Betriebsystem.
- Der Switch muss dem Command-Switch mit einem Port angebunden sein, das dem gleichen Management-VLAN gehört wie der Command-Switch.
- Der Switch ist kein Command- bzw. Member-Switch in einem anderen Cluster im Netz.

Ein Kandidaten-Switch kann eine IP-Adresse haben, es ist aber nicht unbedingt notwendig.

5.1.1 Erzeugen der Cluster per CMS

Serdar Ayalp

Folgende Schritte sind zu befolgen, um ein Cluster per CMS zu erzeugen :

- Verkabelung zwischen den Switches, die das Clustering unterstützen.
- Zuweisung einer IP-Adresse und die Aktivierung als Command-Switch.
- Starten des *Cluster Builder* unter der CMS. Hinzufügen der Kandidaten-Switches zum Cluster.

Nach der Gestaltung des Clusters kann man nun durch die Eingabe der IP-Adresse im Browser auf alle Switches zugreifen [15].

5.1.2 Erzeugen der Cluster per CLI

Serdar Ayalp

Beginnend im *Privileged-EXEC*-Modus auf dem Command-Switch sind folgende Schritte zu befolgen, um den Command-Switch zu aktivieren und die Kandidaten-Switches zum Cluster hinzuzufügen.

```
Cat_141.26.122.1#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cat_141.26.122.1(config)#Cluster enable Cluster1 0
Cat_141.26.122.1(config)#exit
Cat_141.26.122.1#show cluster
Command switch for cluster "Cluster1"
```

[15] Aussehen eines Clusters(Abbildung 9)

```
        Total number of members:        3
        Status:                         0 members are unreachable
        Time since last status change:  0 days, 0 hours, 6 minutes
        Redundancy:                     Disabled
        Heartbeat interval:             8
        Heartbeat hold-time:            80
        Extended discovery hop count:   3
Cat_141.26.122.1#
```

Die Vorgehensweise lässt sich mit dem folgenden Beispiel vertiefen.

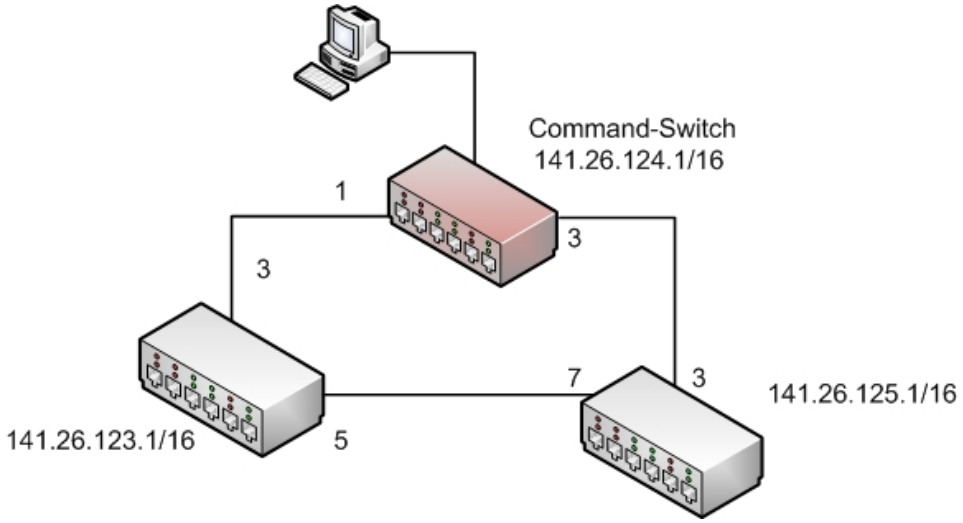

Abbildung 8: Beispiel zur Cluster-Erzeugung(Zahlen = Ports)

```
141.26.124.1#configure terminal
141.26.124.1(config)#cluster enable Cluster01 0
141.26.124.1(config)#end
141.26.124.1#
```

Mit diesen Zeilen wird zuerst in den Konfigurations-Modus gewechselt und anschließend der aktuelle Switch als Command-Switch festgelegt (mit dem Parameter 0). Hierbei zeigt *Cluster01* in der Zeile den Namen des Clusters an. Mit einem *end* kehrt man wieder in den Privileged-EXEC-Modus zurück.

```
141.26.124.1#show cluster candidates

                                                  |--Upstream--|
MAC Address      Name           PortIf     Hops   SN PortIf
0005.dd3e.85c0   141.26.123.1   Fa0/3      1      0  Fa0/1
0005.dd3e.ce80   141.26.125.1   Fa0/3      1      0  Fa0/3
```

Anschließend wird geprüft, welche Kandidaten-Switches zur Verfügung stehen, die zum Cluster hinzuzufügen sind. Hierbei werden sie mit ihren MAC-Adressen, Namen und Port-Nummern aufgelistet. Unter *Upstream* sind die gegenseitigen Port-Nummern (auf dem Command-Switch) gelistet.

```
141.26.124.1#show cluster members
                                              |--Upstream--|
SN MAC Address    Name          PortIf Hops   SN PortIf     State
0  0006.5377.0f40 141.26.124.1         0                   Up (Cmdr)
```

Wie man sieht, beinhaltet das Cluster aktuell nur einen Switch, also den Command-Switch. Hierbei ist zu beachten, dass der Command-Switch mit der *Switch-Nummer (SN)* 0 ausgestattet ist. Alle weiteren Kandidaten-Switches bekommen eine Nummer zwischen 1 und 15. Mit den folgenden Befehlen werden die Kandidaten-Switches zum Cluster hinzugefügt. Die Zahlen nach *member* sind die zugeordneten Switch-Nummern. Hinzu kommen die mit *show cluster candidates* herausgefundenen MAC-Adressen, welche benutzt werden, um den jeweiligen Switch anzusprechen. Wenn ein Kandidaten-Switch ein *enable-Passwort* haben sollte, dann muss es auch für die Autorisierung am Ende der Zeile angegeben werden[16].

```
.124.1#configure terminal
.124.1(config)#cluster member 1 mac-address 0005.dd3e.85c0 password xxx
.124.1(config)#cluster member 2 mac-address 0005.dd3e.ce80 password xxx
.124.1(config)#end
.124.1#
```

Es kann noch einmal mit *show cluster members* verifiziert werden, ob die Kandidaten-Switches jetzt im Cluster sind.

```
141.26.124.1#show cluster members
                                              |--Upstream--|
SN MAC Address    Name          PortIf Hops   SN PortIf    State
0  0006.5377.0f40 141.26.124.1         0                   Up (Cmdr)
1  0005.dd3e.85c0 141.26.123.1 Fa0/3   1     0  Fa0/1      Up
2  0005.dd3e.ce80 141.26.125.1 Fa0/3   1     0  Fa0/3      Up

141.26.124.1#
```

Nach der Ausgabe haben wir den gewünschten Zustand erreicht. Mithilfe der CMS ist es leichter nachzuvollziehen, ob alles wie geplant gelaufen ist. Wie in Abb. 9 zu sehen ist, lässt sich der aktuelle Zustand des Clusters mit der Schaltfläche namens *Front Panel* zeigen.

[16]IP-Adresse 141.26.124.1 als .124.1 verkürzt.

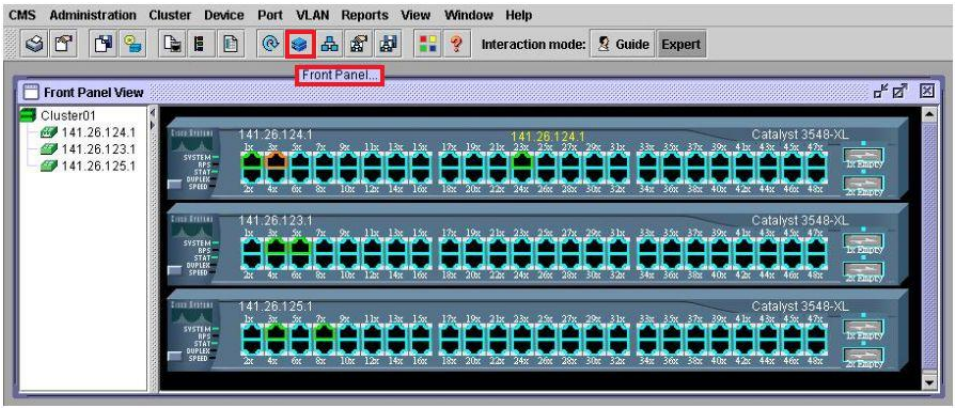

Abbildung 9: Ein Clusters im Front Panel

Mit diesem Panel hat man einen erleichterten Überblick über das Cluster. Zusätzlich ist man in der Lage, Konfigurationen am Command-Switch und an den Member-Switches direkt von hier aus vorzunehmen. Im Panel werden alle Switches mit ihren aktiven (dargestellt grün), passiven bzw. blockierten (dargestellt rot) Ports dargestellt. Die CMS bietet auch weitere Möglichkeiten, das erzeugte Netz zu betrachten bzw. zu kontrollieren. Mit dem Symbol ***Topology*** lässt sich das Ganze auch topologisch aufzeichnen (siehe Abb. 10). Alle Switches werden mit den jeweiligen Verbindungen, den IP-Adressen, Switch-IDs und MAC-Adressen dargestellt. Die bei der Verbindung genutzten Ports werden mit ihren Port-Nummern und Port-Geschwindigkeiten präsentiert.

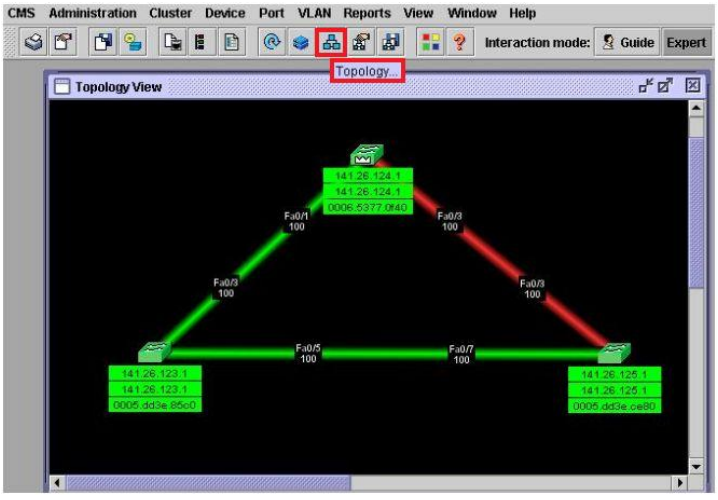

Abbildung 10: Topologisches Aussehen eines Clusters(CMS)

DieVerbindung zwischen 141.26.124.1 (Command-Switch) und 141.26.125.1 wurde rot dargestellt, weil der jeweilige Port 3 auf dem Command-Switch wegen der Schleife im Netz per ***Spanning Tree Protokoll*** (STP) blockiert wurde (siehe Kapitel 5.3).
Nach der Erzeugung des Clusters werden, unter der CMS, noch weitere Aktionen zur Verfügung gestellt:

- Löschen des Clusters
- Hinzufügen weiterer Switches zum Cluster
- Entfernen eines Member-Switches vom Cluster
- Festlegung eines *Standby-Switches*
- Hop-Anzahl in der Topologie etc.

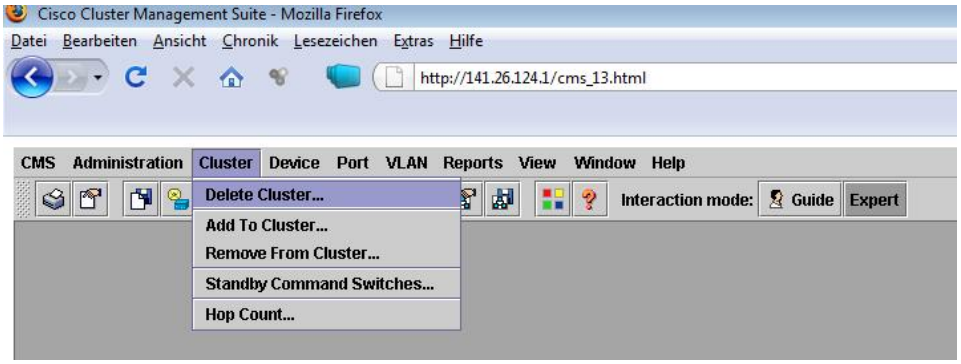

Abbildung 11: Weitere Aktionen in einem Cluster

Jetzt fügen wir noch einen weiteren Switch zu unserem Netz hinzu. Dieser besitzt die IP-Adresse 141.26.122.1 und ist über den Port 5 mit dem Command-Switch verbunden. Nach der Aktualisierung sieht unsere Topologie folgt aus.

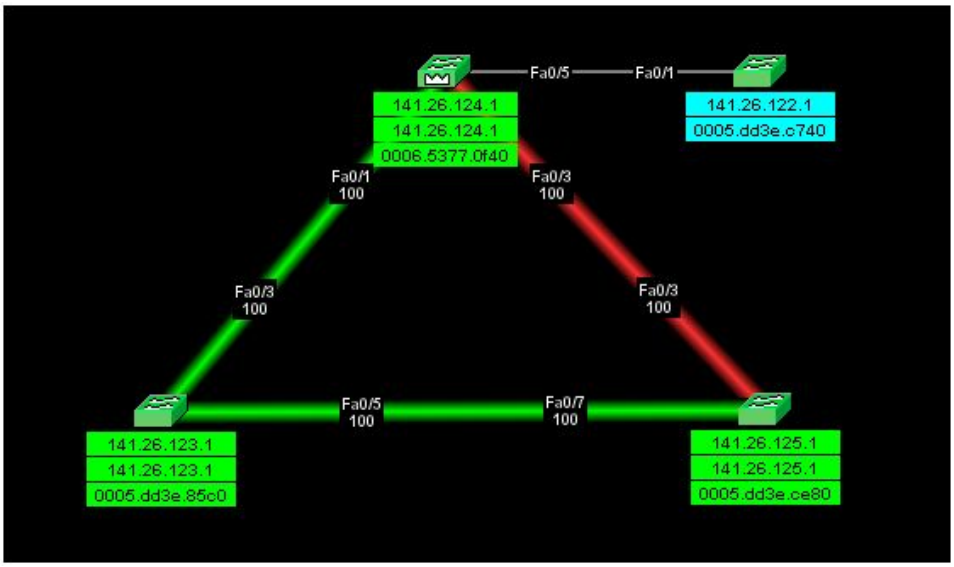

Abbildung 12: Neuer Kandidaten-Switch

Der neue Switch wird mit einem türkis-farbigen Label dargestellt, weil dieser dem Cluster noch nicht zugehört. Man kann ihn wieder durch die oben gezeigte Methode zum Cluster hinzufügen oder einfach mit einem Rechtsklick(Maus) auf den Switch , im kommenden Pop-Up-Menü die Option *Add Cluster* auswählen. Anschließend wird das Passwort für den jeweiligen Switch angegeben. Nach der Aktualisierung wird dieser auch als Mitglied des Clusters angezeigt.
Man entfernt ein Cluster-Mitglied durch das Auswählen der passenden Option im CMS unter dem Menü-Eintrag *Cluster / Remove from Cluster* oder durch die folgenden Befehle per CLI im Command-Switch :

- *show cluster members* : Aktuellen Status des Clusters anzeigen. Hier muss man sich die Member-Nummer (SN) und MAC-Adresse des Switches merken, der vom Cluster entfernt werden soll.

```
141.26.124.1#show cluster members
                                                |--Upstream--|
SN  MAC Address     Name          PortIf  Hops  SN  PortIf    State
0   0006.5377.0f40  141.26.124.1           0                  Up(Cmdr)
1   0005.dd3e.85c0  141.26.123.1  Fa0/3    1     0   Fa0/1    Up
2   0005.dd3e.ce80  141.26.125.1  Fa0/3    1     0   Fa0/3    Up
3   0005.dd3e.c740  141.26.122.1  Fa0/1    1     0   Fa0/5    Up
141.26.124.1#
```

- *configure terminal* : In den Konfigurations-Modus wechseln.

```
141.26.124.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.124.1(config)#
```

- *no cluster member 3* : Switch mit der IP-Adresse 141.26.122.1 (SN im Cluster 3) wird vom Cluster entfernt.

```
141.26.124.1(config)#no cluster member 3
141.26.124.1(config)#end
141.26.124.1#
```

- *show cluster members* : Verifizierung des Clusters nach der Entfernung

```
141.26.124.1#show cluster members
                                          |--Upstream--|
SN  MAC Address    Name          PortIf  Hops  SN  PortIf    State
0   0006.5377.0f40 141.26.124.1           0                 Up (Cmdr)
1   0005.dd3e.85c0 141.26.123.1 Fa0/3     1     0   Fa0/1     Up
2   0005.dd3e.ce80 141.26.125.1 Fa0/3     1     0   Fa0/3     Up
141.26.124.1#
```

5.2 Virtual Local Area Network(VLAN)

Suat Algin

Eine inzwischen wesentliche Technik im Ethernet-Switching ist die logische Aufteilung von physikalischen Netzwerken in sogenannte „Virtual Local Area Networks“(VLANs). Diese logische Aufteilung des physikalischen Netzwerks in VLANs hat viele Vorteile. Wo früher Clients, die nicht miteinander kommunizieren durften, physikalisch voneinander getrennt werden mussten, kann man heute einfach das physikalische Netz in verschiedene VLANs trennen und so die Kommunikation zwischen bestimmten Stationen untersagen. Durch diese einfachen Mittel ist man im Stande in einem Unternehmen Abteilungen voneinander zu trennen oder ein Intranet mit und ein Intranet ohne eine Internetanbindung aufzubauen.

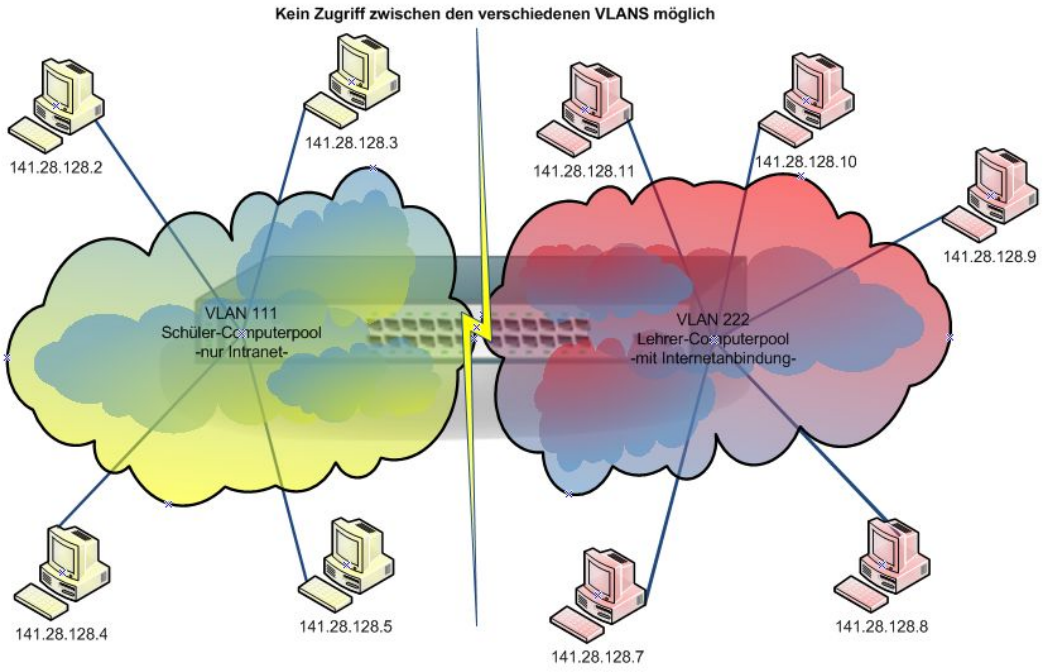

Abbildung 13: VLAN - Intranet mit und ohne Internetanbindung

Diese Technik unterteilt die verschiedenen Stationen in logische Gruppen, so dass die Geräte nur noch innerhalb der Gruppe kommunizieren können. So erhält man eine Topologie, die aus mehreren voneinander getrennten LANs besteht.

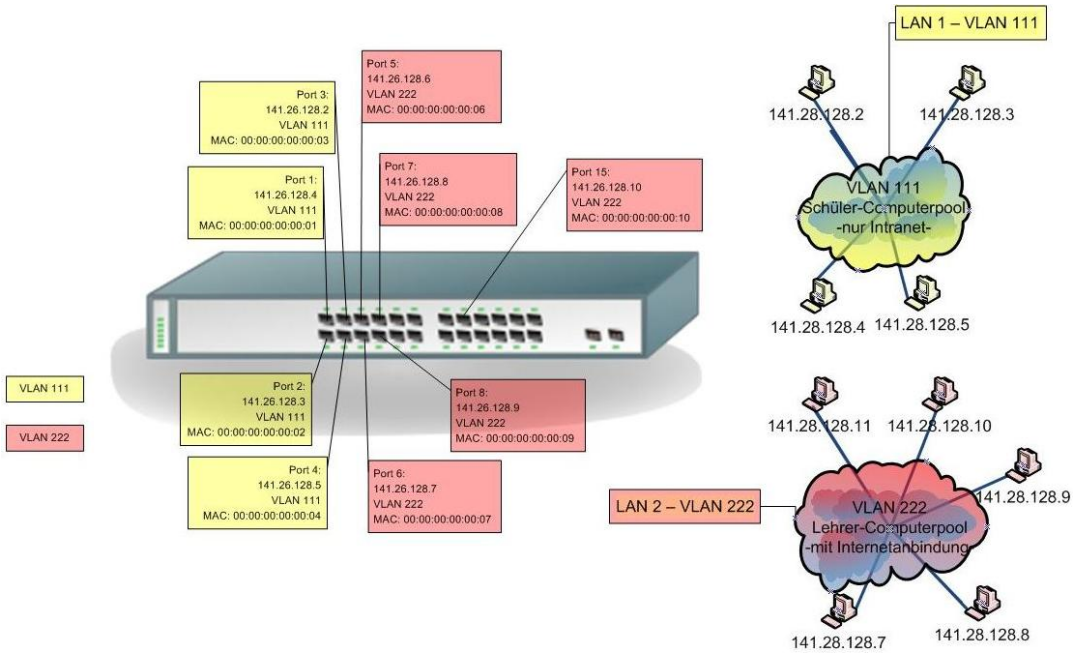

Abbildung 14: VLAN-Zuweisung anhand der MAC-Adressen
(Dynamic Port Single VLAN Membership)

Um ein Netzwerk-Gerät einem bestimmten VLAN zuzuweisen, existieren zwei Ansätze. Einerseits kann man ein Gerät anhand seines Standorts einem bestimmten VLAN zuordnen, andererseits ist es aber auch möglich die Stationen anhand ihrer MAC-Adressen in ein VLAN aufzunehmen. Bei den heutigen Anforderungen in einem Unternehmen ist die Möglichkeit, VLAN-Zuordnungen anhand der MAC-Adresse vorzunehmen, unerlässlich. So kann man zum Beispiel die VLAN-Zuordnungen in einer WLAN-Umgebung per MAC-Adresse vornehmen und im Gegensatz dazu die Büro-Station anhand ihres physikalischen Standorts zuweisen.(Abb. 14)
So ist man im Stande mobile Netzwerkgeräte wie z.B. Notebooks, die ihren Standort wechseln, anhand Ihrer MAC-Adressen zu identifizieren und die Zuweisung in ein bestimmtes VLAN vorzunehmen. In modernen Switches, so auch im Catalyst 3548 XL, ist der Administrator in der Lage, Multi-VLANs zuzuweisen. Diese Multi-VLANs erlauben den jeweiligen Clients den Zugriff auf zwei verschiedene VLANs. Diese Art der VLAN Zuordnung wird in vielen Unternehmen dazu genutzt, um z.B. Abteilungsleitern oder Teamleitern den Zugriff auf das Firmeninterne Intranet zu gewähren, aber auch gleichzeitig den Zugriff auf Abteilungsübergreifende Netze zu erlauben. So haben die „normalen“ Mitarbeiter nur Zugriff auf das Abteilungsinterne Intranet, ohne dabei auf die Netze der höher gelegenen Abteilungen zuzugreifen.
Die Anzahl der VLANs, die man in einem Switch höchstens einstellen kann, hängt vom

Hersteller und Modell des Switches ab. Dabei kann diese Zahl sehr variieren. So sind z.B. bei einem Catalyst 2900 XL mit 8 Megabyte und einem festen Modul nur 64 VLANs zulässig. Im Gegensatz dazu sind bei einem Catalyst 2900 XL mit erweiterbaren Modulen und im Catalyst 3548 XL schon 250 VLANs erlaubt. Diese Anzahl der zulässigen VLANs wird durch den Einsatz des Spanning Tree Protokolls limitiert. Diese Limitierung gilt nur, wenn in allen VLANs das Spanning Tree Protokoll zum Einsatz kommen soll. Die maximale Anzahl an einsetzbaren STPs pro Switch beträgt nämlich 64. Wenn der Administrator größere Netzwerke verwaltet und somit der Einsatz des STP unerlässlich ist, muss er mit 64 VLANs auskommen. Für den Fall, dass erman das STP nicht einsetzt, kann man zwar bis zu 250 VLANs aufbauen, riskiert aber gleichzeitig Ausfälle in seinem VLAN. Diese Ausfälle können z.B. durch Loops entstehen. (Abb. 15)
Eine Ausnahme bilden die in Kapitel 5.4 vorgestellten Etherchannel-Ports. Diese können nämlich mehrere Verbindungen zu einer virtuellen Verbindung zusammenfassen und so die Loop-Erzeugung verhindern. Zusätzlich bieten Sie eine gewisse Ausfallsicherheit.
Im Fall eines oder mehrerer Loops werden die Switches überlastet und das Netzwerk bricht zusammen. So muss der Administrator entscheiden, ob er dieses Risiko eingehen möchte oder ob er sein Netz auf diese 64 VLANs begrenzt.

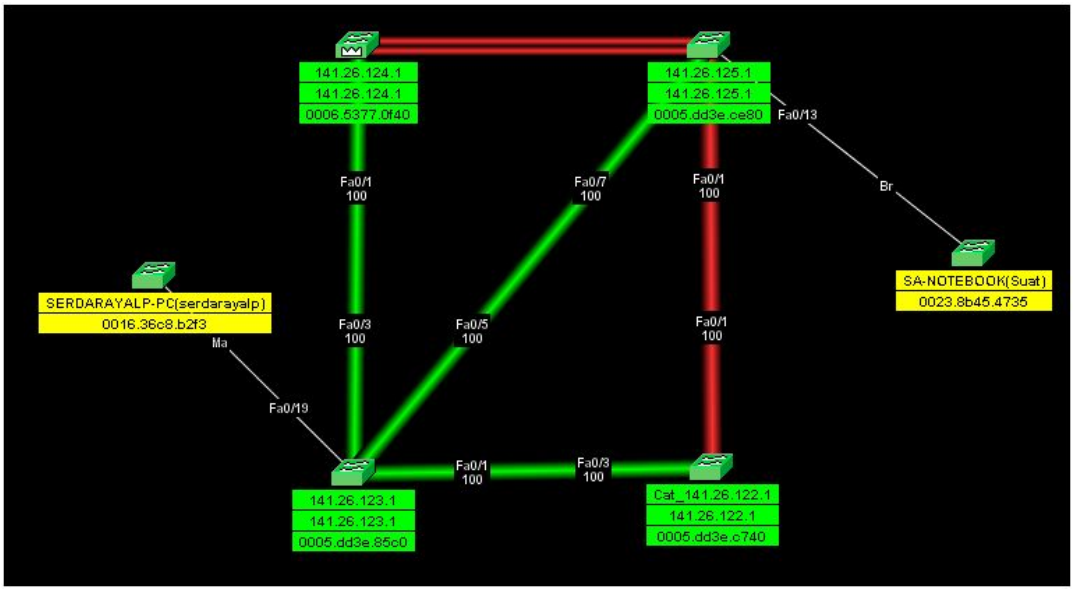

Abbildung 15: Verhinderung von Loops durch das STP

5.2.1 VLAN: Port-Konfiguration

Suat Algin

Wie schon erwähnt sind in unserem Fall 250 VLANs ohne das Spanning Tree Protokoll zulässig. Dabei werden die VLANs anhand Ihrer ID identifiziert. Diese ID besteht aus einer Zahl zwischen 1 und 1001.
Bevor man ein Port einem bestimmten VLAN zuweist, muss man sich Gedanken über die Art eines Ports machen.
Hierbei gelten folgende Einschränkungen für die verschiedenen Port-Arten:

- **Statische Ports** können nur zu einem VLAN gehören und werden vom Administrator manuell zugewiesen. Standardmäßig sind alle Ports, statische Ports und dem VLAN 1 zugewiesen. Dieses VLAN 1 ist das Verwaltungs-VLAN und wird zur Konfigurierung des Switches genutzt.

- **Multi-VLAN Ports** können bis zu 250 VLANs gehören. Der Verkehr auf diesem Port kann mit mehreren VLANs ausgetauscht werden. Ein Port kann kein Multi-VLAN Port sein, sobald *trunking* auf diesem aktiviert wird.

- **Trunk Ports** gehören standardmäßig zu allen VLANs, können aber durch entsprechende Einträge in der VLAN-Datenbank auf bestimmte VLANs begrenzt werden. Durch die Modifizierung der „pruning-eligible list“ kann der traffic, der über einen Trunk Port an ein VLAN gesendet wird, reduziert werden. Das Virtual Trunk Protokoll(VTP) verwaltet das Hinzufügen, Löschen oder Umbenennen von VLANs im ganzen Netzwerk. Dabei tauscht VTP Konfigurations-Informationen zwischen Switches über die Trunk-Ports aus.

Die Konfiguration eines VLANs in der CMS erfolgt über das Menü VLAN/Configure VLAN. Dies ist aber nur möglich, falls der Switch sich im VTP-Modus „Server“ oder „Transparent“ befindet(Kapitel 5.2.2).

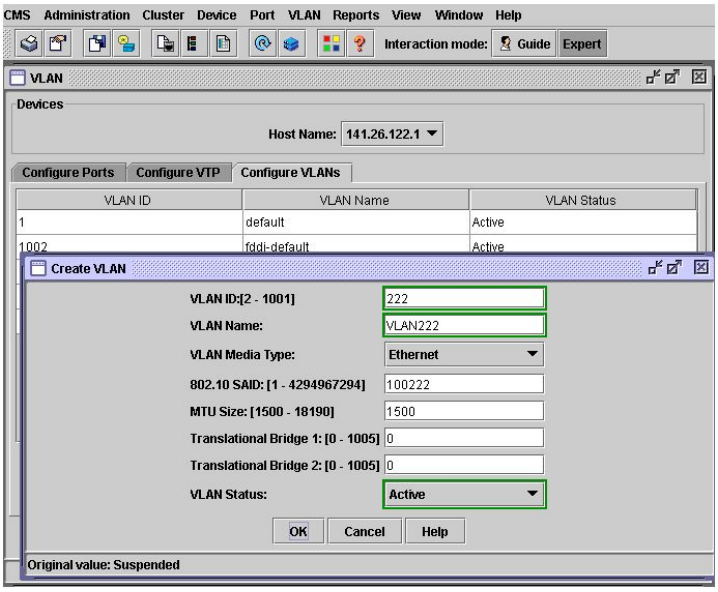

Abbildung 16: VLAN Konfiguration(CMS)

Die VLAN-Konfiguration über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#
141.26.122.1#vlan database
141.26.122.1(vlan)#vlan 222 name VLAN222
VLAN 222 added:
    Name: VLAN222
141.26.122.1(vlan)#
```

Die verschiedenen Konfigurationsoptionen im CLI:

```
141.26.122.1(vlan)#vlan 222 ?
  are         Maximum number of All Route Explorer hops
                                          for this VLAN
  backupcrf   Backup CRF mode of the VLAN
  bridge      Bridging characteristics of the VLAN
  media       Media type of the VLAN
  mtu         VLAN Maximum Transmission Unit
  name        Ascii name of the VLAN
  parent      ID number of the Parent VLAN of FDDI or
                                          Token Ring type VLANs
  ring        Ring number of FDDI or Token Ring type VLANs
```

```
  said        IEEE 802.10 SAID
  state       Operational state of the VLAN
  ste         Maximum number of Spanning Tree Explorer hops
                                        for this VLAN
  stp         Spanning tree characteristics of the VLAN
  tb-vlan1    ID number of the first translational VLAN
                                        for this VLAN (or zero if none)
  tb-vlan2    ID number of the second translational VLAN
                                        for this VLAN (or zero if none)
  <cr>
141.26.122.1(vlan)#exit
APPLY completed.
Exiting....
141.26.122.1#
```

Zuweisung eines Ports zu einem VLAN:

```
141.26.122.1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#interface fa0/2
141.26.122.1(config-if)#switchport mode access
141.26.122.1(config-if)#switchport access vlan 222
141.26.122.1(config-if)#end
```

Verifizierung der Einstellungen:

```
141.26.122.1#show interface fa0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 222 (VLAN222)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: NONE
Pruning VLANs Enabled: NONE

Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
Self Loopback: No
141.26.122.1#
```

5.2.2 Virtual Trunk Protocoll - VTP

Suat Algin

Das VLAN Trunk Protocol(VTP) ist ein Layer 2 Protokoll und wird zur zentralen Konfiguration von Switch-übergreifenden VLANs genutzt. Es kommt zum Einsatz, wenn über einen Link zwischen zwei Switches mehrere VLANs miteinander kommunizieren müssen(Abb. 17). Diese Links zwischen den Switches werden Trunk-Links genannt und können im CMS über das Menü VLAN/Configure Ports/Modify für jeden Port einzeln festgelegt werden(Abb. 21). Diese VTP Pakete werden entweder durch die Cisco eigene ISL-Methode oder über die standardisierte IEEE 802.1Q-Methode gesendet. Dabei wird bei der ISL-Methode das VTP-Paket in einen ISL-Frame integriert. Im Gegensatz dazu wird bei der IEEE 802.1Q-Methode zusätzlich zum VTP-Paket, jedem Frameheader jedes Paketes eine Marke(Tag) hinzugefügt, welche Informationen über die VLAN Zugehörigkeit enthält. Anhand dieser Information wird das Paket an den entsprechenden Switch weitergeleitet. Der letzte Switch vor dem Empfänger entfernt dann diese Marke und das VTP-Paket stellt das Original-Paket zu. Dies muss geschehen, da der Empfänger nichts mit diesen zusätzlichen Informationen anfangen kann.

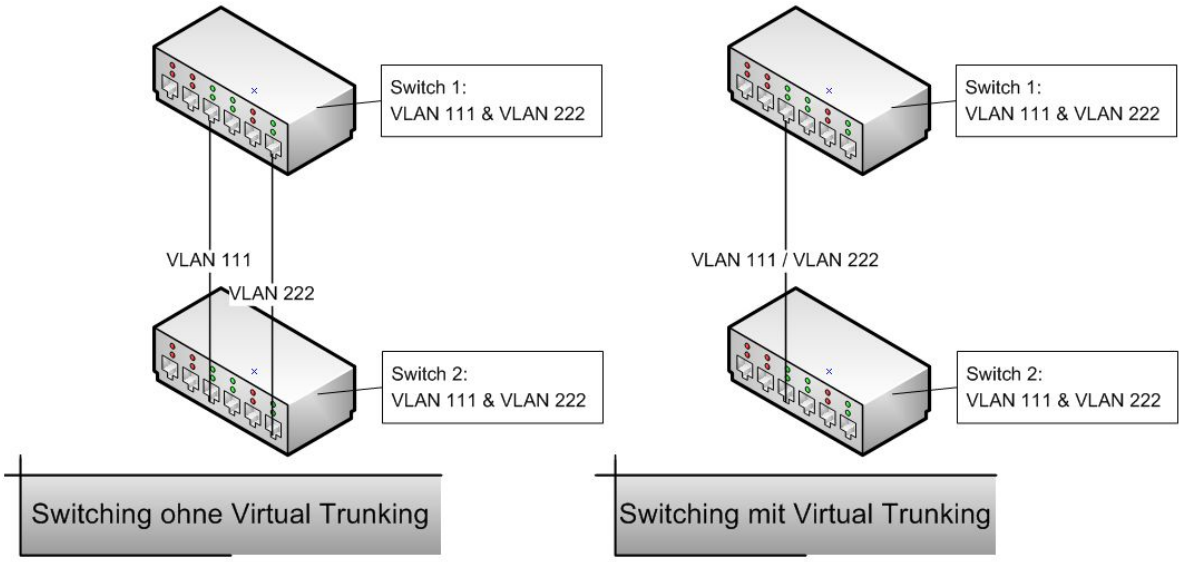

Abbildung 17: Virtual Trunking

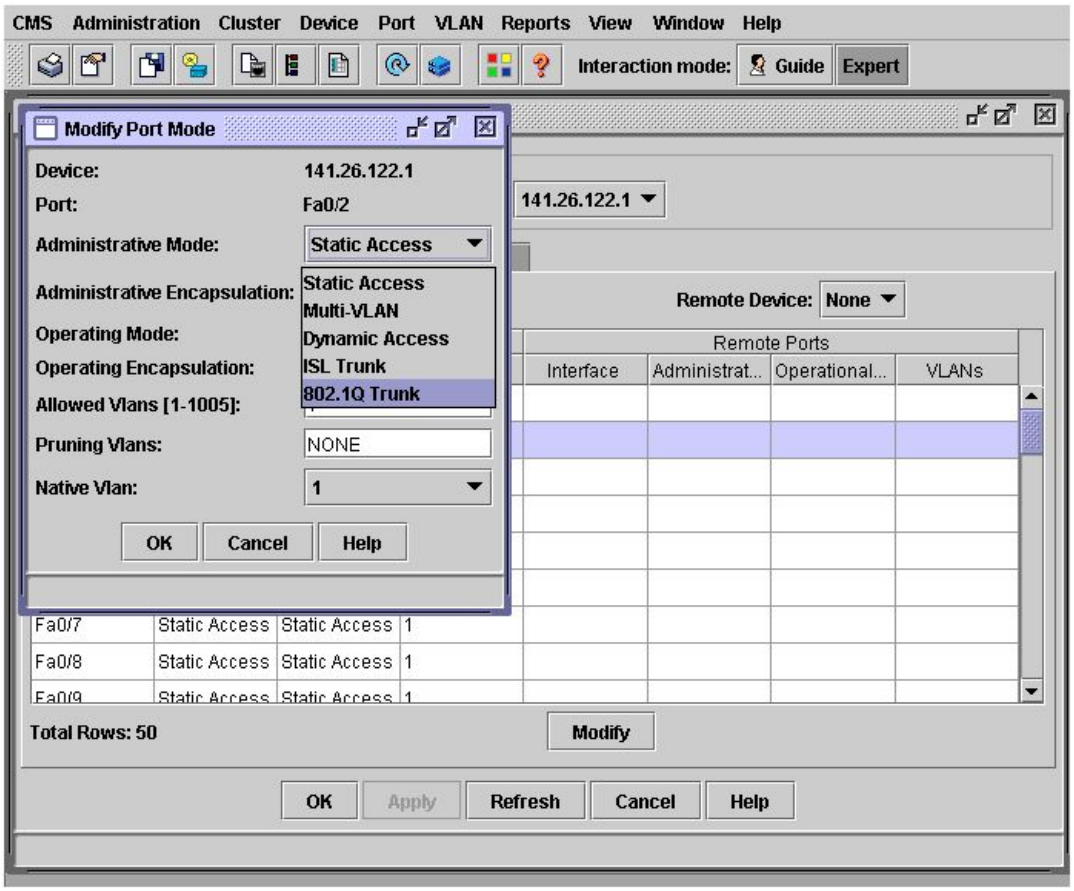

Abbildung 18: Konfigurierung des Trunkports(CMS

Konfigurierung des Trunkport über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#interface fa0/2
141.26.122.1(config-if)#switchport mode trunk
141.26.122.1(config-if)#switchport trunk encapsulation dot1q
141.26.122.1(config-if)#end
141.26.122.1#show interface fa0/2 switchport
Name: Fa0/2
Switchport: Enabled
```

```
Administrative mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: 1,1002-1005
Trunking VLANs Active: 1
Pruning VLANs Enabled: NONE

Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
Self Loopback: No
141.26.122.1#
```

Durch die Nutzung von VTP-Paketen werden Misskonfigurationen oder Inkonsistenzen im gesamten Netzwerk vermieden. Per VTP können VLAN-Änderungen zentral von einem Switch aus vollzogen werden. Die Änderungen werden dann allen Switches im Netz offeriert. Dabei werden Informationen an eine vorreservierte Multicast-Adresse gesandt und dann von den benachbarten Switches abgeholt, um ihre VTP und VLAN Konfigurationen bei Bedarf zu ändern. Ohne VTP können keine VLAN Informationen zwischen Switches ausgetauscht werden. Um VTP nutzen zu können, wird bei der Konfiguration der Switches ein VTP-Domainname ausgesucht und die VTP Methode festgelegt. Ein Switch in einem Netzwerk darf nur zu einer VTP-Domain zugeordnet werden. Standardmäßig befindet sich ein Switch im „no-management-domain“-Status bis es entweder VTP-Domain Informationen über einen Trunkport erhält oder manuell einen VTP-Domainnamen zugewiesen bekommt. Der „default“ VTP-Modus ist zwar der Server-Modus, doch die VLAN Informationen werden solange nicht an das Netzwerk weitergegeben, bis ein VTP-Domainname manuell zugewiesen oder über einen Trunk-Link gelernt wird. Die manuelle Vergabe des VTP-Domainnamens erfolgt über das Menü VLAN(Abb. 19).

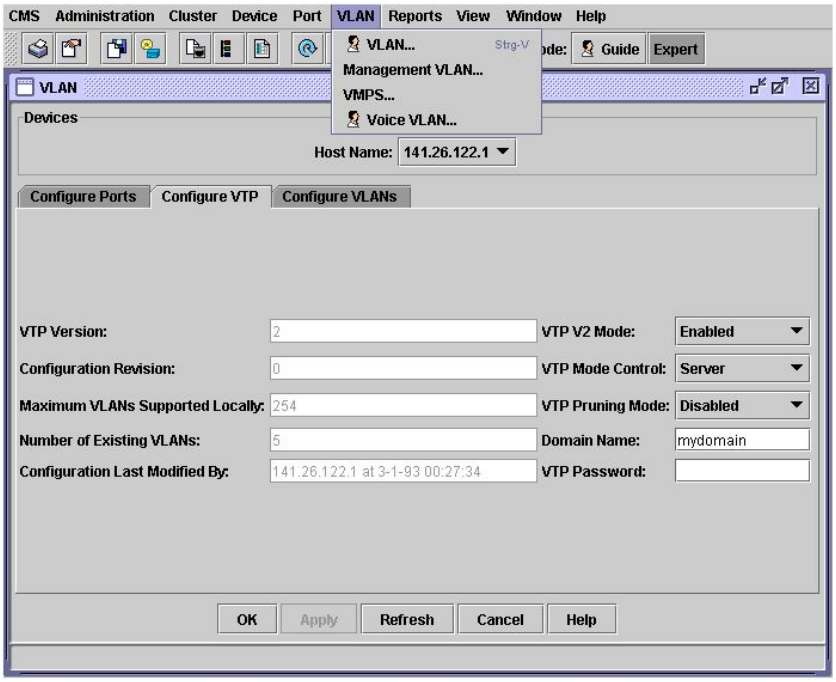

Abbildung 19: VTP

Falls der Domainname nicht manuell eingegeben wird, übernimmt der Switch den Domainnamen, welchen es über einen Trunkport mitgeteilt bekommt. Zusätzlich zu diesem Domainnamen erhält der Switch unter anderem eine Revisionsnummer, über die er die Aktualität seiner VTP-Konfiguration überprüfen kann. So werden daraufhin Umkonfigurationen über andere VTP-Domainen oder durch VTP-Pakete mit jüngeren Revisionsnummern ignoriert.
Bei der Konfiguration des VTP auf einem Switch existieren drei verschiedene Modis:

- **VTP-Server Modus**: Erstellung, Änderung, Löschung von VLANs und Domainweite Konfiguration bzw. Synchronisierung von VTP-Parametern.
- **VTP-Transparent Modus**: Erstellung, Änderung, Löschung von VLANs, aber keine Veröffentlichung an die VTP-Domain und keine Änderungen durch andere VTP-Teilnehmer.
- **VTP-Client**: Keine eigene Modifikation von VLANs möglich. Änderungen nur über einen im VTP-Server Modus befindlichen Switch.

VLAN-Änderungen von einem Switch im VTP-Server Modus werden über alle Trunk Verbindungen an alle Switches in der VTP-Domain propagiert und beeinflussen so die anderen Domainen-Teilnehmer. Im Gegensatz dazu werden Änderungen an einem im

VTP-Transparent Modus befindlichen Switch nicht an das Netzwerk weitergegeben. Es werden nur Informationen weitergegeben, die der Switch selbst von anderen Domainen-Teilnehmern erhalten hat. Im VTP-Client Modus kann der Switch keine VLANs erstellen, löschen oder ändern und auch keine VTP Modifikationen vornehmen. Ansonsten verhält sich ein Switch in diesem Modus, wie ein Switch im VTP-Server Modus.
Switches im VTP-Server oder im VTP-Client Modus können in Sonderfällen Ihren Modus automatisch in den VTP-Transparent Modus wechseln, um die Konsistenz des Netzwerks zu wahren. Dies ist der Fall wenn zum Beispiel mehr als 64 VLANs auf einem Switch konfiguriert werden oder wenn ein Port zu einem Multi-VLAN Port umkonfiguriert wird.
VTP-Pakete beinhalten Informationen über den VTP-Domainnamen, die VTP Revisionsnummer, die VLAN-ID, den VLAN Namen und abhängig vom VLAN Typ bestimmte Informationen über die VLAN Konfiguration. Um die Menge der durch das Netzwerk wandernden Informationen zu begrenzen, wir das VTP-Pruning eingesetzt. Beim VTP-Pruning werden die Informationen nur an die Trunk-Ports weitergeleitet, die nötig sind um das Ziel zu erreichen. So wird vermieden, dass Broadcast-, Multicast- oder unbekannte Unicast-Pakete an alle Trunkports in einer VTP-Domain weitergeleitet werden und so auch unbeteiligte Switches mit unnötigem Traffic belasten. Dies wird dadurch erreicht, dass beim Pruning Pakete von unzulässigen VLANs blockiert werden. Diese Blockierung erfolgt über die „pruning-eligible list“. Die defaultmäßig ausgeschaltete Funktion erlaubt den Austausch von Traffic für die in seiner Liste befindlichen VLANs. Ein Beispiel für das VTP-Pruning wird in der nächsten Abbildung aufgezeigt. Hier wird der Verkehr von Switch 4 empfangen und nur an Switch 6 weitergeleitet, um nicht das restliche Netzwerk mit diesem Traffic zu belasten.

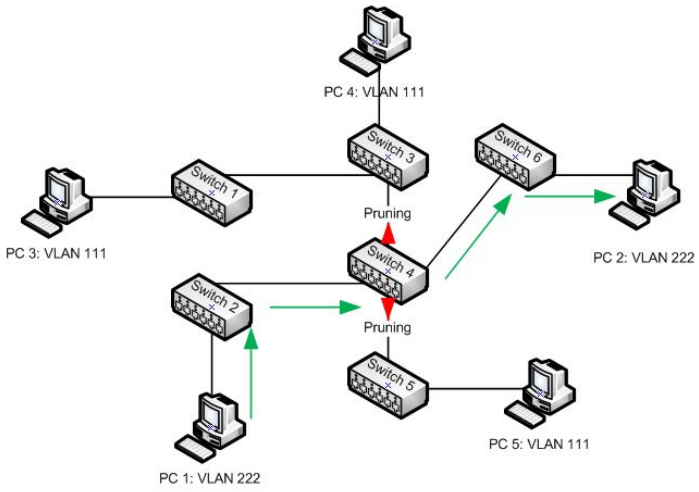

Abbildung 20: Pruning

Die Konfiguration über die CMS erfolgt über das in Abbildung 19 gezeigte Menü und ist selbsterklärend.
Im folgenden wird die Konfiguration über das CLI erklärt.

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#vlan database
141.26.122.1(vlan)#
```

Einen Domainnamen setzen:

```
141.26.122.1(vlan)#vtp domain mydomain
Changing VTP domain name from null to mydomain
141.26.122.1(vlan)#
```

Ein Passwort für die Domain festlegen:
Achtung: Das Passwort muss an jedem Switch in der Domain gesetzt werden, da ansonsten die VTP Nachrichten abgewiesen werden.

```
141.26.122.1(vlan)#vtp password xxx
Setting device VLAN database password to xxx.
141.26.122.1(vlan)#
```

Den VTP-Modus modifizieren:

```
141.26.122.1(vlan)#vtp server
Setting device to VTP SERVER mode.
141.26.122.1(vlan)#
```

Das VTP-Pruning einschalten:

```
141.26.122.1(vlan)#vtp pruning
Pruning switched ON
141.26.122.1(vlan)#exit
APPLY completed.
Exiting....
141.26.122.1#
```

Die Änderungen verifizieren:

```
141.26.122.1#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 254
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 : mydomain
VTP Pruning Mode                : Enabled
VTP V2 Mode                     : Enabled
```

```
VTP Traps Generation            : Disabled
MD5 digest              : 0x07 0xD4 0x57 0x3C 0xDE 0x89 0x62 0x4C
Configuration last modified by 141.26.122.1 at 3-1-93 00:27:34
141.26.122.1#
```

Anzeige des VTP-Status aller Switches in der VTP-Domain: In diesem Zusammenhang nutzen wir den Befehl „rcommand“, um über einen Switch auf alle anderen in der Domain befindlichen Switches zuzugreifen. So kann man die Informationen der anderen Switches auslesen, ohne direkt mit diesem verbunden zu sein.

```
141.26.124.1#show vtp status
VTP Version                     : 2
Configuration Revision          : 2
Maximum VLANs supported locally : 254
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : mydomain
VTP Pruning Mode                : Enabled
VTP V2 Mode                     : Enabled
VTP Traps Generation            : Disabled
MD5 digest              : 0x79 0x42 0x6F 0x9E 0x3D 0xBC 0x50 0x48
Configuration last modified by 141.26.124.1 at 3-1-93 02:51:51
141.26.124.1#
141.26.124.1#rcommand mac-address 0005.dd3e.ce80
Trying ... Open

141.26.125.1#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 254
Number of existing VLANs        : 7
VTP Operating Mode              : Client
VTP Domain Name                 : mydomain
VTP Pruning Mode                : Enabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest              : 0x50 0x40 0x2F 0x55 0x28 0xF6 0x54 0x49
Configuration last modified by 141.26.125.1 at 3-1-93 00:54:11
141.26.125.1#exit

[Connection closed by foreign host]
141.26.124.1#rcommand mac-address 0005.dd3e.c740
Trying ... Open

141.26.122.1#show vtp status
VTP Version                     : 2
Configuration Revision          : 3
Maximum VLANs supported locally : 254
Number of existing VLANs        : 6
VTP Operating Mode              : Server
VTP Domain Name                 : mydomain
```

```
VTP Pruning Mode                    : Enabled
VTP V2 Mode                         : Enabled
VTP Traps Generation                : Disabled
MD5 digest                 : 0xB4 0x81 0x31 0x1B 0xC9 0xBE 0x9C 0x4B
Configuration last modified by 141.26.122.1 at 3-1-93 02:21:52
141.26.125.1#exit

[Connection closed by foreign host]
141.26.124.1#
```

Standardmäßig empfängt ein Trunk Port den Traffic von allen in der VLAN Datenbank befindlichen VLANs und leitet diesen auch zu allen bekannten VLANs weiter. So wird ein Trunk Port Mitglied jedes erlaubten VLANs. Diese Eigenschaft kann aber durch den Verbot für bestimmte VLANs aufgehoben werden.
Dies wird folgendermaßen konfiguriert:

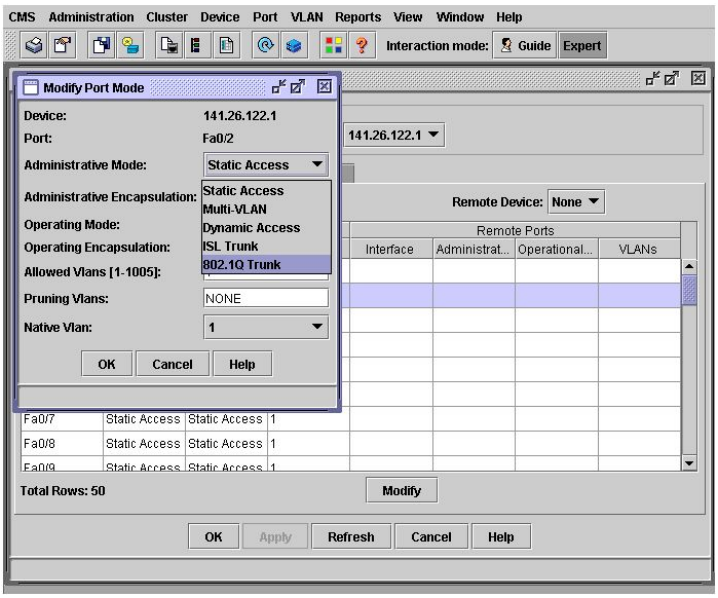

Abbildung 21: Trunkport

Konfigurierung über das CLI:

```
141.26.122.1(config)#interface fa0/2
141.26.122.1(config-if)#switchport mode trunk
141.26.122.1(config-if)#switchport trunk allowed vlan add 222
141.26.122.1(config-if)#switchport trunk allowed vlan remove 333
141.26.122.1(config-if)#end
141.26.122.1#show interface fa0/2 switchport allowed-vlan
"1,222,1002-1005"
```

```
141.26.122.1#copy running-config startup-config
Destination filename [startup-config]? y
141.26.122.1#
```

Der Parameter „pruning-eligible list“ setzt fest, welche VLANs Paket-Flutungen empfangen dürfen. Dieser Parameter wird nur für Trunk Ports gesetzt.
Einstellung der „pruning-eligible list“:

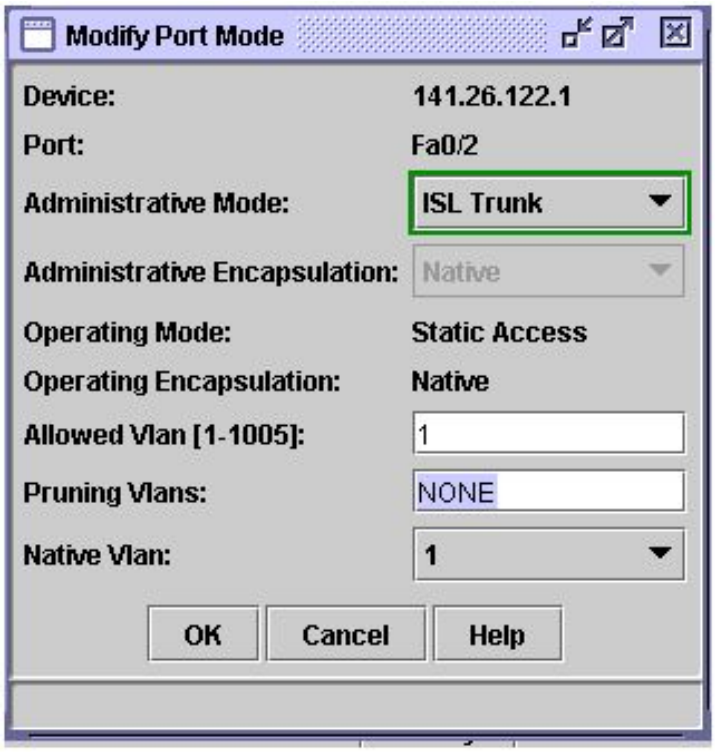

Abbildung 22: Pruning via CMS

Konfigurierung über das CLI:

```
141.26.122.1(config)#interface fa0/2
141.26.122.1(config-if)#switchport trunk pruning vlan  add 222
141.26.122.1(config-if)#switchport trunk pruning vlan  remove 222
141.26.122.1(config-if)#end
141.26.122.1#show interface fa0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative mode: trunk
Operational Mode: trunk
```

```
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: 1,222,1002-1005
Trunking VLANs Active: 1,222
Pruning VLANs Enabled: 222

Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
Self Loopback: No
141.26.122.1#
```

Das „Native VLAN“ auf einem Trunk Port leitet alle „untagged Frames“ weiter. Standardmäßig ist das „Native VLAN“ das VLAN 1. Es kann aber auch gefahrlos auf ein anderes VLAN gelegt werden, da es nicht vom „Management VLAN“ abhängt. Falls ein weiterzuleitendes Paket die VLAN-ID des „Native VLAN“ trägt, wird es ungetagged übertragen. Falls dies nicht der Fall ist, fügt der Switch ein „Tag“ hinzu. Die Konfigurierung über das CMS erfolgt über das in Abbildung 22 gezeigte Menü.
Konfigurierung über das CLI:

```
141.26.122.1(config)#interface fa0/2
141.26.122.1(config-if)#switchport trunk native vlan 1
141.26.122.1(config-if)#end
141.26.122.1#copy running-config startup-config
Destination filename [startup-config]? y
141.26.122.1#
```

Das sogenannte „Load Sharing“ nutzt die Eigenschaften von Trunk Ports um mehrere Verbindung zu einem Switch aufrecht zu erhalten. Dabei wird verhindert, dass das Spanning Tree Protokoll einen oder mehrere Ports sperrt, um ein Loop zu verhindern. Dies geschieht über die STP Port-Prioritäten.
So kann man z.B. bei einer doppelten Verbindungen zu einem Gerät, die STP Port-Priorität auf einem der Verbindungen für ein VLAN(VLAN 222) mit einem niedrigen Wert belegen und auf der anderen Verbindung einen höheren Wert festlegen. Umgekehrt belegt man die Priorität eines anderen VLAN(VLAN 333) auf der zweiten Verbindung mit einem niedrigeren Wert als auf der ersten Leitung. So wird die erste Leitung für das VLAN 222 genutzt und die zweite Leitung für das VLAN 333. Umgekehrt werden die Pakete der VLANs auf den Leitungen mit dem höheren Wert blockiert. Da jedes VLAN seinen eigenen STP-Baum aufbaut, ensteht bei diesem Beispiel kein Loop.
Die Konfigurierung über die CMS erfolgt über die in Kapitel 5.3 gezeigten Menüs.

Die Einstellung über das CLI sieht wie folgt aus:

```
141.26.122.1(config)#interface fa0/3
141.26.122.1(config-if)#switchport mode trunk
141.26.122.1(config-if)#spanning-tree vlan 222 port-priority 10
141.26.122.1(config-if)#spanning-tree vlan 333 port-priority 100
141.26.122.1(config-if)#exit
141.26.122.1(config)#interface fa0/7
141.26.122.1(config-if)#switchport mode trunk
141.26.122.1(config-if)#spanning-tree vlan 222 port-priority 100
141.26.122.1(config-if)#spanning-tree vlan 333 port-priority 10
141.26.122.1(config-if)#end
141.26.122.1#copy running-config startup-config
Destination filename [startup-config]? y
141.26.122.1#
```

Diese Selektierung der verschiedenen Verbindungen für bestimmte VLANs kann auch analog anhand der Port-Kosten vorgenommen werden. Auf diese Konfigurierungen wird nicht weiter eingangen, da die in dem Beispiel „Port-Priority“ vorgestellten Befehle denen der Port-Kosten gleich zu setzen sind. Der einzige Unterschied ist, dass der Befehlsteil „port-priority“ mit dem Befehl „cost“ ersetzt wird.

5.3 Spanning Tree Protokoll

Suat Algin

Ein Switch dient zur Segmentierung eines Netzwerks in kleinere *Collision Domains* und damit zur Effizienzsteigerung des Netzes. Er ist als eine Weiterentwicklung einer Bridge zu betrachten. In der Technik wird er als Multiport-Bridge bezeichnet und zeigt auch ähnliche Eigenschaften. Ein Switch stellt durch eine sogenannte ***Backplain*** (*der interne Bus*) mehr Ports als eine Bridge zur Verfügung, die zusätzlich auch leistungsfähiger sind als die einer Bridge[16].

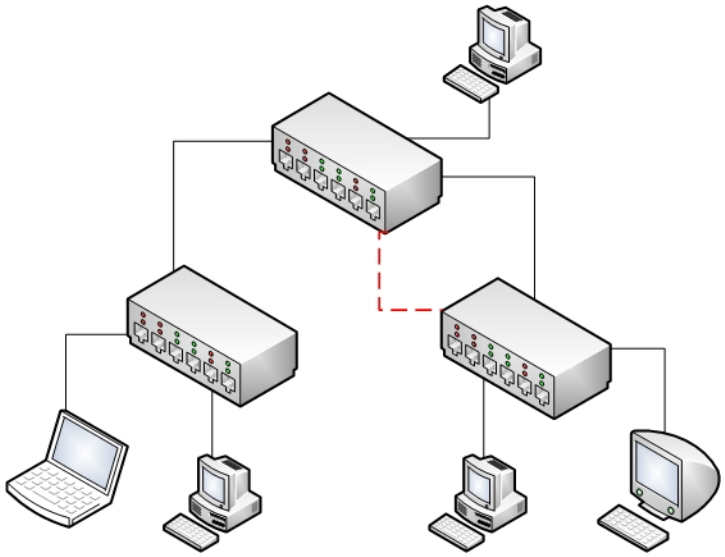

Abbildung 23: Ein Netz ohne redundante Pfade

Redundante Verbindungen in einem Netz sind besonders wichtig, weil damit Netzwerke fehlertoleranter werden. Andererseits wird aber durch diese Redundanz in der Topologie die Möglichkeit für die fehlerhafte Doppelübertragungen von Informationen möglich.
In der Abb. 23 wird ein optimales und kostengünstiges Netz vorgestellt, weil das Netz keine redundanten Pfade zwischen den Switches beinhaltet (ohne roten Link). So ein Netz hat aber den Nachteil, dass nach einem Ausfall einer Leitung das ganze Teilnetz lahm gelegt werden kann. Um das zu verhindern, werden redundante Links zum Netz hinzugefügt (mit dem roten Link). In diesem Fall besteht die Gefahr, dass die Switches Broadcasts anderer Switches wegen den Schleifen mehrfach bekommen (*Broadcast-Sturm*). Auf diese Weise wird auch das ganze Netz lahm gelegt. Was uns da eine Hilfe bietet, ist das ***Spanning Tree Protokoll***, welches für das dynamische Erkennen von Schleifen in einem Netz zuständig ist. Mit dem STP wird das gesamte Netz als Baum

dargestellt. Die angeschlossenen Stationen bilden die Blätter und die Switches die Äste des Baums.
Die Erkennung der Schleifen erfolgt mit Hilfe von *Bridge Packet Data Unit* (BPDU)-Paketen, die alle zwei Sekunden ausgesandt werden und zum Aufbau eines Baums dienen. Ein Baum hat in sich keine Schleifen und die Wurzel des Baums wird Root-Bridge bzw. -Switch genannt.
Root-Switch wird der Switch im Netz, der die niedrigste Switch-ID hat.
Eine Switch-ID setzt sich zusammen aus :

- Priorität (per Admin. konfigurierbar und hat den *Default-Wert* **32768**)
- MAC-Adresse des Switches

Je geringer die Zahl für die Priorität, desto höher ist die Priorität.

5.3.1 Aufbau des Baums

Suat Algin

Beim Start verhält sich jeder Switch als Root-Switch und schickt BPDU-Pakete aus. Ein Switch stoppt die Aussendung seiner eigenen Pakete, wenn er ein Paket von einem anderen Switch empfängt, der eine höhere Priorität hat. Anschließend werden nur die empfangenen Pakete vom anderen Switch ausgesandt und nicht mehr die eigenen Pakete. Bevor die Pakete ausgesandt werden, sind auch die angegebenen Port-Kosten zu den Kosten im Paket zu addieren.Diese setzt sich aus den gesamten Linkkosten vom Root-Switch bis zum aktuellen Punkt zusammen. Am Ende dieses Prozesses bleibt nur ein Switch übrig, der als Root-Switch gekennzeichnet wird [16]. Ab diesem Punkt ist nur der Root-Switch für die Erstellung von BPDU-Paketen zuständig. Ein BPDU-Paket[17] enthält folgende Informationen :

- Switch-ID des Absenders
- Switch-ID des Root-Swiches
- Summe der Kosten aller Verbindungen vom Root-Switch bis zur aktuellen Position.

Weil am Ende nur ein Switch (Root-Switch) BPDU-Pakete senden wird, müssen alle anderen Non-Root-Switches wissen, welche Pakete vom Root-Switch empfangen wurden, damit sie auch direkt weitergesandt werden können. Dafür ist eine Bestimmung der sogenannten ***Root-Ports (RP)*** für jeden Non-Root-Switch notwendig. Weil ein

[17] *Aufbau eines BPDU-Konfigurations-Frames* : siehe Kapitel 5.3.2

Non-Root-Switch viele BPDU-Pakete von seinen Interfaces mit unterschiedlichen Kosten zum Root-Switch bekommt, wird derjenige Port Root-Port, der die niedrigsten Kosten zum Root-Switch aufweist. Falls Port-Kosten manuell vergeben werden ist zu beachten, dass durch eine falsche Kosten-Vergabe eine 1GBit-Leitung geblockt werden und alles über eine 100MBit-Leitung gesandt werden könnte. Wurden per Administrator keine Port-Kosten vergeben, werden Standard-Werte verwendet(z.B. 1Gbit-Leitung :10, 100MBit-Leitung :100). Der Port mit den höchsten Kosten wird geblockt (***Blocking-Port***). Hiermit werden keine Daten ausgesandt bzw. empfangen, bis der Switch diesen Port z.B. nach Ausfall einer Leitung wieder freigibt. Alle vom Root-Switch empfangenen Pakete werden über den (***Designated Port***), welcher die niedrigsten Kosten hat, weitergesandt.
Ein Port im Blocking-Zustand geht nicht einfach in den Forwarding-Modus, wenn es wieder freigegeben wird. Es durchläuft zuerst bestimmte Zustände. Diese Schritte vom Blocking- bis zum Forwarding-Modus sind [16] [4]:

- **Blocking-Modus:** In diesem Modus werden keine Datenpakete empfangen (außer BPDU-Pakete) und keine MAC-Adressen mehr in die Adresstabelle geschrieben. Mit diesem Modus wird der Forwarding-Modus vorläufig vollständig abgeschaltet. Der Switch-Port verweilt so lange in diesem Modus, bis dem Switch durch ein BPDU-Paket mitgeteilt wird, dass der jeweilige Port wieder aktiviert werden soll. Infolge dessen wechselt der Port seinen Zustand und geht in den *Listening-Modus* über [4].

- **Listening-Modus:** In diesem Zustand werden weder Datenpakete noch Konfigurations-BPDUs weitergeleitet.Der jeweilige Port ist auch in den Switch-Algorithmus einzubeziehen. Nach Ablauf eines Timers wird dieser Modus deaktiviert und der Port geht in den *Learning-Modus* über.

- **Learning-Modus:** In diesem Modus bereitet sich der Port auf die Übermittlung von Datenpaketen vor. Obwohl die Datenpakete in diesem Zustand noch nicht weitergeleitet werden, werden BPDU-Konfiguratations-Pakete weitergeleitet, damit STP seine Arbeit erledigen kann. Nach Ablauf eines Timers geht der Port in den *Forwarding-Modus*, solange keine BPDU-Pakete kommen, die irgendeine Information beinhalten, welche den Port in den Blocking-Zustand versetzen soll.

- **Forwarding-Modus:** Nur in diesem Modus lassen sich Datenpakete weiterleiten. Zusätzlich werden die BPDU-Pakete weiter ausgewertet. Diesen Modus verlässt der Port so schnell wie möglich, wenn durch ein BPDU-Paket eine Schleifen-Bildung signalisiert wird.

- **Disabled-Modus:** Nur durch den Eingriff des Administrators erreicht ein Port diesen Modus. Es werden weder Datenpakete noch BPDU-Pakete empfangen bzw. weitergesandt. Bei der Berechnung des aktiven Pfads werden Ports nicht berücksichtigt, die sich in diesem Zustand befinden.

Mit folgendem Diagramm werden alle Zustände mit den möglichen Übergängen dargestellt [7].

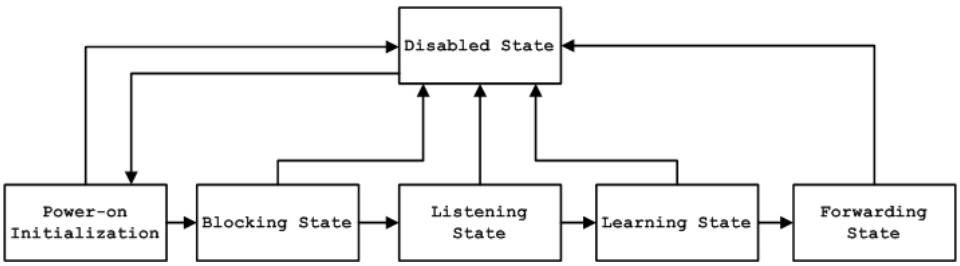

Abbildung 24: Übergänge zwischen den Portzuständen beim STP

5.3.2 Aufbau eines BPDU-Konfigurations-Frames

Suat Algin

Die sogenannten Bridge Protocol Data Units (BPDUs) werden für die Bestimmung der aktuellen Topologie verwendet. BPDUs, die für die Berechnung der Topologie zuständig sind, werden auch als *Configuration BPDU* bezeichnet. Ein Switch nutzt diese empfangenen Pakete, um seine eigenen Konfigurationspakete zu erzeugen und an seine Nachbarn weiterzuschicken.
Wie in Abb. 25 zu sehen ist, sind die Bestandteile eines Konfigurationspaketes, die eindeutige Bezeichnung des Root-Switches bzw. der Root-Bridge, des sendenden Switches und des sendenden Ports. Die Protokoll-ID wird meistens auf 0 gesetzt, was hier „IEEE 802.1d“-STP bedeutet[18][5]. Mit dem *Message Typ* wird gekennzeichnet, um was für ein Paket es sich handelt. Beim Wert 0 z.B. handelt es sich um eine Konfigurations-BPDU, die für die Bestimnung der Netz-Topologie zuständig ist. Für jedes Paket werden auch im Feld *Root Path Cost* die Wegkosten zwischen dem sendenden Port und dem Root-Switch berechnet. Mit diesen Informationen ist ein Switch in der Lage zu berechnen, ob der jeweilige Port *Root-Port* bzw. *Designated Port* werden soll.

[18] **Zitat** : Das IEEE hat mit dem ersten Standard 802.1d für STP 1998 die Grundlage dieser Protokollfamilie geschaffen. Im Jahr 2002 stellte IEEE 802.1w das RSTP und 802.1s das MSTP vor.

Bytes	Field
2	Protocol ID
1	Version
1	Message Type
1	Flags
8	Root ID
4	Cost of Path
8	Bridge ID
2	Port ID
2	Message Age
2	Max Age
2	Hellotime
2	Forward Delay

Abbildung 25: Aufbau eines BPDU-Konfigurations-Frames. Quelle:[1]

Wenn im Netz eine Topologie-Änderung stattfindet, wird das *Flags*-Feld (1 Byte) im Paket auf 00000001 gesetzt, um die anderen Mitglieder des Netzes auf die Änderung aufmerksam zu machen. Die weiteren vier Felder im Paket haben folgende Bedeutungen:

- **Message Age:** Das Alter des Paketes. Wenn ein Paket vom Root-Switch geschickt wird, enthält dieser Bereich den Wert 0. Dieser Wert wird in jedem Switch mit der *Max Age* des Pakets (siehe unten) verglichen und daraufhin wird entschieden, ob das Paket weitergeleitet bzw. gelöscht werden soll.
- **Max Age:** Dieses Feld wird vom Root-Switch gesetzt und hat einen konstanten Wert, welches zur Bestimmung des Paket-Alters verwendet wird.
- **Hello Time:** Das Zeitintervall, nach dessen Ablauf der Root-Switch weitere BPDUs versendet. Meistens auf 2 Sekunden gesetzt.

Jetzt können wir das Ganze mit einem praktischen Beispiel veranschaulichen. Das folgende Netz besteht aus fünf Switches, die mit der gleichen Priorität versehen sind. Jeder Switch wurde hierzu mit seiner MAC-Adresse dargestellt. Anfangs senden alle Switches Konfigurations-BPDUs, weil sie glauben, dass sie als Root-Switch agieren. Am Ende dieses Prozesses wird derjenige Switch zur Root-Switch, der die kleinste MAC-Adresse hat, weil alle Switches in diesem Beispile die gleiche Priorität haben. In unserem Beispiel ist es Switch1 mit der MAC-Adresse 0000:0000:0001. Nach der Feststellung ist Switch1 der einzige, der Konfigurations-BPDUs zu den anderen aussenden darf.

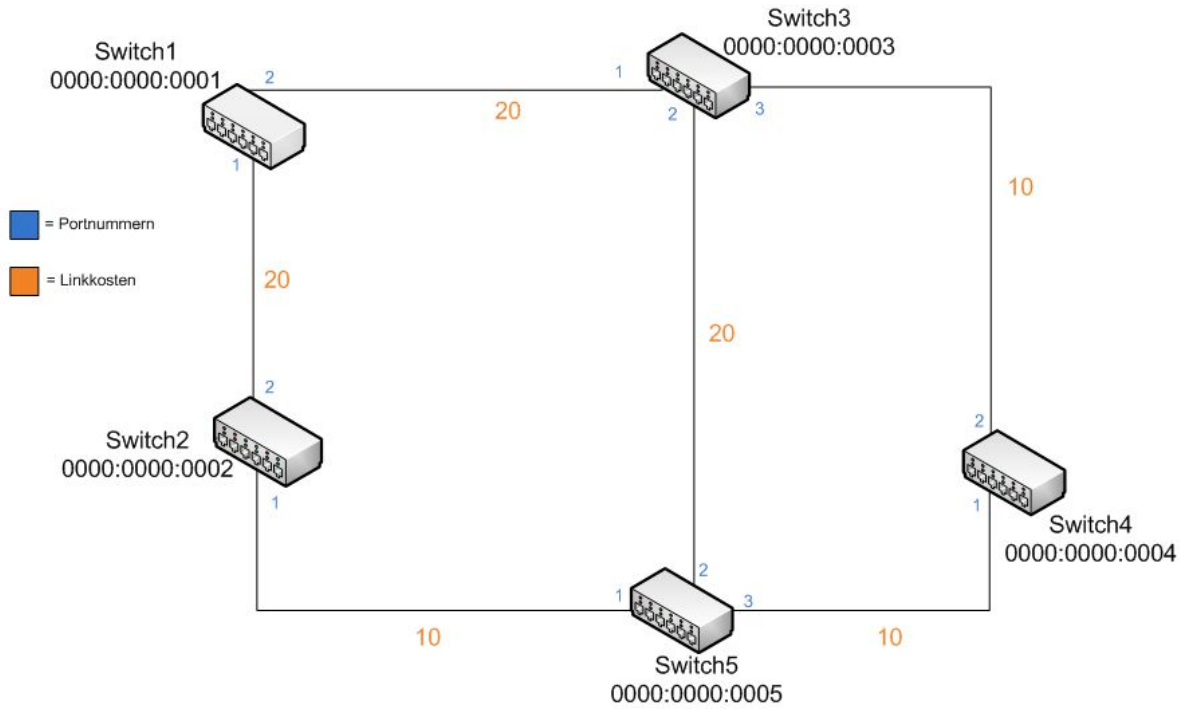

Abbildung 26: Netzwerk mit mehreren Loops

Auf diese Weise empfangen alle Non-Root-Switches BPDU-Pakete, die über unterschiedliche Verbindungen vom Root-Switch kommen. Weil jedes Paket die bislang hinterlegten Kosten beinhaltet, ist jeder Switch in der Lage, seinen sogenannten *Root-Port* festzustellen. Dieser Prozess findet in jedem Switch statt. Die ganze Kommunikation zwischen dem Root-Switch und dem Non-Root-Switch wird über diesen Port erledigt.
Für den Switch5 in unserem Netz gibt es drei Möglichkeiten den Root-Switch (Switch1) zu errreichen.

1. Über Switch2 mit den Kosten 30
2. Über Switch3 mit den Kosten 40
3. Über Switch4, dann über Switch3 mit den Kosten 40

Zur Berechnung der Kosten muss man darauf achten, dass nur die Kosten des sendenden Ports in Betracht gezogen werden.

Betrachtet man z.B. die Option über Switch3 mit den Kosten 40:

- Man verlässt den Switch5 über Port 2. Port-Kosten = 20,
  Pfad-Kosten = 0 + 20 = 20
- Man erreicht Switch3. Die Port-Kosten des empfangenden Ports(20) werden ignoriert,
  Pfad-Kosten = 20
- Man verlässt Switch3. Port-Kosten = 20,
  Pfad-Kosten = 20 + 20 = 40
- Man erreicht den Root-Switch, Port-Kosten des empfangenden Ports(20) werden ignoriert,
  Pfad-Kosten = 40

Switch5 erreicht den Root-Switch über Port 1 mit den geringsten Kosten (30). Port 2 wird zum Root-Port. Auf diese Weise wird für jeden Switch ein Root-Port festgestellt. Für jeden Link ist es notwendig, einen sogenannten *Designated-Switch* auszuwählen. Zuerst müssen die Switches für jeden Link entscheiden, welcher der zwei verbundendenen Switches die geringsten Kosten zum Root-Switch hat. Dieser wird dann für den anderen Switch zum *Designated-Switch*. Der Port, der den Designated Switch mit dem anderen Switch verbindet, wird als *Designated-Port* bezeichnet. In unserem Beispiel haben die Wege für Switch4 über Switch3 und für Switch5 über Switch2 die gleichen Kosten (30). Durch die Gleichheit der Priorität und die Gleichheit der Kosten, werden die MAC-Adressen zur Entscheidung eingesetzt. Da Switch4 eine kleinere MAC-Adresse besitzt, wird dieser zum Designated Switch für Switch5. Der Port 1 von Switch4 wird dadurch zum Designated Port. Als letztes gehen alle Non-Root- bzw. Non-Designated-Ports in den Blocking-Zustand über und damit hat unser Netz keine Schleifen mehr. Dadurch wird verhindert, dass Pakete mehrfach im Netz hin und her geschickt werden (siehe Abb. 27).

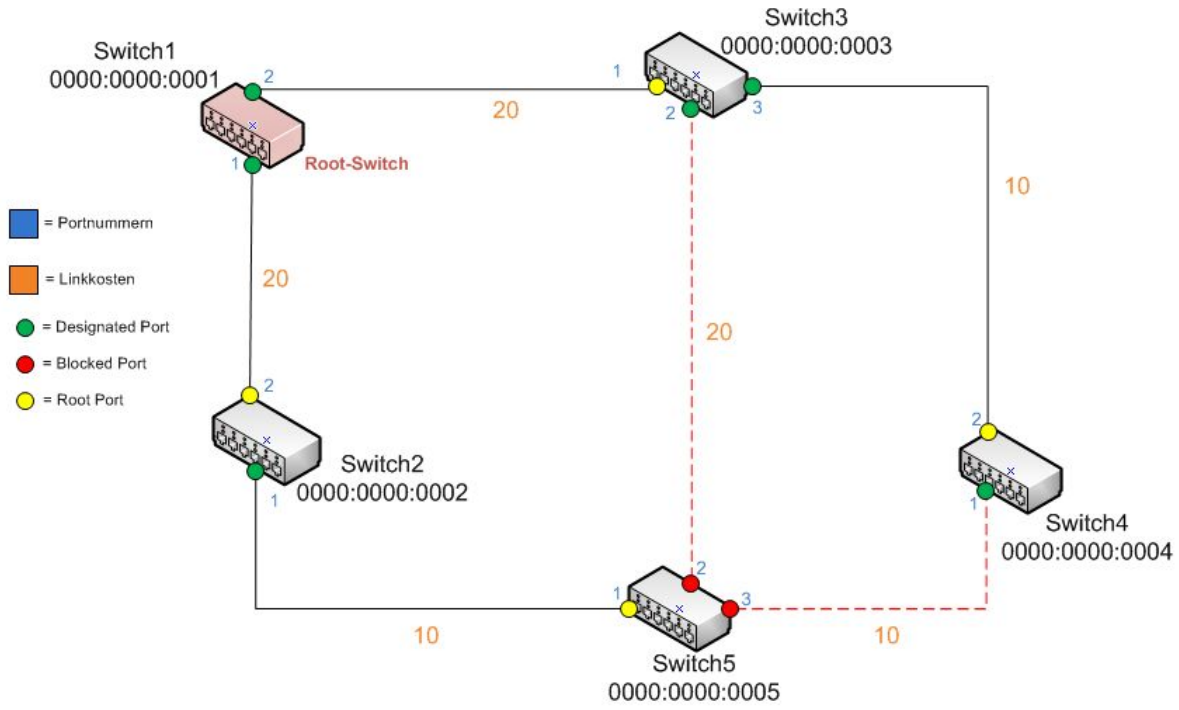

Abbildung 27: Ergebnis-Struktur des Netzes nach dem STP.

5.3.3 Konfigurierung des Spanning Tree Protokolls

Suat Algin

In jedem größeren Netzwerk mit mehreren miteinander verbundenen Switches ist das Spanning Tree Protokoll unerlässlich. Das STP wird pro VLAN konfiguriert und enthält folgende Konfigurationsmöglichkeiten:

- Ein-/Ausschalten des STP für ein oder mehrere Switches
- Konfigurierung der STP-Parameter je VLAN
- Änderung der STP Port-Parameter je VLAN
- Anzeige der STP-Informationen des STP Root-Switches

Änderung des STP-Status(CMS):

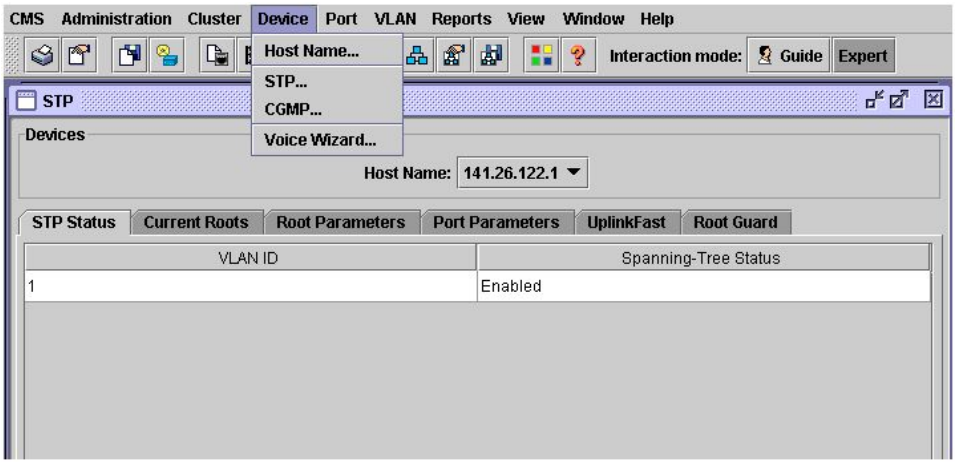

Abbildung 28: Konfigurierung des STP

Änderung des STP-Status(CLI):

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#spanning-tree vlan 1
141.26.122.1(config)#
```

Im selben Menüpunkt unter der Reiterkarte „Port Parameters“ werden Parameter wie „Port Fast“, „Path Cost“ und „Priority“ eingestellt. Der Parameter „Port Fast“ wird genutzt um die Statusänderung eines Ports zu beschleunigen. Hierbei wird ein Port im Falle einer Statusänderung vom „Blocking“-Zustand direkt in den „Forwarding“-Zustand gebracht, um die Zeit der Änderungen zu beschleunigen und so eine schnellere Ereichbarkeit der Ports zu gewährleisten. Die einzige Ausnahme stellt der „Restart“ eines Switches da. Bei einem „Restart“ durchläuft der Port alle Zustände des Spanning Tree Protokolls. Die Konfigurierung der „Port-Fast“ Einstellung sollte mit Bedacht eingesetzt werden, da es dazu führen kann, dass keine Loops im Netzwerk erkannt werden und dadurch das Netzwerk zusammenbricht.
Die Option „Path Cost“ beeinflusst den Aufbau des STP-Baumes. So kann der Administrator durch die Änderung dieser Kosten den STP-Baum nach seinen Vorstellungen aufbauen lassen. Falls ein Loop in einem Netzwerk ensteht, kann man so beeinflussen, welcher Port in den „Blocking“ und welcher Port in den „Forwarding“-Zustand übergeht. Umso niedriger diese Kosten eingestellt sind, desto höher wird die

Übertragungsgeschwindigkeit dieses Links angenommen. Der Parameter „Priority“ hat großen Einfluss auf das vom STP aufzubauenden Baum. So wird im Falle von zwei gleich schnellen Links mit gleichen Portkosten, der Link in den „Forwarding“-Zustand gesetzt, der die hörere Priorität hat. Dabei bekommt ein Port eine höhere Priorität, wenn sein Prioritätswert kleiner ist, als der Wert des gleichwertigen Links.

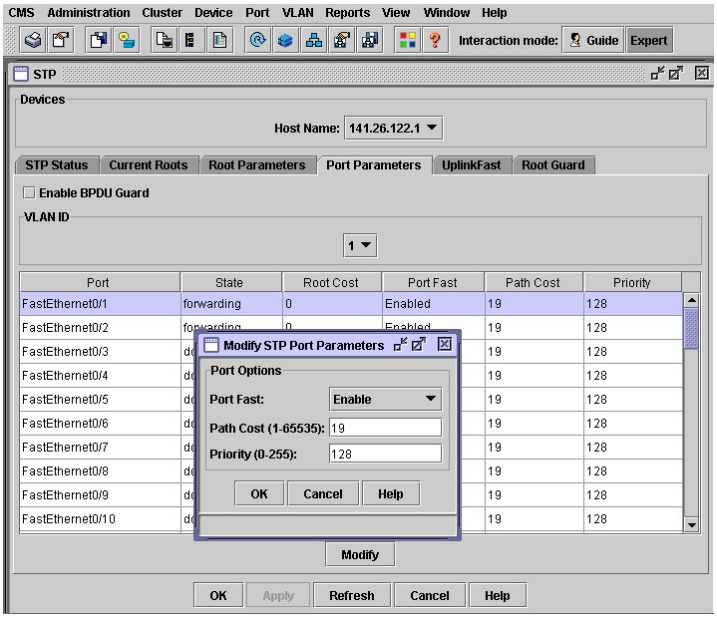

Abbildung 29: Port-Parameter des STP

5.3.4 Änderung der STP Parameter für VLANs

Suat Algin

Um die STP Parameter für ein VLAN zu ändern, öffnet man das in Abbildung 28 gezeigte Menü und wählt die Reiterkarte „Root Parameters“. Unter der Reiterkarte „Current Root“ sieht man die Einstellungen des, momentan als Root agierenden, Switches. Dabei wird jeweils der Switch mit der höchsten Priorität zum jeweiligen Root-Switch und sendet die sogenannten STP-Frames“ . Jedes VLAN besitzt dabei natürlich einen eigenen Root-Switch, um den Spanning Tree Protokoll-Baum aufzubauen. Die in diesem Fenster gezeigten Parameter werden nur im Read-Only Modus angezeigt und können an dieser Stelle nicht geändert werden. Im Gegensatz dazu kann man in dem Fenster „Root Parameters“ den zu konfigurierenden Switch auswählen und folgende Parameter je VLAN auf diesem einstellen:

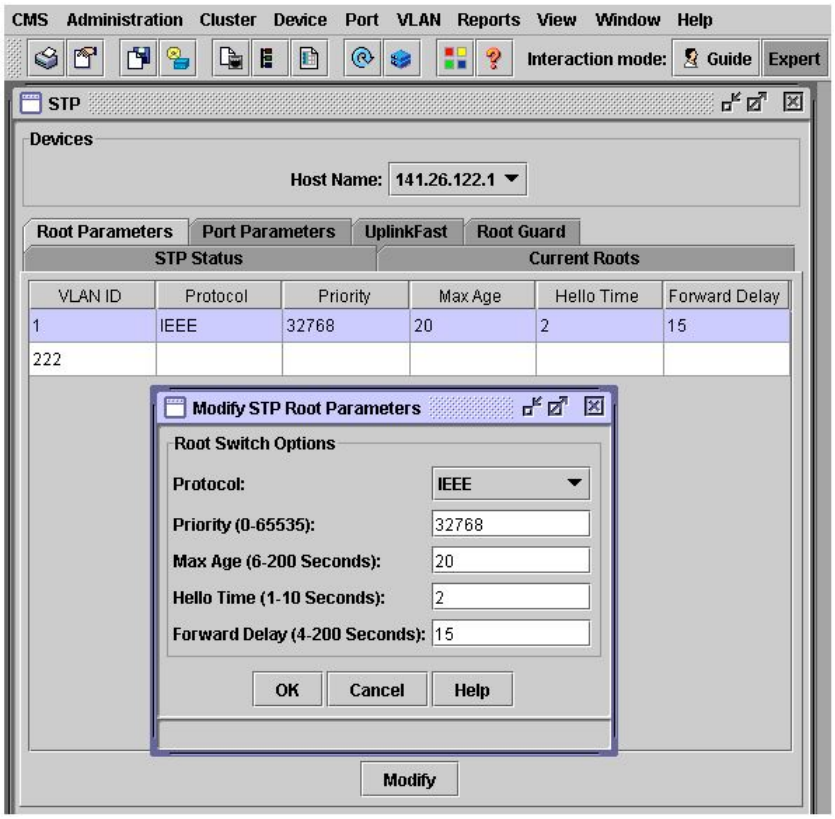

Abbildung 30: Konfigurierung der Root-Parameter des STP

Unter dem oberen „Dropdown-Menü“ wählt man zuerst den zu konfigurierenden Switch aus und sieht daraufhin in der ersten Spalte die auf dem Switch befindlichen VLANs. In der zu der VLAN-ID zugehörigen Zeile kann man nun die gewünschten Parameter setzen.

Diese Parameter sind folgendermaßen definiert:

- Protocol: Auswahl des STP-Protokolls(IEEE, IBM)
- Priority: Zur Auswahl des Root-Switches(niedriger Wert = hohe Priorität) Minimal 0 und maximal 65535
- Max Age: Die Zeit(in Sekunden) ohne Erhalt von STP-Konfigurationsmeldungen bevor ein Switch eine Rekonfiguration versucht. Dieser Parameter wird für den Root-Switch eingestellt und von allen anderen Geräten hieraus übernommen. Minimum 6 und maximal 200 Sekunden.
- Hello Time: Die Zeit, in der ein Switch Hello Pakete sendet um seinen Aktivitätsstatus zu veröffentlichen. Die Konfiguration des Root-Switches wird durch

alle anderen Geräten übernommem.
Minimum 1 und maximal 10 Sekunden.

- Forward Delay: Die Zeit(in Sekunden), die ein Switch wartet bevor er seinen Status vom „Learning“- oder „Listening“-Status in den „Forwarding“ ändert. Diese Zeit ist notwendig, damit die anderen Switches im Netzwerk sicher sein können, dass mit dem einschalten dieses Ports kein Loop entsteht.
  Minimum 4 und maximum 200 Sekunden.

Konfigurierung über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
141.26.122.1(config)#
```

Änderung des Spanning Tree Protokolls:

```
141.26.122.1(config)#spanning-tree vlan 1 protocol ieee
141.26.122.1(config)#end
141.26.122.1#show spanning-tree

Spanning tree 1 is executing the IEEE compatible STP
  Bridge Identifier has priority 32768, address 0005.dd3e.c740
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set, changes 0
  Times:  hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0

Interface Fa0/1 (port 13) in Spanning tree 1 is FORWARDING
   Port path cost 19, Port priority 128
   Designated root has priority 32768, address 0005.dd3e.c740
   Designated bridge has priority 32768, address 0005.dd3e.c740
   Designated port is 13, path cost 0
   Timers: message age 0, forward delay 0, hold 0
   BPDU: sent 73, received 0
   The port is in the portfast mode
.
.
.
141.26.122.1#
```

Änderung der Switch Priorität, des BPDU-Nachrichteninvalls und
des Hello-Nachrichtenintervalls:

```
141.26.122.1(config)#spanning-tree vlan 1 priority 15000
141.26.122.1(config)#spanning-tree vlan 1 max-age 60
141.26.122.1(config)#spaning-tree vlan 1 hello-time 5
141.26.122.1(config)#end
141.26.122.1#show spanning-tree

Spanning tree 1 is executing the IEEE compatible STP
  Bridge Identifier has priority 15000, address 0005.dd3e.c740
  Configured hello time 5, max age 60, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set, changes 0
  Times:  hold 1, topology change 75, notification 5
          hello 5, max age 60, forward delay 15
  Timers: hello 4, topology change 0, notification 0
.
.
.
```

Änderung des Forward Delay:

```
141.26.122.1(config)#spanning-tree vlan 1 forward-time 20
141.26.122.1(config)#end
141.26.122.1#show spanning-tree
Spanning tree 1 is executing the IEEE compatible STP
  Bridge Identifier has priority 15000, address 0005.dd3e.c740
  Configured hello time 5, max age 60, forward delay 20
  We are the root of the spanning tree
.
.
.
```

Konfigurierung der Port-Fast Option und der Pfad-Kosten:

```
141.26.122.1(config)#interface fa0/1
141.26.122.1(config-if)#spanning-tree portfast
%Warning: portfast enabled on FastEthernet0/1.
 Usually portfast should be enabled on ports connected to
 a single host.
 When portfast is enabled, connecting hubs, concentrators,
 switches, bridges,
 etc. to this interface may cause temporary spanning tree loops.
 Use with CAUTION.
141.26.122.1(config-if)#spanning-tree vlan 1 cost 300
141.26.122.1(config-if)#end
141.26.122.1#
```

Einstellung der Port-Priorität:

```
141.26.122.1(config)#interface fa0/1
141.26.122.1(config-if)#spanning-tree vlan 1 port-priority 60
141.26.122.1(config-if)#end
141.26.122.1#
```

Je nach Netzwerktopologie kann es notwendig sein, dass man verhindert das Switches zu Root-Switches werden. Dies ist zum Beispiel der Fall, wenn ein Service-Provider mit Switches verbunden ist, die nicht zu seinem Eigentum gehören. Um zu verhindern, dass einer dieser Switches zum Root-Switch wird, kann man die Option „Root-Guard“ einsetzen. Die Option wird auf den Ports gesetzt, welche mit den außenstehenden Switches verbunden sind. Dabei wird der Port geblockt, sobald einer der verbundenen Switches zum „Root-Switch“ wird. Danach wird ein neuer Root-Switch bestimmt. Die Option gilt hierbei für alle VLANs zu denen dieser Port gehört.
Konfigurierung über das CLI:

```
141.26.122.1(config)#interface fa0/1
141.26.122.1(config-if)#spanning-tree rootguard
141.26.122.1(config-if)#end
141.26.122.1#show running-config
Building configuration...

Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 141.26.122.1
!
enable secret 5 $1$wNMEH2ERoi1Ha9.KSwVvAi2vk0
!
!
!
!
!
!
ip subnet-zero
ip name-server 141.26.1.1
!
cluster commander-address 0006.5377.0f40 member 3 name Cluster01
!
!
interface FastEthernet0/1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport trunk pruning vlan none
 spanning-tree portfast
 spanning-tree vlan 1 cost 300
 spanning-tree vlan 1 port-priority 60
 spanning-tree rootguard
!
.
```

5.4 EtherChannel-Port-Gruppen Erstellung

Suat Algin

Die Option der EtherChannel-Port Erstellung ist ein von Cisco entwickeltes Verfahren, welches zur Bündelung von beliebigen Ethernet-Schnittstellen zu einem logischen Kanal genutzt wird. Durch die Bündelung von „Fast Ethernet- oder Gigabit Ethernet-Ports“ wird sowohl der Datendurchsatz erhöht, wie auch für eine höhere Ausfallsicherheit gesorgt. Die Erstellung von “Fast EtherChannel- und Gigabit EtherChannel-Port- Gruppen“ erfolgt über den Menüpunkt “Port“ / “EtherChannels...“. Doch bevor wir zur Erstellung übergehen ein paar Punkte zum Verständnis der Gruppierung. Wie soeben erwähnt, versucht die „EtherChannel-Port“ Gruppierung den Datendurchsatz, durch Verteilung des Traffics auf mehrere Leitungen, zu optimieren. Hierbei werden die „Fast EtherChannel-Ports“, beim Einsatz des Spanning Tree Protokolls, als ein logischer Port angesehen.
Es existieren zwei Möglichkeiten der Port-Gruppierung:

Source-based forwarding orientiert sich, wie der Name schon besagt, an der Quelladresse der Pakete. Dabei werden die Pakete dieser Quelladresse entsprechend an die zugehörige Gruppe weitergeleitet. Diese Option ist standardmäßig eingestellt. Die maximale Anzahl an konfigurierbaren Source-based forwarding Ports pro Gruppe beträgt 8.

Destination-based forwarding leitet im Gegensatz zum Source-based forwarding, die Pakete anhand Ihrer Zieladresse an die entsprechende Gruppe. Die Anzahl der Destination-based forwarding Ports pro Gruppe ist nicht limitiert.

Pro Switch können maximal 12 Port Gruppen mit der Source-based oder Destination-based Methode erstellt werden. Das heißt aber nicht, dass man zwangläufig alle 12 Gruppen mit einer der zwei Methoden erstellen muss. Die einzige Voraussetzung ist, dass alle Ports innerhalb einer Gruppe zu genau einer der Methoden gehören. Auch bei Gruppierungen von Verlinkungen über mehrere Switches, sollte drauf geachtet werden, dass die Schnittstellen konsistent konfiguriert werden.

Beispiel zur Verwendung von EtherChannel-Ports:

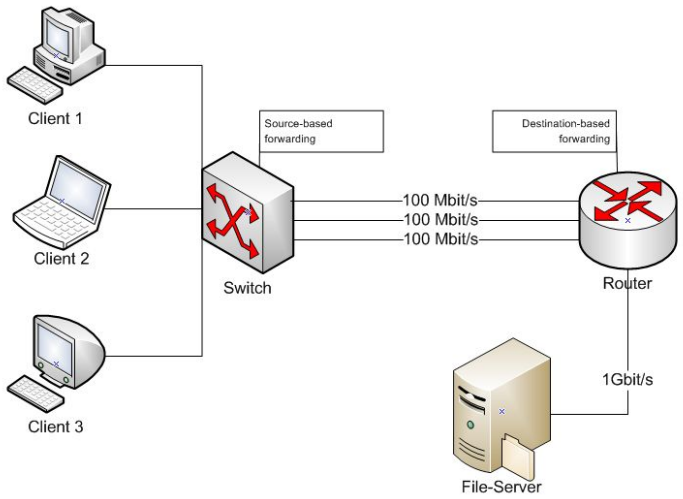

Abbildung 31: Fast EtherChannel-Port Gruppierung

Wir nehmen an, dass wir 3 Client-Zugriffe(je 100Mbit/s) auf einen, hinter einem 1Gbit-Router sitzenden, File-Server mit Hilfe von EtherChannels optimieren möchten. In diesem Fall werden wir mehrere Ports zwischen dem Switch und dem Router zu einer Fast-Ethernet-Port Gruppe zusammenfassen, um den Datendurchsatz zu vervielfachen. Da der Router nur eine MAC-Adresse für alle drei Leitungen besitzt, programmieren wir die Source-based forwarding Methode auf dem Switch. So ist gewährleistet, dass alle Anfragen die an den, hinter dem Router sitzenden, File-Server gehen, auf alle drei Leitungen der Port Gruppe verteilt werden. So können im Optimalfall drei Clients mit Fast-Ethernet Verbindungen,mit ihren vollen 100Mbit/s je über einen Port der Port-Gruppe auf den File-Server zugreifen. Auf der anderen Seite programmieren wir auf dem Router die Destination-based Methode, damit die Pakettransporte zu den einzelnen Clients auf die Port-Gruppe verteilt wird. So wird für den MAC jedes Clients eine 100Mbit/s Leitung belegt, um den Traffic auf alle drei Leitungen zu verteilen[19]. Die Konfigurierung des Catalyst 3548XL kann sowohl über das CLI, wie auch über die CMS erfolgen.

[19]Diese Verteilung wird auch Load-Balancing genannt

Konfigurierung beider Möglichkeiten:

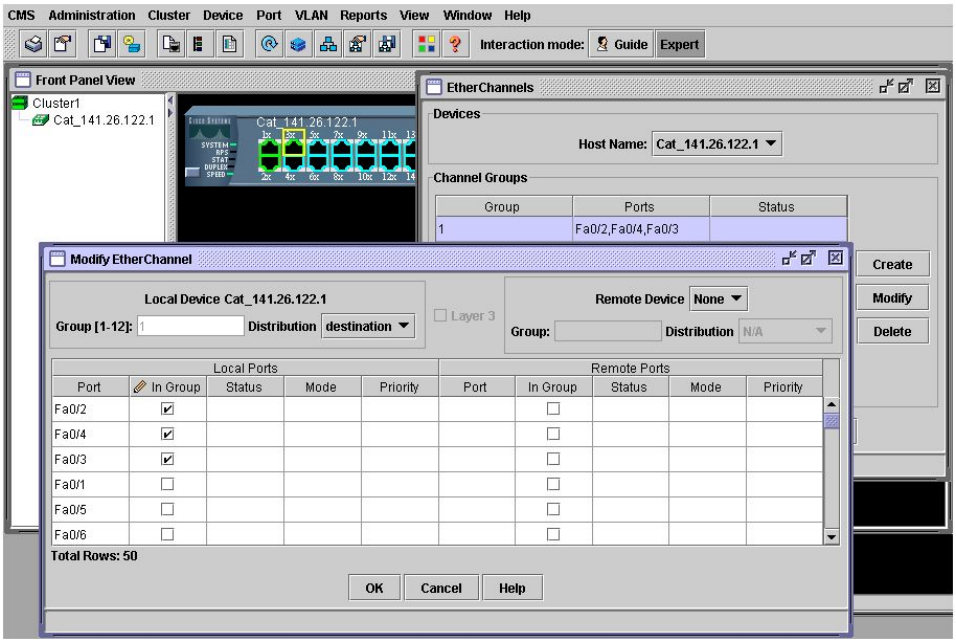

Abbildung 32: Konfigurierung über die CMS

Konfigurierung über das CLI:

```
User Access Verification

Password:
Cat_141.26.122.1>enable
Cat_141.26.122.1#config terminal
Enter configuration commands, one per lin. End with CNTL/Z.
Cat_141.26.122.1(config)#interface fa0/2
Cat_141.26.122.1(config-if)#port group 1 distribution destination
Cat_141.26.122.1(config-if)#interface fa0/3
Cat_141.26.122.1(config-if)#port group 1 distribution destination
Cat_141.26.122.1(config-if)#interface fa0/4
Cat_141.26.122.1(config-if)#port group 1 distribution destination
Cat_141.26.122.1(config-if)#exit
Cat_141.26.122.1(config)#exit
Cat_141.26.122.1#
```

5.5 Switch Port Analyzer(SPAN)

Suat Algin

Ein für Administratoren sehr nützliches Werkzeug wird bei Cisco mit SPAN abgekürzt. Der Switch Port Analyzer wird dazu genutzt, um eingehende und ausgehende Traffic auf bestimmten Ports von einem Port aus zu beobachten. Dabei werden alle Informationen des Traffics, welche über die vorher bestimmten Ports laufen, an das beobachtende Port weitergeleitet.
Um als SPAN Port agieren zu können, muss ein Port folgende Voraussetzungen erfüllen:

- Der beobachtende und der beobachtete Port müssen sich im selben VLAN befinden
- Ein SPAN Port muss mit der Zugriffsmethode *static-access* versehen werden
- Beobachtende Ports dürfen sich in keinem EtherChannel befinden

Konfigurationsmöglichkeiten:

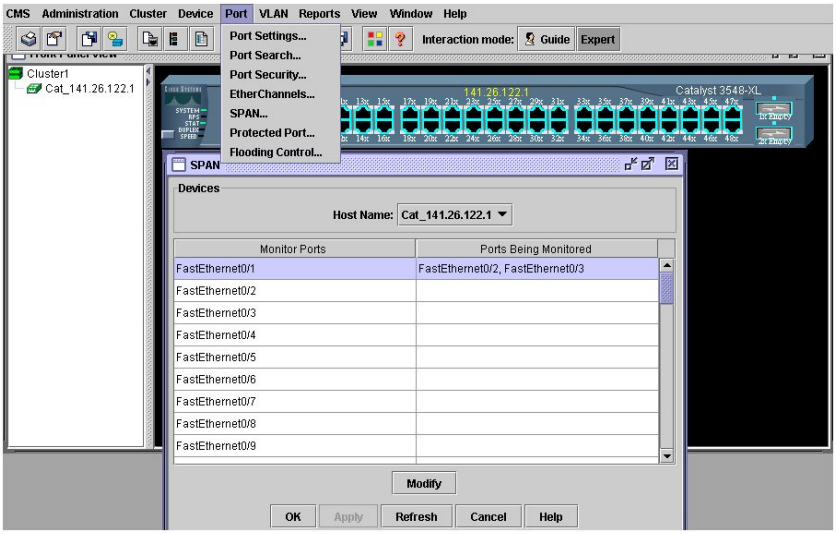

Abbildung 33: Konfigurierung über die CMS

Konfigurierung über das CLI:

```
User Access Verification

Password:
Cat_141.26.122.1>enable
```

```
Password:
Cat_141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Cat_141.26.122.1(config)#interface fa0/1
Cat_141.26.122.1(config-if)#port monitor fa0/2
Cat_141.26.122.1(config-if)#port monitor fa0/3
Cat_141.26.122.1(config-if)#end
Cat_141.26.122.1#show running-config
Building configuration...

Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Cat_141.26.122.1
!
enable secret 5 $1$WYrq$lja0u1CTcdzyWE9k3Z0zx1
!
!
!
!
!
!
ip subnet-zero
ip name-server 141.26.1.1
!
!
interface FastEthernet0/1
 port monitor FastEthernet0/2
 port monitor FastEthernet0/3
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport trunk pruning vlan none
 spanning-tree portfast
.....................................
```

Deaktivierung des SPAN über das CLI:

```
Cat_141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Cat_141.26.122.1(config)#interface fa0/1
Cat_141.26.122.1(config-if)#no port monitor
Cat_141.26.122.1(config-if)#end
Cat_141.26.122.1#show running-config
Building configuration...
hostname Cat_141.26.122.1
```

```
!
enable secret 5 $1$WYrq$lja0u1CTcdzyWE9k3Z0zx1
.
.
.
interface FastEthernet0/1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport trunk pruning vlan none
 spanning-tree portfast
```

5.6 Zuweisung eines Domain-Namens/eines Domain-Name Servers

Suat Algin

Jede mit einer IP-Adresse versehene Verbindung, sei es ein Server, Router, Computer oder ein anderer Switch, kann mit einem Hostnamen assoziiert werden. Um diesen zu speichern, hält die IOS des Catalyst 3548XL einen Cache bereit. In diesem Cache werden die einzelnen IP-Adressen mit den zusammengehörigen Hostnamen gespeichert. So kann man z.B. die Geschwindigkeiten bei Telnet-Verbindungen, durch die schnellere Übersetzung der Hostnamen in IP-Adressen, erhöhen. Ein Domain-Name setzt sich aus der Top-Level-Domain, dem Hostnamen(Second-Level-Domain) und den Subdomains(Third-Level-Domain) zusammen.

- Top-Level-Domain: .com, .de, .eu...
- Second-Level-Domain: .uni-koblenz,.google,.gmx...
- Third-Level-Domain: www, ftp, de...

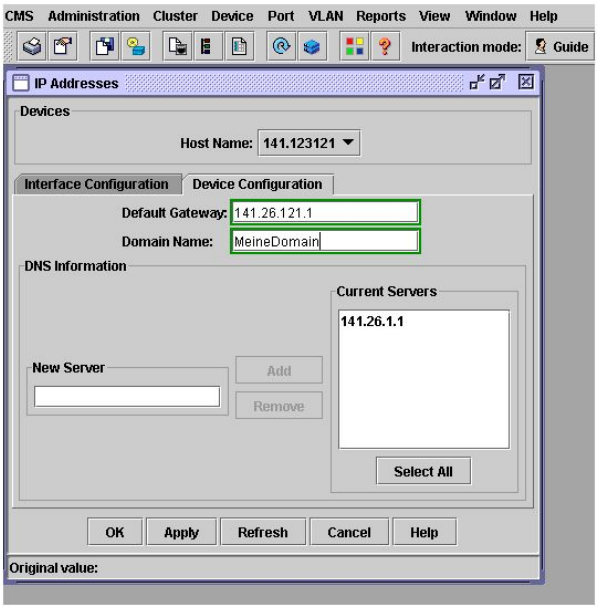

Abbildung 34: Konfigurierung eines Domain Name Servers

Konfigurierung über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
141.26.122.1(config)#hostname 141.26.122.1
141.26.122.1(config)#ip domain-name MeineDomain
141.26.122.1(config)#ip default-gateway 141.26.121.1
141.26.122.1(config)#ip dhcp-server 141.26.1.1
141.26.122.1(config)#no ip dhcp-server 141.26.1.1
141.26.122.1(config)#end
141.26.122.1#
```

5.7 Flooding-Control

Suat Algin

Paketstürme treten auf, wenn eine große Anzahl von Broadcast, Unicast oder Multicast Paketen auf einem Port eintreffen. Wenn diese Pakete an weitere Ports weitergeleitet werden, besteht die Gefahr einer Verlangsamung des Netzwerks oder sogar der komplette Ausfall. So kann solch ein Broadcast-Storm auch Router beeinflussen und so zu

Störungen über VLAN Grenzen hinaus führen. Um solch einem Problem vorzubeugen, existieren drei verschiedene Techniken. Bei diesen Techniken wir die Weiterleitung von unnötigen, viel „Traffic“ verursachenden, Daten blockiert.

- Allgemeine Sturmkontrolle der Unicast-, Multicast- oder Broadcast-Pakete
- Blockung der Weiterleitung der Unicast- oder Multicast-Pakete von unbekannten MAC-Adressen für einen bestimmten Port
- Weiterleitung aller unbekannten Pakete an einen Netzwerkport

Die allgemeine Sturmkontrolle kann für den gesamten Switch oder für einen einzelnen Port konfiguriert werden. Standardmäßig ist die Sturmkontrolle ausgeschaltet. Um auftretende Paketstürme zu erkennen und die Weiterleitung zu blockieren bzw. wieder zu ermöglichen, existieren zwei Variablen. In diesen Variablen werden zwei Schwellen festgelegt. In der niedrigeren Schwelle der beiden wird die Anzahl der einkommenden Pakete, mit der ein Switch normal arbeitet festgelegt. Dabei gilt, umso höher diese Schwelle desto niedriger die Qualität der Broadcast-Sturm erkennung. Die maximale Einstellung liegt bei 4.294.967.295 Paketen pro Sekunde. Die höhere Schwelle der beiden Variablen legt die maximale Anzahl an einkommenden Paketen, bevor die Weiterleitung blockiert wird, fest.

Konfigurierung der allgemeinen Blockung über die CMS:

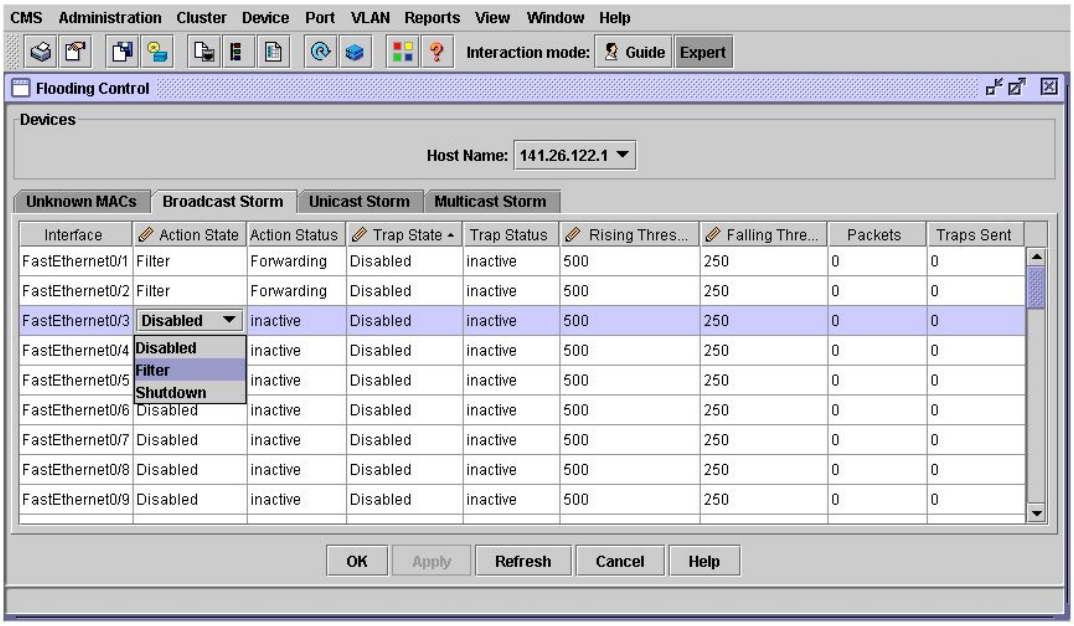

Abbildung 35: Flooding Control

Konfigurierung über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#interface fa0/1
...(config-if)#$control broadcast threshold rising 1000 falling 500
141.26.122.1(config-if)#port storm broadcast trap
141.26.122.1(config)#interface fa0/2
...(config-if)#$control broadcast threshold rising 1000 falling 500
141.26.122.1(config-if)#port storm broadcast trap
141.26.122.1(config-if)#end
141.26.122.1#show port storm-control broadcast
Interface Filter St.  Trap State   Rising Falling Current Traps Snd
--------- ----------- ------------ ------ ------- ------- ---------
Fa0/1     Forwarding  Below rising   1000     500       0         0
Fa0/2     Forwarding  <inactive>     1000     500       0         0
.
.
```

Blockung der Unicast-, Multicast-Pakete über unbekannte MAC-Adressen:

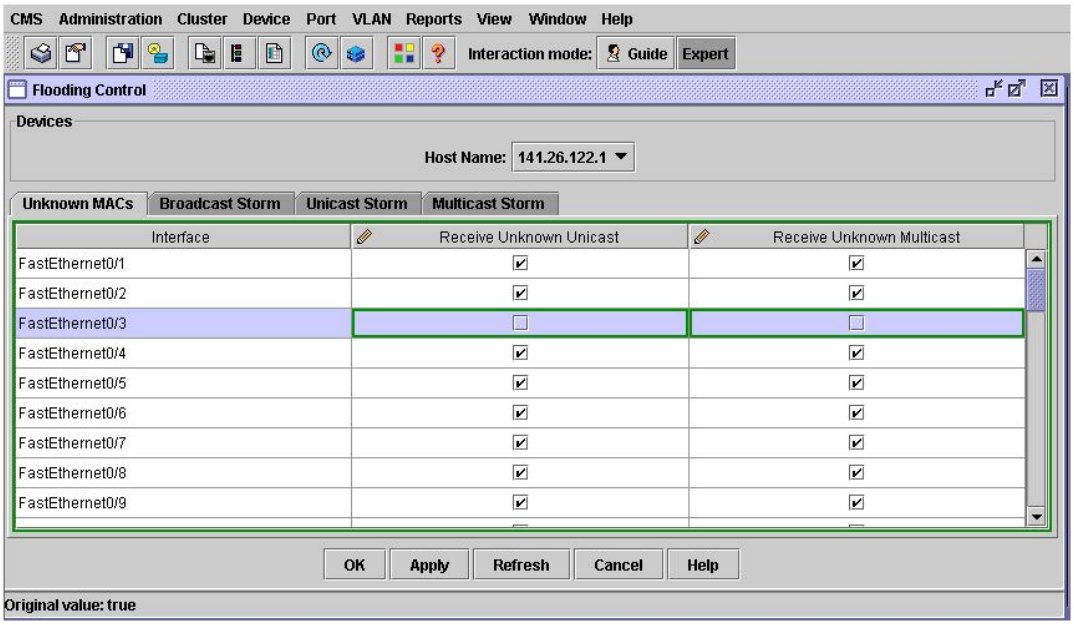

Abbildung 36: Blockung der Unicast-/Multicast-Pakete

Konfigurierng über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#interface fa0/3
141.26.122.1(config-if)#port block multicast
141.26.122.1(config-if)#port block unicast
141.26.122.1(config-if)#end
141.26.122.1#show port block multicast
VLAN1 is blocked from unknown multicast addresses
FastEthernet0/1 is receiving unknown multicast addresses
FastEthernet0/2 is receiving unknown multicast addresses
FastEthernet0/3 is blocked from unknown multicast addresses
.
.
141.26.122.1#
141.26.122.1#show port block unicast
VLAN1 is blocked from unknown unicast addresses
```

```
FastEthernet0/1 is receiving unknown unicast addresses
FastEthernet0/2 is receiving unknown unicast addresses
FastEthernet0/3 is blocked from unknown unicast addresses
.
.
```

Netzwerkports zur Paket Weiterleitung

Ein Netzwerkport wird zur Reduzierung von Paketflutungen genutzt. Falls ein Switch ein Paket mit unbekannter Empfänger-Adresse erhält, leitet er dieses Paket an alle Ports weiter. Um dieses Vorgehen zu verhindern ist man in der Lage einen so genannten Netzwerkport zu definieren. Es kann pro VLAN nur ein Netzwerkport definiert werden. Alle Pakete mit unbekanntem Empfänger werden vom Switch an diesen Port weitergeleitet um unnötigen Traffic an den Ports zu unterbinden. Gleichzeitig löscht der Switch alle in Verbindung mit diesem Port gelernten Adressen aus der Adressentabelle und nimmt auch keine neuen Adressen mehr von diesem Port in seine Tabelle auf. Die einzige Ausnahme stellt ein als „Secure Port“ konfigurierter Port da. Von solch einem Port gelernte Adressen werden nicht aus der Adressentabelle entfernt.
Ein Netzwerkport besitzt folgende Eigenschaften:

- Blockierung von Unicast-, Multicast-Paketen nicht möglich
- Ein Netzwerkport pro VLAN
- Darf nicht als Verbindung zwischen zwei Switches im Cluster agieren
- Nur Quelladress-basierte Port-Gruppierung möglich
- Darf kein SPAN-Port sein

Die Konfigurierung eines Netzwerk-Ports ist nur über das CLI möglich:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#interface fa0/4
141.26.122.1(config-if)#port network
141.26.122.1(config-if)#end
141.26.122.1#show running-config
Building configuration...

Current configuration:
```

```
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 141.26.122.1
.
.
.
interface FastEthernet0/1
 port storm-control broadcast action filter
 port storm-control broadcast trap
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport trunk pruning vlan none
 spanning-tree portfast
!
interface FastEthernet0/2
 port storm-control broadcast action filter
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport trunk pruning vlan none
 spanning-tree portfast
!
interface FastEthernet0/3
 port block unicast
 port block multicast
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport trunk pruning vlan none
 spanning-tree portfast
!
interface FastEthernet0/4
 port network
!
interface FastEthernet0/5
141.26.122.1#
```

5.8 Konfigurierung der IP-Adresse

Suat Algin

Dieser Abschnitt behandelt die Zuweisung einer IP-Adresse. Vorab ist zu erwähnen, dass ein Switch nicht unbedingt eine IP-Adresse zur Administration benötigt. Falls ein Switch Mitglied eines Clusters ist, kann dieser auch über den Command-Switch angesprochen und konfiguriert werden. Falls nötig, erfolgt die erstmalige Zuweisung einer IP-Adresse über den Konsolenport. Hierzu wird eine Verbindung über einen seriellen Port des PCs und der Konsolenschnittstelle des Switches aufgebaut. Falls das zu verbindende Gerät keinen seriellen Port besitzt, existiert auch die Möglichkeit die Verbindung über einen USB-Serial-Adapter herzustellen. Diese erstmalige Verbindung kann zum Beispiel über ein Programm wie „Putty“ erfolgen:

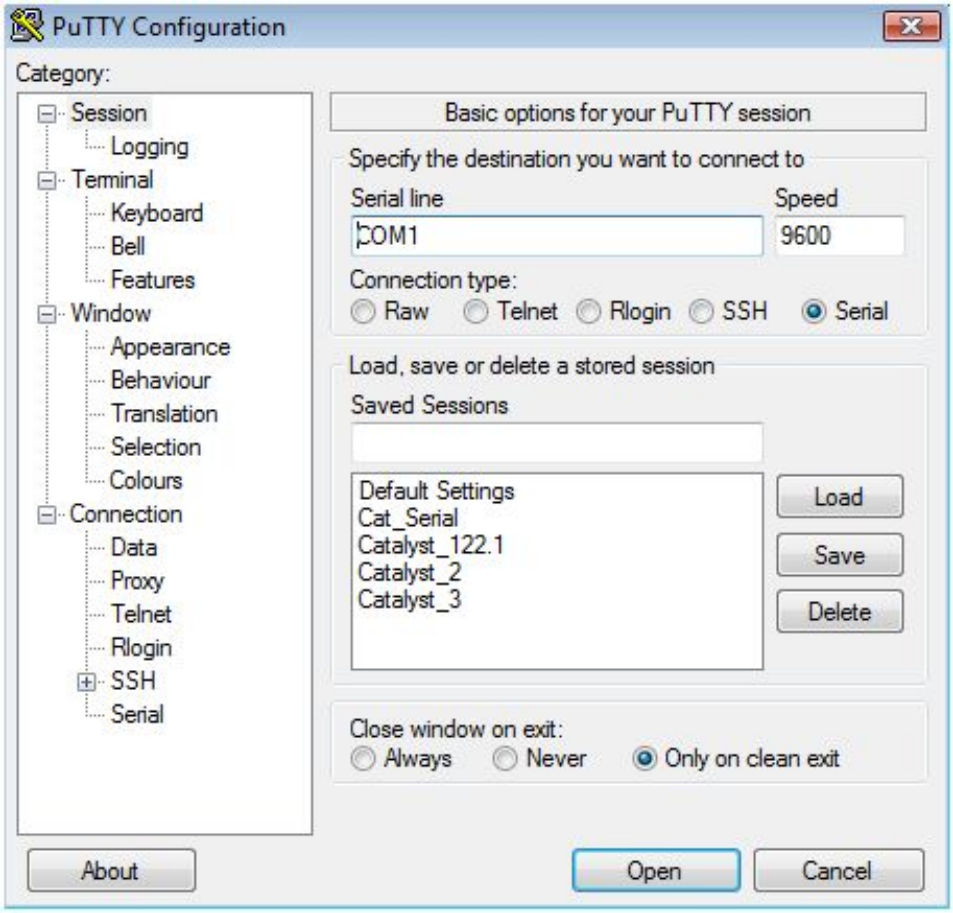

Abbildung 37: Putty: Verbindungsherstellung über den seriellen Port

Nach der erstmaligen Verbindung und der Vergabe einer IP-Adresse kann man den Switch über einen Browser und die vergebene IP-Adresse ansprechen.

Konfigurierung über die CMS:

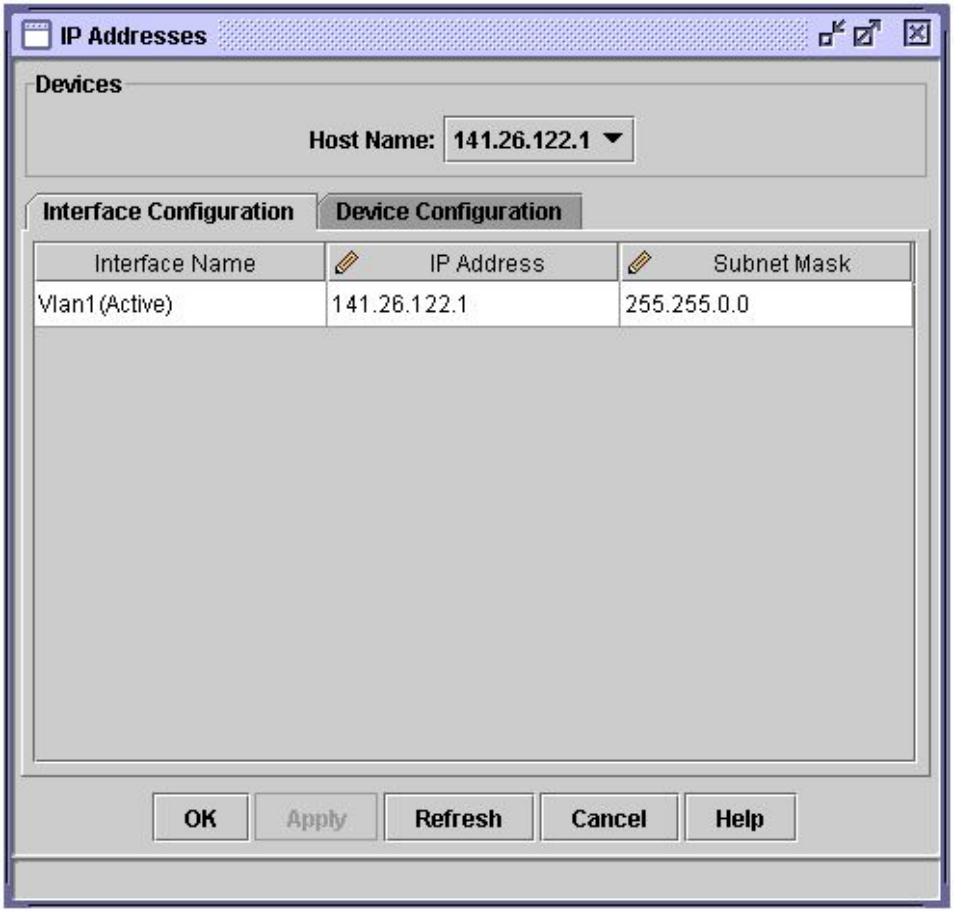

Abbildung 38: Einstellung der IP-Adresse

Konfigurierung über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#interface vlan 1
141.26.122.1(config-if)#ip address 141.26.122.1 255.255.0.0
141.26.122.1(config-if)#end
141.26.122.1#show running-config
Building configuration...

Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
.
.
 ip address 141.26.122.1 255.255.0.0
 no ip directed-broadcast
 ip nat outside
 no ip route-cache
!
141.26.122.1#
```

5.9 Das Simple Network Protocol(SNMP)

Suat Algin

Dieses Netzwerkprotokoll, welches 1988 definiert wurde, wird zur Steuerung von Netzwerkelementen über eine zentrale Station genutzt. Hierbei regelt das Protokoll die Kommunikation der Elemente mit der zentralen Station. Es existieren drei verschiedene Einsatzmöglichkeiten:

- Das „Monitoring“: Sammeln der Informationen über Netzwerkstatisken oder über Ereignisse
- Das „Controlling“: Fernsteuerung der Netzwerkelemente
- Die „Administration“: Sammeln der Informationen über die Entwicklung des Netzaufbaus

Das Management per SNMP funktioniert nach dem Client-Server-Prinzip. Dabei überwachen ein oder mehrere Netzwerk Management Stationen(NMS), die ihnen zugewiesenen Netzwerk Management Elemente(NME). Auf der NMS sammelt eine Management-Software Daten über den Zustand der Netzwerk Management Elemente. Diese Daten werden über bestimmte Agenten, die auf den NME laufen und auf den UDP-Port 161 reagieren, geliefert. Das SNMP Kommunikationsprotokoll strukturiert hierbei die Daten in einem Management Information Base(MIB)-Baum.

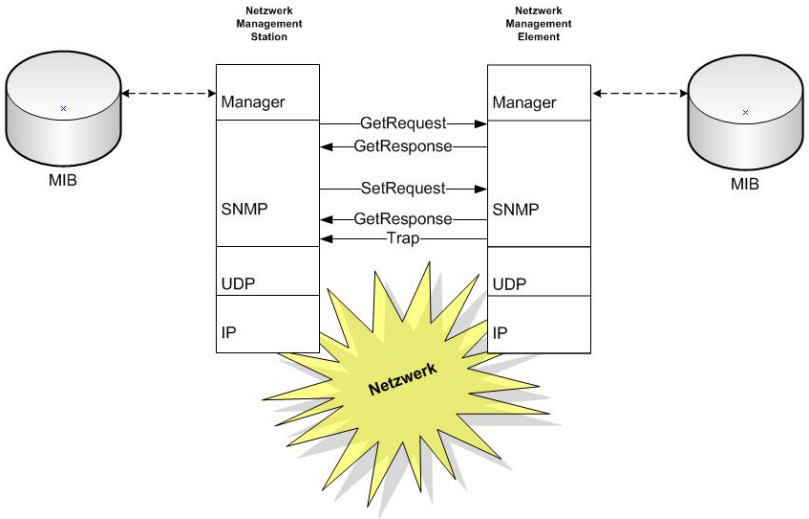

Abbildung 39: SNMP-Struktur der Management Informationen(SMI)

Wie in Abbildung 39 zu sehen ist, sendet die Management Station ein GetRequest-Paket an den Agenten, um den momentanen Zustand abzufragen. Dieses GetRequest-Paket wird Protocol Data Unit(PDU) genannt. Diese GetRequest-Paket Anfrage wird vom Agenten mit einem GetResponse-Paket beantwortet. Daraufhin kann die NMS ein GetNextRequest-Paket versenden, welches wiederum mit einem GetResponse-Paket erwidert wird. Die NMS kann aber auch ein SetRequest-Paket versenden, um die Konfiguration eines NME zu modifizieren. Daraufhin antwortet das NME auch wieder mit einem GetResponse-Paket, in dem die Werte nach der Modifikation enthalten sind. Ein sogenanntes Trap-Paket, welches vom NME Agenten an die NMS gesendet wird, enthält Informationen über ein erkanntes Problem. Dieses Trap-Paket wird unaufgefordert über den UDP-Port 162 an die NMS gesendet, um diesen auf dieses Problem aufmerksam zu machen. So ist der Administrator über plötzlich auftretende Probleme informiert.
Diese Informationen können zum Beispiel Statusänderungen eines Ports oder Authentifizierungsprobleme enthalten. Ein großes Problem des SNMPv1 ist die mangelhafte Authentifizierung. Die Authetifizierung efolgt beim SNMP über ein Community-String. Der Community-String teilt dem Agenten mit, dass diese Station berechtigt ist einen Auftrag zu erteilen. Dabei existiert ein Community-String für Get-Abfragen und ein Community-String für Set-Abfragen. Wenn eine NMS diese Zeichenkette kennt, ist es in der Lage Daten aus einem Agenten auszulesen oder diesem Befehle zu erteilen. Das Haupt-Sicherheitsproblem der Version SNMPv1 liegt in der Klarext-Übertragung des Passwortes. Durch diese unsichere Übertragung kann das Passwort von unberechtigten Stationen abgehört und genutzt werden[20].
Deshalb wird auch vor dem Einsatz des SNMP in nicht öffentlichen Netzwerken gewarnt. Trotz dieser Sicherheitsprobleme wird die SNMP Version 1 noch weitgehend verwendet. Um diese Sicherheitsprobleme zu beheben, wurde 1993 die SNMP Version 2 entwickelt. Diese setzt sich aus dem nie eingeführten Secure SNMP und der SNMP Version 1 zusammen. Aber diese Version verbreitete sich wegen weiteren Sicherheitsproblemen nicht besonders und wird auch nicht mehr genutzt.
Im Jahr 2002 wurde die SNMP Version 3 veröffentlicht und mit den RFCs 3410 bis 3418 definiert. Durch diese Version wird eine Authentifizierung und Verschlüsselung auf Nachrichtenebene ermöglicht. Dabei wird die Authentifikation anhand von „Hashed Message Authentication Codes“ durchgeführt.HMAC ist eine Methode, welches zur Authentifizierung durch kryptografische Hash-Funktionen genutzt wird[21].
Aber nach neuesten wurde auch hier eine Schwachstelle in den Sicherheitsmechanismen entdeckt. Durch eine Sicherheitslücke bei einigen SNMP Implementationen, kann eine unberechtigte NMS die Länge der HMAC-Verschlüsselung selbst festlegen und so die Anzahl der möglichen Schlüssel dieser Verschlüsselung auf ein minimum reduzieren. So existieren zum Beispiel bei einer Verschlüsselung von 1 Byte nur 256 Schlüssel[22].

[20]Definition: RFC 1157
[21]RFC 2104
[22]1 Byte = 8 Bit = 2*2*2*2*2*2*2*2 = 256 Schlüssel

Falls SNMP auf dem Catalyst deaktiviert ist, kann man keine Graphen und Berichte in der Cluster Management Suite anzeigen lassen. Diese werden in der CMS des Cisco Catalyst 3500XL über SNMP-Nachrichten abgerufen.
In dem Menü aus Abbildung 40 kann der Administrator folgende Aufgaben erledigen:

- SNMP Ein-/Ausschalten
- Anzeige genereller Informationen
- Eingabe eines Community-Strings, das als Passwort für SNMP-Nachrichten dient
- Konfigurierung des Trap Managers, welcher die Switchaktivitäten protokoliert und meldet
- Einstellung der abzufangenden Switchaktivitäten

Konfigurierung des SNMP:

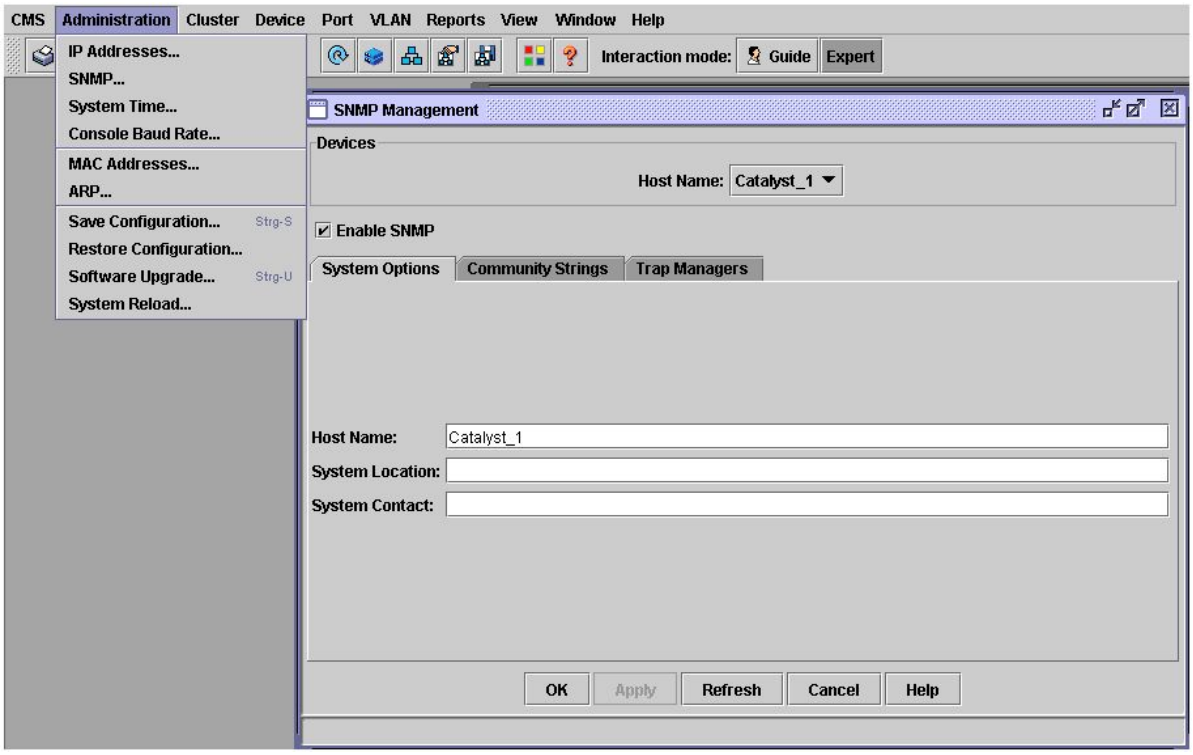

Abbildung 40: SNMP-Einstellungen(CMS)

Konfigurierung über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Catalyst_1(config)#snmp ?
  chassis-id        String to uniquely identify this chassis
  community         Enable SNMP; set community string and access privs
  contact           Text for mib object sysContact
  enable            Enable SNMP Traps or Informs
  engineID          Configure a local or remote SNMPv3 engineID
  group             Define a User Security Model group
  host              Specify hosts to receive SNMP notificationbs
  location          Text for mib object sysLocation
  packetsize        Largest SNMP packet size
  queue-length      Message queue length for each TRAP host
  system-shutdown   Enable use of the SNMP reload command
  tftp-server-list  Limit TFTP servers used via SNMP
  trap-source       Assign interface for the source address of all traps
  trap-timeout      Set timeout for TRAP message retransmissions
  view              Define an SNMPv2 MIB view


Catalyst_1(config)#no snmp
Catalyst_1(config)#snmp enable informs
Catalyst_1(config)#snmp enable traps
Catalyst_1(config)#exit
Catalyst_1#
```

5.10 Die MAC-Adressentabelle

Suat Algin

Jeder Switch besitzt eine MAC-Adressentabelle in der die MAC-Adresse, die VLAN-ID und die dazugehörigen Ports gespeichert werden. Jede MAC-Adresse wird mit mindestens einem VLAN und einem oder mehreren Ports assoziiert. Es existieren drei verschiedene Möglichkeiten der Aufnahme einer MAC-Adresse in diese Tabelle:

- Dynamic address: Diese Adressen werden durch die Übermittlung von Paketen gelernt und im Falle von Inaktivität wieder gelöscht.
- Secure address: Eine Unicast-Adresse, welche mit einem „secured port“ assoziiert und manuell eingegeben wird. Diese Adressen besitzen keine bestimmte Lebensdauer und werden nicht gelöscht.
- Static Address: Eine statische Unicast-/Multicast-Adresse, welche manuell eingegeben wird und selbst im Falle eines Resets nicht gelöscht wird.

Im folgenden Abschnitt werden die verschiedenen Konfigurationsmöglichkeiten der drei Varianten gezeigt.
Konfigurierung CMS vs. CLI:

Die Einstellungen der drei Möglichkeiten über die CMS erfolgt über die im selben Menüpunkt befindlichen Karteireitern.

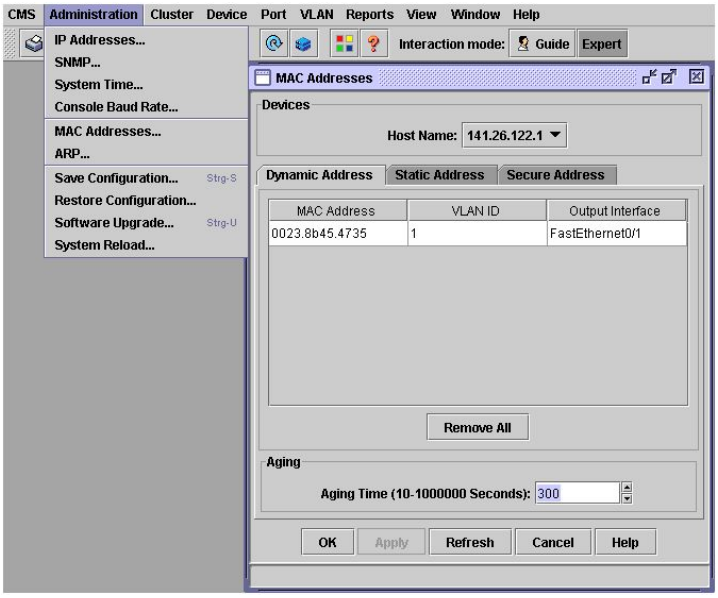

Abbildung 41: MAC-Adressentabelle(CMS)

Konfigurierung über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#show mac-address-table
Dynamic Address Count:                  1
Secure Address Count:                   0
Static Address (User-defined) Count:    0
System Self Address Count:              75
Total MAC addresses:                    76
Maximum MAC addresses:                  8192
Non-static Address Table:
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0023.8b45.4735       Dynamic          1  FastEthernet0/1
```

Die Konfigurierung der Alterungszeit über die CMS erfolgt über die in Abbildung 41 gezeigte Tabelle. Dabei wird die Zeit in Sekunden angegeben.

Die Modifikation über das CLI erfolgt folgendermaßen:

```
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#mac-address aging-time 300
141.26.122.1(config)#end
141.26.122.1#show mac-address-table aging-time
300
141.26.122.1#
```

Enfernung der dynamisch erfassten Mac-Adressen aus der MAC-Adressentabelle:

```
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#no mac-address-table dynamic 0023.8b45.4735
141.26.122.1(config)#end
141.26.122.1#show mac-address-table
Dynamic Address Count:                  0
Secure Address Count:                   0
Static Address (User-defined) Count:    0
System Self Address Count:              75
Total MAC addresses:                    76
Maximum MAC addresses:                  8192
Non-static Address Table:
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
```

Die Festlegung als „Secure Address“:
Die Konfigurierung erfolgt über zwei Schritte und hat zur Folge, dass alle Pakete von diesem Port nur noch an einen bestimmten Port(„Output Interface“) weitergeleitet werden.
Zuerst erfasst man die Adresse in der MAC-Adressentabelle:

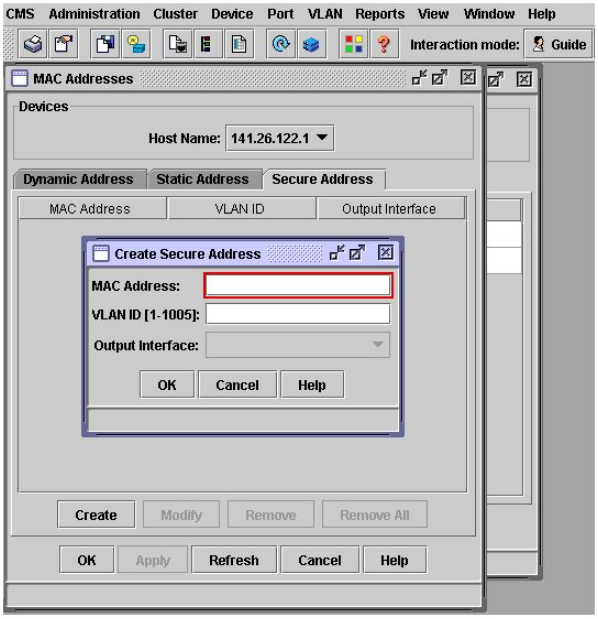

Abbildung 42: Secure Address(MAC-Adressentabelle)

Zusätzlich muss man den zugehörigen Port als sicher deklarieren:

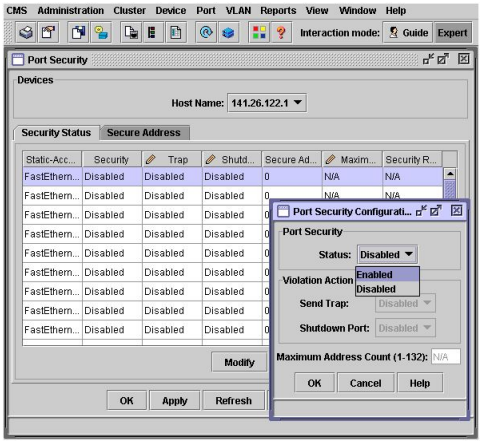

Abbildung 43: Secure Port(Port Security)

Administration der sicheren MAC-Adressen/Ports über das CLI:

```
141.26.122.1#configure terminal
...122.1(config)#mac-address-table secure 0004.0e67.2bc5 fa0/2 vlan 1
141.26.122.1(config)#interface fa0/1
141.26.122.1(config-if)#port security
141.26.122.1(config)#end
141.26.122.1#show mac-address-table secure
Non-static Address Table:
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0004.0e67.2bc5       Secure           1  FastEthernet0/2
141.26.122.1#
```

Die Entfernung von solchen sicheren Ports erfolgt mit dem voran gestellten „no“-Befehl. Zum Beispiel:

```
141.26.122.1(config)#no mac-address-table secure 0004.0e67.2bc5 vlan 1
```

Die Erfassung einer statischen MAC-Adresse in der CMS erfolgt wieder über die in Abbildung 41 gezeigte Tabelle. Zusätzlich gibt man bei einer statischen Adresse die Ports an, an die empfangene Pakete weitergeleitet werden sollen. Ein statischer Port in einem VLAN, muss auch ein staticher Port in einem anderen VLAN sein. Empfängt ein VLAN, in dem die statische Adresse nicht manuell erfasst wurde, ein Paket mit einer statischen Adresse wird diese an alle Ports im VLAN weitergeleitet.
Manuelle Eingabe einer statischen Adresse über die CMS:

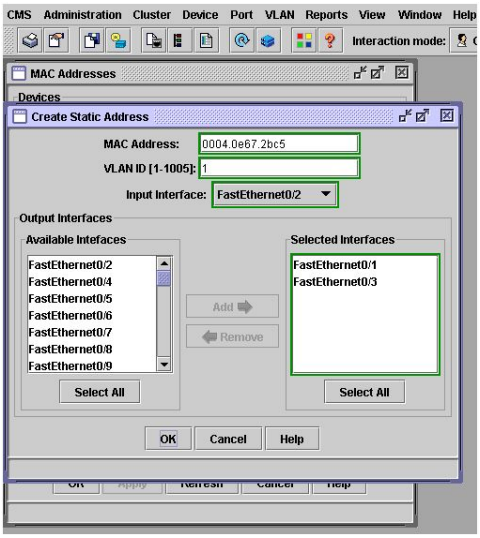

Abbildung 44: Static Ports

Die Erfassung über das CLI sieht wie folgt aus:

```
141.26.122.1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#
...(config)#mac-address-table static 0004.0e67.2bc5 fa0/2 fa0/1

141.26.122.1(config)#end
141.26.122.1#show mac-address-table static
Static Address Table:
Destination Address   VLAN  Input Port  Output Ports
-------------------   ----  ----------  -----------------------
0004.0e67.2bc5           1  Fa0/2       Fa0/1 Fa0/3
```

5.11 Cisco Group Management Protocol(CGMP)

Suat Algin

Das Cisco Group Management Protokoll verhindert, dass IP Multicast Pakete an alle Ports weitergeleitet werden, in dem es diese Pakete nur an die Ports weiterleitet, die es auch benötigen. Dazu wird eine CGMP-Gruppe erzeugt und die Multicast Pakete werden auch nur an diese Gruppe weitergeleitet.
In diesem Zusammenhang existiert die Option „Fast Leave“, die das Löschen von ungenutzten CGMP Gruppen beschleunigt. Um Teil dieser Gruppe zu werden bzw. aus dieser Gruppe auszusteigen, senden die Endstationen sogenannte „Join“ bzw. „Leave“ Nachrichten. Eine CGMP-Gruppe gilt VLAN weit. Das heißt IP Multicast Pakete können in einem VLAN an eine bestimmte Liste von Ports gesendet werden und in einem anderen VLAN an eine ganz andere Liste von Ports. Das hinzufügen oder entfernen von CGMP Gruppen gilt immer nur für ein bestimmtes VLAN.

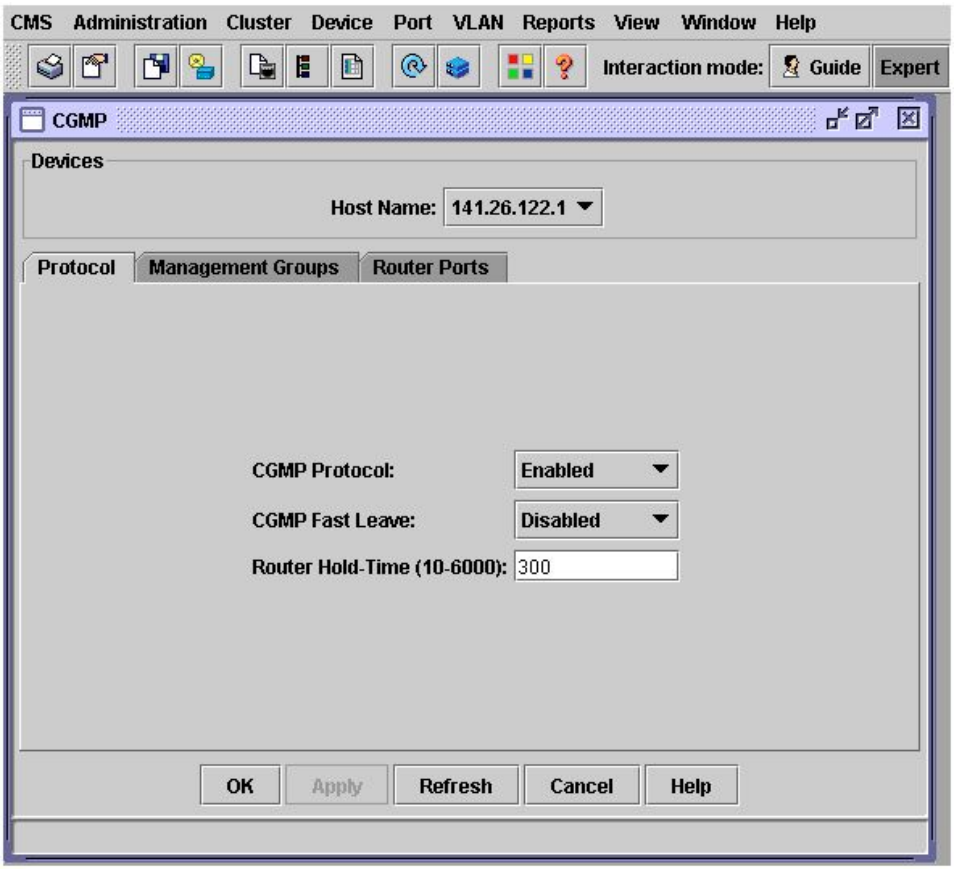

Abbildung 45: Konfigurierung der CGMP Optionen

In dem in Abbildung 45 gezeigtem Menü können folgende Einstellungen vorgenommen werden:

- Ein-/Ausschalten des CGMP
- Ein-/Ausschalten der Fast-Leave Option
- Entfernen von Multicast Gruppen
- Entfernen von Router Ports
- Änderung der Router Hold-Time

Konfigurierung der CGMP Option über das CLI:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#cgmp leave-processing
141.26.122.1(config)#cgmp holdtime 360
141.26.122.1(config)#end
141.26.122.1#clear cgmp group
141.26.122.1#clear cgmp router
141.26.122.1#
```

Das Ausschalten des CGMP erfolgt wie üblich über das voranstellen des „no“-Befehls:

```
User Access Verification

Password:
141.26.122.1>enable
Password:
141.26.122.1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
141.26.122.1(config)#no cgmp leave-processing
141.26.122.1(config)#end
141.26.122.1#
```

5.12 Cisco Discovery Protocol

Suat Algin

Das „Cisco Discovery Protocol“ arbeitet auf der zweien Schicht des OSI- Referenzmodells und dient zur Entdeckung von angeschlossenen Netzwerkgeräten. Dabei senden alle CDP-fähigen Geräte in regelmäßigen Abständen Nachrichten an eine Ethernet-Multicast-Adresse(01:00:0C:CC:CC:CC) und empfangen gleichzeitig Informationen von benachbarten Geräten. Diese Nachrichten enthalten Informationen über die zugehörigen Ports, die Host-Namen, die Betriebssystemversionen, die Gerätenamen und die Management IP-Adressen. Zusätzlich befindet sich unter den Informationen eine sogenannte Holdtime. Diese Holdtime beinhaltet die Zeit, bis alle Informationen über einen „Neighbour-Switch“ verworfen werden. Durch diesen Parameter wird gewährleistet, dass der Ausfall einer Netzwerk-Komponente durch fehlende CDP-Pakete erkannt und nach Ablauf der Holdtime die gespeicherten Informationen aus den aufgebauten CDP-Tabellen der Switches gelöscht werden. Anhand dieser Informationen wird auch die grafische Darstellung der „Cluster Management Suite“ realisiert. Zusätzlich kann der Administrator den Cisco eigenen CDP-Monitor zur Fehlerbeseitigung nutzen. Der Nachteil an diesem Protokoll ist, dass es keine Sicherheitsmechanismen besitzt und dadurch leicht gefälscht bzw. auch abgehört werden kann. So können Angreifer z.B. den Switch mit gefälschten CDP-Paketen fluten(Foolding) und so die Konfigurierung des Switches erschweren. Aber auch das Abschalten kann zu Komplikationen führen, weil diese Informationen von einigen IP-Telefonen verschiedener Marken zur Modifizierung Ihrer Konfigurationen genutzt werden. Deshalb sollte in so einem Fall gründlich überlegt werden, ob man das CDP ausschaltet.
Trotzdem sollte man im Falle von sicherheitskritischen Interfaces, welche öffentlich zugänglich sind, CDP ausschalten. CDP kann sowohl für jedes einzelne Interface, wie auch für das gesamte System eingestellt werden.

Befehle zur Konfiguration des CDP über das Cisco CLI:

```
User Access Verification

Password:
Cat_141.26.122.1>enable
Password:
Cat_141.26.122.1#
```

- Statusanzeige des CDP: Hierbei werden Informationen üder das Sendeintervall der CDP-Pakete, die Aufbewahrungszeit der Infos und den Status der IP-Phone Erkennung ausgegeben.

```
Cat_141.26.122.1#show cdp
Global CDP information:
```

```
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled
Cat_141.26.122.1#
```

- Aktivierung des CDP-Prozesses: Mit dieser Befehlszeile wird das Cisco Discovery Protokoll aktiviert bzw. deaktiviert

```
Cat_141.26.122.1#config terminal
Cat_141.26.122.1(config)#cdp run
Cat_141.26.122.1(config)#exit

Cat_141.26.122.1(config)#no cdp run
Cat_141.26.122.1(config)#exit
Cat_141.26.122.1#show cdp
CDP ist not enabled
Cat_141.26.122.1#
```

- Aktivierung des CDP für einzelne Interfaces:

```
Cat_141.26.122.1#config terminal
Cat_141.26.122.1(config)#interface fa0/1
Cat_141.26.122.1(config-if)#cdp enable
Cat_141.26.122.1(config-if)#interface fa0/2
Cat_141.26.122.1(config-if)#cdp enable
Cat_141.26.122.1(config-if)#exit
Cat_141.26.122.1(config)#exit
Cat_141.26.122.1#
```

- Konfigurierung des Sendeintervalls der CDP-Updates:
  (Minimal 5 Sekunden / Maximal 254 Sekunden)

```
Cat_141.26.122.1#config terminal
Cat_141.26.122.1(config)#cdp timer 30
Cat_141.26.122.1(config)#exit
Cat_141.26.122.1#
```

- Konfigurierung der Verwerfungszeit der CDP-Pakete:
  (Minimal 10 Sekunden / Maximal 255 Sekunden)

```
Cat_141.26.122.1#config terminal
Cat_141.26.122.1(config)#cdp holdtime 180
Cat_141.26.122.1(config)#exit
Cat_141.26.122.1#
```

- CDP-Informationen aus der CDP-Tabelle löschen:

```
Cat_141.26.122.1#clear cdp table
```

- Anzeige des CDP-Datenverkehrs:

```
Cat_141.26.122.1#show cdp traffic
CDP counters :
 Total packets output: 158, Input: 0
 Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
 No memory: 0, Invalid packet: 0, Fragmented: 0
 CDP version 1 advertisements output: 0, Input: 0
 CDP version 2 advertisements output: 158, Input: 0
 Trigger responses sent: 0, throttled: 0
Cat_141.26.122.1#
```

- Anzeige der Informationen über benachbarte CDP-Geräte:

```
Cat_141.26.122.1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID     Local Intrfce  Holdtme    Capability  Platform  Port ID
141.26.124.1  Fas 0/1         130         T S       WS-C3548-XFas 0/7
141.26.125.1  Fas 0/13        175         T S       WS-C3548-XFas 0/6
Cat_141.26.122.1#
```

- Anzeige der Informationen über ein bestimmtes CDP-Gerät[23]:

```
Cat_141.26.122.1#show cdp entry 141.26.124.1
-------------------------
Device ID: 141.26.124.1
Entry address(es):
  IP address: 141.26.124.1
  IP address: 141.26.124.1
Platform: cisco WS-C3548-XL,  Capabilities: Trans-Bridge Switch
Interface: FastEthernet0/1,  Port ID (outgoing port): FastEthernet0/7
Holdtime : 151 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) C3500XL Software (C3500XL-C3H2S-M), Version 12.0(5)WC17,
                        RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2007 by cisco Systems, Inc.
```

[23](* um alle CDP-Geräte oder Gerätenamen anzuzeigen)

```
Compiled Tue 13-Feb-07 15:04 by antonino

advertisement version: 2
Protocol Hello:  OUI=0x00000C, Protocol ID=0x0112; payload len=27,
value=0A770F4000000000010123FF000653770F40000653770F40000001
VTP Management Domain: 'mydomain'
Native VLAN: 1
Duplex: full
Management address(es):
  IP address: 141.26.124.1
  IP address: 141.26.124.1
Cat_141.26.122.1#
Cat_141.26.122.1#show cdp entry *
-------------------------
Device ID: 141.26.124.1
Entry address(es):
  IP address: 141.26.124.1
  IP address: 141.26.124.1
Platform: cisco WS-C3548-XL,  Capabilities: Trans-Bridge Switch
Interface: FastEthernet0/1,  Port ID (outgoing port): FastEthernet0/7
Holdtime : 166 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) C3500XL Software (C3500XL-C3H2S-M), Version 12.0(5)WC17,
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Tue 13-Feb-07 15:04 by antonino

advertisement version: 2
Protocol Hello:  OUI=0x00000C, Protocol ID=0x0112; payload len=27,
value=0A770F4000000000010123FF000653770F40000653770F40000001
VTP Management Domain: 'mydomain'
Native VLAN: 1
Duplex: full
Management address(es):
  IP address: 141.26.124.1
  IP address: 141.26.124.1

-------------------------
Device ID: 141.26.125.1
Entry address(es):
  IP address: 141.26.125.1
  IP address: 141.26.124.1
Platform: cisco WS-C3548-XL,  Capabilities: Trans-Bridge Switch
Interface: FastEthernet0/13,  Port ID (outgoing port): FastEthernet0/6
Holdtime : 172 sec

Version :
Cisco Internetwork Operating System Software
```

```
IOS (tm) C3500XL Software (C3500XL-C3H2S-M), Version 12.0(5)WC17,
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Tue 13-Feb-07 15:04 by antonino

advertisement version: 2
Protocol Hello:  OUI=0x00000C, Protocol ID=0x0112; payload len=27,
value=0A3ECE8000000002010123FF000653770F400005DD3ECE80010001
VTP Management Domain: 'mydomain'
Native VLAN: 1
Duplex: full
Management address(es):
  IP address: 141.26.124.1
  IP address: 141.26.125.1

141.26.122.1#
```

Anzeige der Informationen über ein Interface oder über alle Interfaces auf denen CDP aktiviert ist:

```
Cat_141.26.122.1#show cdp interface
FastEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 30 seconds
  Holdtime is 180 seconds
FastEthernet0/2 is down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 30 seconds
  Holdtime is 180 seconds
.
.
FastEthernet0/5 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
.
.
FastEthernet0/13 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Cat_141.26.122.1#
Cat_141.26.122.1#show cdp interface fa0/1
FastEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 30 seconds
  Holdtime is 180 seconds
Cat_141.26.122.1#
```

6 Virtual Square

Serdar Ayalp

Virtual Square (VS) ist der Name eines Projektes, das mehrere Tools und Bibliotheken beinhaltet.VS wurde als Forschungsprojekt gestartet und bildet derzeit ein Framework, in dem mehrere Projekte parallel entwickelt werden. Obwohl sich virtuelle Systeme auf gleiche Konzepte beziehen, sind sie meistens verschiedenartig bzw. heterogen und miteinander inkompatibel. Das Ziel des VS ist eine Umgebung mit virtuellen Maschinen (VMs), Systemen und Netzwerken zu schaffen, die miteinander kommunizieren und interagieren. Also die Bereitstellung einer Umgebung, in der verschiedenartige VMs miteinander kommunizieren können.
Richtlinien des Projektes sind:

- **Wiederverwendung der existierenden Tools**: VS-Tools erweitern bzw. verbessern die existierenden Tools.
- **Modularität**: Jedes Tool wird nach einer modularen Architektur hergestellt.
- **Keine Architektur-Beschränkung**: VS-Tools sind konfigurierbar und adaptierbar für unterschiedliche Umgebungen und Applikations-Bedürfnisse. Durch die große Auswahl kann sich der Benutzer entscheiden, welches Tool für seine Zwecke geeignet ist.

Alle in diesem Projekt geschriebenen Programme werden auf frei zugänglichen „Repositories“ unter einer freien Lizenz zur Verfügung gestellt. Jeder kann sich die VS-Tools herunterladen und die Effektivität der vorgeschlagenen Konzepte bzw. Ideen testen.

6.1 Virtual Distributed Ethernet (VDE)

VDE ist eines der Tools, das in dem Virtual-Square-Projekt entwickelt wurde, um eine Kommunikationsplattform für VMs bereitzustellen.
Die Haupteigenschaften von VDE sind :

- konsistentes Verhalten mit realen Ethernet-Netzwerken.
- ermöglicht Verbindungen zwischen VMs, Applikationen und Virtual Verbindung-Tools (z.B. tun/tap)
- zu guter Letzt erfordert es keine administrativen Rechte, um zu funktionieren.

6.1.1 Haupt-Komponenten des VDE

Serdar Ayalp

In der Praxis besteht die Struktur eines VDE-Netzwerks aus den gleichen Elementen eines realen Netzes.

- **VDE-Switch :** Wie ein physikalischer Switch hat ein VDE-Switch virtuelle Ports, an die verschiedene VMs, Applikationen, virtuellen Interfaces bzw. Schnittstellen, Verbindungstools oder auch andere VDE-Switches angeschlossen werden können.
- **VDE-Plug :** Die zu einem VDE-Switch kommenden bzw. von einem VDE-Switch ausgehenden Daten kontrollierende Socket-ähnliche Programme.
- **VDE-Wire :** Das Tool, das für die Daten-Übertragung zuständig ist.
- **VDE-Cable:** VDE-Komponenten werden miteinander per VDE-Cable verbunden, das aus einem VDE-Wire und zwei VDE-Plugs besteht.

6.1.2 Installation des VDE

Serdar Ayalp

Der Source-Code des VDE lässt sich als tar-Datei von der Homepage herunterladen oder per SVN ausfindig machen. Derzeit funktioniert VDE leider nur unter Linux. Die auf der Seite verfügbaren tar-Dateien sind meistens veraltet. Deshalb ist zu empfehlen, den aktuellsten Source-Code mit einem SVN-Client herunterzuladen.

```
$svn co https://vde.svn.sourceforge.net/svnroot/vde vde
```

Danach wechselt man in den Speicherort und fährt mit der Installion fort.

```
~$ cd vde/trunk/vde-2
vde/trunk/vde-2$ autoreconf --install
vde/trunk/vde-2$ ./configure --enable-experimental
vde/trunk/vde-2$ make
vde/trunk/vde-2$ make install
```

6.2 VDE-Switch

Serdar Ayalp

Ein VDE-Switch verbindet mehrere virtuelle Geräte miteinander. VDE-Switches können auch miteinander per VDE-Cable verbunden werden.
Haupt-Eigenschaften eines VDE-Switches sind :

- **VLAN :** Ermöglicht, die Ports eines Switches in Gruppen zu unterteilen. Jede Gruppe wird als Virtual LAN bzw. VLAN bezeichnet. Mit dieser logischen Aufteilung ist es möglich, mehrere voneinander unabhängige virtuelle Netze zu erzeugen.
- **Fast Spanning Tree Protocol (FSTP) :** wurde implementiert, um Loops in einem VDE-Switch zu verhindern. Genauso wie in realen Netzen erzeugt dieses Protokoll einen *Span-Baum*, in dem redundate Pfade blockiert werden.
- **Command Line Management :** CLM ist nützlich, um VLANs zu erzeugen, FSTP zu aktivieren, Switch Ports zu überwachen oder den Switch-Status zu kontrollieren.

Da VDE-Switch das zentrale Element in der virtuellen Netzwerkstruktur ist,wurde es auch so flexibel wie möglich konzipiert.Wenn ein VDE-Switch gestartet wird, können einige Eigenschaften vom User angepasst werden.
Diese sind :

- Die Anzahl der Ports
- Management Socket und Rechte
- Data Socket und Rechte
- Tap Interface

Um einen VDE-Switch zu erzeugen, gibt man den Befehl *vde_switch* ein.

```
vde/trunk/vde-2$vde_switch
vde$
```

Wenn man als Parameter keinen Wert eingegeben hat, wird die Default-Datei (bzw. *Management-Socket*) */tmp/vde.ctl* erzeugt. Durch drücken der Return-Taste erscheint der Management-Prompt des VDE-Switches, in dem man durch einen *help*-Befehl alle verfügbaren Befehle anzeigen lassen kann.

```
vde$ help
0000 DATA END WITH '.'
COMMAND PATH    SYNTAX          HELP
------------    --------------  ------------
ds              ============    DATA SOCKET MENU
ds/showinfo                     show ds info
help            [arg]           Help (limited to arg when specified)
logout                          logout from this mgmt terminal
shutdown                        shutdown of the switch
showinfo                        show switch version and info
```

```
load            path            load a configuration script
debug           ============    DEBUG MENU
debug/list                      list debug categories
debug/add       dbgpath         enable debug info for a given category
debug/del       dbgpath         disable debug info for a given category
plugin          ============    PLUGINS MENU
plugin/list                     list plugins
plugin/add      library         load a plugin
plugin/del      name            unload a plugin
hash            ============    HASH TABLE MENU
hash/showinfo                   show hash info
hash/setsize    N               change hash size
hash/setgcint   N               change garbage collector interval
hash/setexpire N                change hash entries expire time
hash/setminper N                minimum persistence time
hash/print                      print the hash table
hash/find       MAC [VLAN]      MAC lookup
fstp            ============    FAST SPANNING TREE MENU
fstp/showinfo                   show fstp info
fstp/setfstp    0/1             Fast spanning tree protocol 1=ON 0=OFF
fstp/setedge    VLAN PORT 1/0   Define an edge port for a vlan 1=Y 0=N
fstp/bonus      VLAN PORT COST  set the port bonus for a vlan
fstp/print      [N]             print fst data for the defined vlan
port            ============    PORT STATUS MENU
port/showinfo                   show port info
port/setnumports N              set the number of ports
port/sethub     0/1             1=HUB 0=switch
port/setvlan    N VLAN          set port VLAN (untagged)
port/create     N               create the port N (inactive|notallocatable)
port/remove     N               remove the port N
port/allocatable N 0/1          Is the port allocatable as unnamed? 1=Y 0=N
port/setuser    N user          access control: set user
port/setgroup   N user          access control: set group
port/epclose    N ID            remove the endpoint port N/id ID
port/resetcounter [N]           reset the port (N) counters
port/print      [N]             print the port/endpoint table
port/allprint   [N]             print the port/endpoint table
                                (including inactive port)
vlan            ============    VLAN MANAGEMENT MENU
vlan/create     N               create the VLAN with tag N
vlan/remove     N               remove the VLAN with tag N
vlan/addport    N PORT          add port to the vlan N (tagged)
vlan/delport    N PORT          add port to the vlan N (tagged)
vlan/print      [N]             print the list of defined vlan
vlan/allprint   [N]             print the list of defined vlan
                                (including inactive port)
.
1000 Success

vde$
```

Der Switch lässt sich auch als „daemon“-prozess starten.

```
vde_switch --daemon --sock /tmp/myvde.ctl --mgmt /tmp/myvde.mgmt
```

Um in diesem Fall das Command-Line-Management-Interface (CLMI) zu erreichen, muss man das dafür zuständige Tool (*unixterm*) nutzen.

```
~/vde/trunk/vde-2$ unixterm /tmp/myvde.mgmt
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$
```

Es ist auch möglich den VDE-Switch mit dem tap-interface des Betriebsystems zu verbinden. Wegen den Sicherheitsproblemen ist die Erzeugung und die Konfiguration eines tun/tap-interfaces eingeschränkt. Um einem User das Öffnen und Nutzen einer tun/tap-Datei unter dem Pfad /dev/net/tun zu ermöglichen, muss der Administrator ein persistentes tap-interface vorkonfigurieren. Das lässt sich mit einem Tool wie ***tunctl***, welches als „uml“-Werkzeug mit User Mode Linux mitgeliefert wird, erzeugen. Aber es ist dem User nicht gestattet, die grundlegenden Aspekte des Interfaces zu verändern.

```
root#  tunctl -u root -t tap0
Set 'tap0' persistent and owned by uid 0
root#
```

Ist das tap-interface erstellt und konfiguriert, ist der User in der Lage, einen VDE-Switch zu starten und ihn mit den jeweiligen Interfaces zu verbinden.

```
~/vde/trunk/vde-2$ vde_switch --tap tap0

vde$

vde$ port/allprint

vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          0          bytes                    0
 OUT: pkts          0          bytes                    0
  -- endpoint ID 0005 module tuntap      : tap0
.
1000 Success

vde$
```

6.2.1 Management-Befehle eines VDE-Switches

Serdar Ayalp

Wie nach dem *help*-Befehl zu sehen ist, sind die Konfigurations-Befehle in Bereiche unterteilt.

1. **Allgemeine Befehle** :

   - ***help [Thema]***: listet alle verfügbaren Switch-Befehle auf [24]. Die Ausgabe lässt sich auf ein spezielles Thema einschränken. z.B :

```
vde$ help vlan
0000 DATA END WITH '.'
COMMAND PATH        SYNTAX        HELP
------------        ------------  ------------
vlan                ===========   VLAN MANAGEMENT MENU
vlan/create         N             create the VLAN with tag N
vlan/remove         N             remove the VLAN with tag N
vlan/addport        N PORT        add port to the vlan N (tagged)
vlan/delport        N PORT        add port to the vlan N (tagged)
vlan/print          [N]           print the list of defined vlan
vlan/allprint       [N]           print the list of defined vlan
                                  (including inactive port)
.
1000 Success

vde$
```

   - ***shutdown*** : Terminiert den Switch-Prozess.

```
vde$ shutdown
vde_switch: Shutdown from mgmt command
~/vde/trunk/vde-2$
```

   - ***showinfo*** : Allgemeine Informationen über den Switch, wie z.B. Process-ID, MAC-Adresse, die Länge der noch nicht gesendeten Paketen etc.

```
vde$ showinfo
0000 DATA END WITH '.'
VDE switch V.2.2.3
(C)Virtual Square Team(coord. R. Davoli)2005,2006,2007-GPLv2

pid 8725 MAC 00:ff:3a:23:78:73 uptime 1555
unsent_pktq_len 0
.
1000 Success

vde$
```

[24](siehe 6.2)

- ***load filename***: Ermöglicht dem User, alle Management-Befehle in einer Datei einzugeben und alles direkt aus dieser Datei auszuführen.

2. **Data-Sockets-Befehle**:
   - ***ds/showinfo***: Listet generelle Informationen über das Data-Socket, dessen Path und Access-Mode auf.

3. **Hash-Tabelle-Befehle**: Die Speicherung der MAC-Adressen im VDE-Switch wird durch eine Hash-Tabelle bewältigt.
   - ***hash/showinfo***: Gibt Einzelheiten der Hash-Tabelle, wie z.B. die Größe der Tabelle, Zeitintervall für *garbage collector* etc., aus.

     ```
     vde$ hash/showinfo

     0000 DATA END WITH '.'
     Hash size 128
     GC interval 2 secs
     GC expire 100 secs
     Min persistence 3 secs
     .
     1000 Success

     vde$
     ```

   - ***hash/setsize*** *size*: Passt die Größe der Hash-Tabelle an Netze, die mehrere Hosts bzw. MAC-Adressen beinhalten können, an.

     ```
     vde$ hash/setsize 256
     1000 Success

     vde$ hash/showinfo
     0000 DATA END WITH '.'
     Hash size 256
     GC interval 2 secs
     GC expire 100 secs
     Min persistence 3 secs
     .
     1000 Success

     vde$
     ```

   - ***hash/setgcint*** *sekunde*: Ändert das Zeitintervall für den *Garbage-Collector* (GC). Nach Ablauf dieses Intervalls schaut der GC, welche Einträge in der Hash-Tabelle zu löschen sind.

     ```
     vde$ hash/setgcint 4
     1000 Success
     ```

```
vde$ hash/showinfo
0000 DATA END WITH '.'
Hash size 256
GC interval 4 secs
GC expire 100 secs
Min persistence 3 secs
.
1000 Success

vde$
```

- ***hash/setexpire*** *sekunde*: Für jeden zur Tabelle eingefügten Eintrag wird ein Zähler zur Verfügung gestellt. Jede Sekunde wird der Wert dieses Zählers um eins erhöht, bis er den Wert überschreitet, der mit diesem Befehl eingestellt wurde. In diesem Fall wird der Eintrag in der Hash-Tabelle vom GC gelöscht.

```
vde$ hash/setexpire 200
1000 Success

vde$ hash/showinfo
0000 DATA END WITH '.'
Hash size 256
GC interval 4 secs
GC expire 200 secs
Min persistence 3 secs
.
1000 Success

vde$
```

- ***hash/print***: Druckt den ganzen Inhalt der Tabelle aus. Für jeden Eintrag in der Tabelle werden Informationen über Eintrag-ID (Hash-Key), MAC-Adresse, Port-Nummer und Alter des Eintrags aufgelistet.

```
vde$ hash/print
0000 DATA END WITH '.'
Hash: 0011 Addr: 00:ff:38:2f:af:dd VLAN 0000 to port: 001
 age 187 secs
.
1000 Success

vde$
```

- ***hash/find*** *MAC-Adresse [VLAN-id]*: Findet den Eintrag, der die angegebene MAC-Adresse hat. Optional kann man auch die VLAN-ID eingeben.

4. **Befehle des Fast-Spanning-Tree-Protocol(FSTP)**:

   - ***fstp/showinfo***:Allgemeine Informationen über die FSTP-Implementation im VDE-Switch.
     Diese sind:

     a) MAC-Adresse des Switches

     b) Die Switch-Priorität (Default: 32768, Cisco-Standard)

     c) FSTP aktiviert oder deaktiviert?

     ```
     vde$ fstp/showinfo
     0000 DATA END WITH '.'
     MAC 00:ff:fa:19:c3:25 Priority 32768 (0x8000)
     FSTP=false
     .
     1000 Success

     vde$
     ```

   - ***fstp/setfstp*** *0/1*: Aktivierung oder Deaktivierung des FSTPs.

     ```
     vde$ fstp/setfstp 1
     0000 DATA END WITH '.'
     .
     1000 Success

     vde$ fstp/showinfo
     0000 DATA END WITH '.'
     MAC 00:ff:fa:19:c3:25 Priority 32768 (0x8000)
     FSTP=true
     .
     1000 Success

     vde$
     ```

   - ***fstp/setedge*** *VLAN-id port-num 0/1*: Konfiguriert den Port für die gegebene VLAN-ID als *Edge-Port*. Edge-Ports sind die Ports, die direkt zu einem Endsystem angeschlossen sind. Diese Ports werden beim Aufbau des Baums nicht in Betracht gezogen und werden direkt in den Forwarding-Zustand gesetzt. Also werden die vorherigen Phasen (listening und learning) übersprungen. Damit ist eine schnellere Konvergenz zu einer stabilen Topologie im Netz möglich.

   - ***fstp/print [VLAN-id]***: Der aktuelle Status des STPs für die angegebene VLAN-ID.

     ```
     vde$ fstp/print 0
     0000 DATA END WITH '.'
     FST DATA VLAN 0000 ROOTSWITCH
     ```

```
 ++ root 80:00:00:ff:fa:19:c3:25
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 4181 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
.
1000 Success

vde$
```

- ***fstp/bonus*** *VLAN-id port-num cost*: Kosten-Zuweisung für den angegebenenen Port.

```
vde$ fstp/bonus 0 0 20
1000 Success

vde$ fstp/print 0
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH
 ++ root 80:00:00:ff:fa:19:c3:25
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 4181 bonusport 0000 bonuscost 20
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
.
1000 Success

vde$
```

5. **Befehle für das Port-Management**:
   - ***port/showinfo***: Auflistung der Anzahl der verfügbaren Ports und die Angabe ob es sich um einen Switch oder einen Hub handelt.

```
vde$ port/showinfo
0000 DATA END WITH '.'
Numports=32
HUB=false
counters=true
.
1000 Success

vde$
```

   - ***port/setnumports*** *number*: Änderung der Anzahl der Ports.

```
vde$ port/setnumports 48
1000 Success

vde$ port/showinfo
0000 DATA END WITH '.'
Numports=48
HUB=false
```

```
counters=true
.
1000 Success

vde$
```

- ***port/sethub*** *0/1*: Setzt den VDE-Switch in den Hub Modus. Damit werden alle Frames wie erwartet überall broadcastet.

```
vde$ port/sethub 1
1000 Success

vde$ port/showinfo
0000 DATA END WITH '.'
Numports=48
HUB=true
counters=true
.
1000 Success

vde$
```

- ***port/setvlan*** *port-num VLAN-id* : Der jeweilige Port wird zum angegebenen VLAN zugewiesen. In diesem Fall darf dieser Port nur zu diesem VLAN gehören und nicht auch noch zu einem anderen.
- ***port/create*** *port-num*: Erzeugt einen neuen Port, obwohl nichts angeschlossen ist.

```
vde$ port/create 3
1000 Success

vde$
```

- ***port/remove*** *port-num*: Löscht den jeweiligen Port.
- ***port/allocatable*** *port-num 0/1*: Der Port wird als reserviert gekennzeichnet. Es ist nicht möglich etwas an diesen Port anzuschließen, ohne exakt die Nummer dieses Ports anzugeben.
- ***port/print*** *[port-num]*: Alle aktiven Ports werden mit ihren Informationen aufgelistet.

```
vde$ port/print
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          7          bytes                  574
 OUT: pkts        509          bytes                26977
  -- endpoint ID 0005 module tuntap      : MeinTunTap01
.
```

```
1000 Success

vde$
```

- ***port/allprint*** *[port-num]*: Alle Ports werden mit ihren Informationen aufgelistet.

```
vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          7          bytes                  574
 OUT: pkts        509          bytes                26977
  -- endpoint ID 0005 module tuntap      : MeinTunTap01
Port 0002 untagged_vlan=0000 INACTIVE - NOT Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          0          bytes                    0
 OUT: pkts          0          bytes                    0
Port 0003 untagged_vlan=0000 INACTIVE - NOT Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          0          bytes                    0
 OUT: pkts          0          bytes                    0
Port 0004 untagged_vlan=0000 INACTIVE - NOT Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          0          bytes                    0
 OUT: pkts          0          bytes                    0
.
1000 Success

vde$
```

- ***port/epclose*** *port-num endpoint-id*: Das End-Gerät (engl. endpoint), welches über den angegebenen Port erreichbar ist, fällt aus.

6. **Befehle für VLANs** :

- ***vlan/print*** *[VLAN-id]*: Alle VLANs werden aufgelistet.

```
vde$ vlan/print
0000 DATA END WITH '.'
VLAN 0000
VLAN 0001
.
1000 Success

vde$
```

- ***vlan/allprint*** *[VLAN-id]*: Alle VLANs werden mit ihren Ports aufgelistet.

```
vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
```

```
 -- Port 0001 tagged=0 active=0 status=Learning
 -- Port 0002 tagged=0 active=0 status=Learning
 -- Port 0003 tagged=0 active=0 status=Learning
 -- Port 0004 tagged=0 active=0 status=Learning
VLAN 0001
.
1000 Success

vde$
```

- ***vlan/create*** *VLAN-id*: Erzeugt ein neues VLAN mit der angegebenen VLAN-ID.

```
vde$ vlan/create 3
1000 Success

vde$ vlan/print
0000 DATA END WITH '.'
VLAN 0000
VLAN 0001
VLAN 0003
.
1000 Success

vde$
```

- ***vlan/remove*** *VLAN-id*: Löscht ein VLAN. Ein VLAN kann man nicht löschen, wenn es noch Ports gibt, die ihm zugewiesen sind.

```
vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0001 tagged=0 active=0 status=Learning
 -- Port 0002 tagged=0 active=0 status=Learning
 -- Port 0003 tagged=0 active=0 status=Learning
 -- Port 0004 tagged=0 active=0 status=Learning
VLAN 0001
VLAN 0003
.
1000 Success

vde$ vlan/remove 3
1000 Success

vde$

vde$ vlan/print
0000 DATA END WITH '.'
VLAN 0000
VLAN 0001
.
```

```
1000 Success

vde$

vde$ vlan/remove 0
1098 Address already in use

vde$
```

- ***vlan/addport*** *VLAN-id port-num*: Fügt den angegebenen Port, als getagten Port, zum angegebenen VLAN hinzu. Wenn dieser Port Teil eines anderen VLANs ist, ist er auch Teil des VLANs mit der VLAN-id, die als Parameter eingegeben wurde. Solche Ports sind nur dann zu benutzen, wenn VLANs, die auf mehreren Switches verteilt sind, miteinander verbunden werden müssen. Ausgehende Pakete von einem getagten Port haben einen zusätzlichen Header, der Informationen über die Quelle des Paketes beinhaltet. Diese Art der Verteilung der VLANs auf mehrere Switches wird als ***VLAN Trunking*** bezeichnet und ist Teil des Standards 802.1Q.

```
vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0001 tagged=0 active=0 status=Learning
 -- Port 0002 tagged=0 active=0 status=Learning
VLAN 0001
.
1000 Success

vde$ vlan/addport 1 2
1000 Success

vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0001 tagged=0 active=0 status=Learning
 -- Port 0002 tagged=0 active=0 status=Learning
VLAN 0001
 -- Port 0002 tagged=1 active=0 status=Learning
.
1000 Success

vde$
```

- ***vlan/delport*** *VLAN-id port-num* : Löscht den getagten Port vom angegebenen VLAN.

```
vde$ vlan/delport  1  2
```

```
1000 Success

vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0001 tagged=0 active=0 status=Learning
 -- Port 0002 tagged=0 active=0 status=Learning
VLAN 0001
.
1000 Success

vde$
```

6.3 Vernetzung per VDE-Switch

Serdar Ayalp

Mit VDE_Switch lassen sich virtuelle und physikalische Maschinen miteinander verbinden. Wie bei realen und modernen Netzwerken bestehen auch VDE_Netze aus mehreren Switches und Kabeln. Jeder Switch hat ein sogenanntes Kontroll-Socket, an den die virtuellen Maschinen zu verbinden sind.

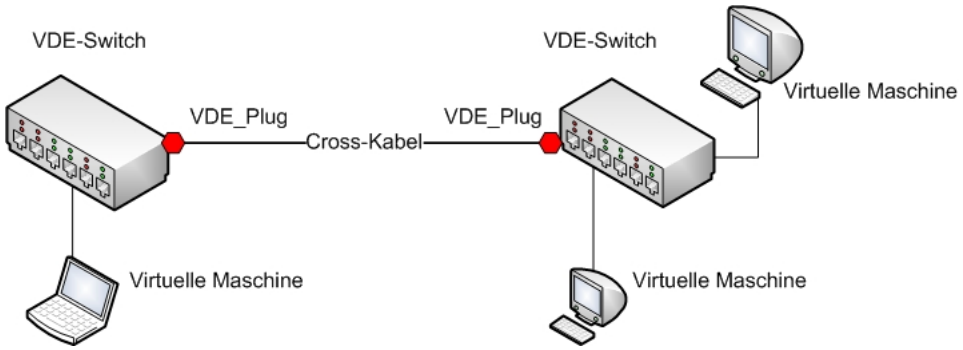

Abbildung 46: Ein Netz mit Kombination der VDE_Switches und Kabeln

1. ***Zwei Virtuelle Maschine per VDE_Switch verbinden***:

   Jeder Switch hat eine Directory, in dem er seine temporären Dateien speichert. Dieses Directory wird auch als Name des Switches verwendet. Mit der folgenden Zeile wird ein VDE_Switch als Hintergrund-Prozess (-d oder –daemon) gestartet. Nach dem Parameter *-s* wird der Pfad der Kontroll-Directory angegeben.

   ```
   ~/vde/trunk/vde-2$ vde_switch -d -s /tmp/vde1
   ```

Anschließend werden zwei virtuelle Maschinen (User-Mode-Linux) erzeugt und direkt mit dem Kontroll-Socket des Switches verbunden.

```
~/vde/trunk/vde-2$ ./linux ubd0=cow1.img,mini_vnuml-1.8.img
                           con0=xterm eth1=daemon,,,/tmp/vde1/ctl &

~/vde/trunk/vde-2$ ./linux ubd0=cow2.img,mini_vnuml-1.8.img
                           con0=xterm eth1=daemon,,,/tmp/vde1/ctl &
```

Genauso wie physikalische Maschinen besitzen auch virtuelle Maschinen Hardware, die kontrolliert werden muss. Im Gegensatz zu den realen Switches haben diese aber eine virtuelle Hardware, was bedeutet, dass man sie frei konfigurieren kann. Oben in den Befehlen weist die erste Angabe ***linux*** auf den UML-Linux-Kernel[25] hin, welchen man im Internet auf vielen Seiten herunterladen kann[26]. Die zweite Angabe, die mit ***ubd0*** anfängt, ist die image-Datei, die das Filesystem der virtuellen Maschine realisiert. Diese Datei (*mini_vnuml-1.8.img*) und ihre Varianten sind auch auf der selben Seite herunterzuladen. Die sogenannten COW (Copy-on-Write)-Dateien ermöglichen die mehrfache Verwendung einer Image-Datei. Damit lassen sich mehrere VMs mit Hilfe einer einzigen Image-Datei booten. Jede virtuelle Maschine muss aber ihre eigene COW-Datei haben und ihre Änderungen in diese schreiben[27].

2. ***Zwei VDE-Switches miteinander verbinden***:

   Auf einem Rechner kann man mehrere VDE_Switches mit unterschiedlichen Namen starten.

   ```
   ~/vde/trunk/vde-2$ vde_switch -d -s /tmp/vde1
   ~/vde/trunk/vde-2$ vde_switch -d -s /tmp/vde2
   ~/vde/trunk/vde-2$
   ```

   Mit Hilfe des ***dpipe***-Befehls ist es sehr einfach die beiden Switches miteinander zu verbinden.

   ```
   ~/vde/trunk/vde-2$ dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde2 &
   [1] 13323
   ```

   Auf diese Weise wird der Output von vde1 zum Input von vde2 und umgekehrt. Nun können alle Maschinen, die an diese Switches angeschlossen sind, Pakete austauschen.

[25] **Zitat**: User Mode Linux (UML) ist eine Variante des Linux-Kernels, die es erlaubt, komplette Linux-Kernel als Anwendungsprozesse innerhalb operierender Linux-Systeme auszuführen, ohne deren Konfiguration und damit Stabilität zu beeinflussen [15].

[26] **zu empfehlen**: http://www.dit.upm.es/vnumlwiki/index.php/Download

[27] Für weitere Informationen über die Konfiguration der virtuellen Maschinen: http://user-mode-linux.sourceforge.net/configure.html

3. ***Überwachen des Switch-Verkehrs***:

Es ist auch möglich, den Switch-Verkehr zu überwachen. Dafür wird ein *Plugin* namens ***pdump*** zur Verfügung gestellt. Um von dieser Möglichkeit Gebrauch machen zu können, muss man unbedingt bei der Installation nach der Angabe *./configure* den Parameter *–enable- experimental* eingeben. Nach dem Start des VDE-Switches wird in seiner Management-Konsole zuerst das *Plugin* geladen und dann wie folgt aktiviert.

```
$ vde_switch -s /tmp/vde1 -M /tmp/mgmt1
$ ./linux ubd0=cow1.img,mini_vnuml-1.8.img
              con0=xterm eth1=daemon,,,/tmp/vde1/ctl &

(Virtuelle_Maschine_Login) : root
(Virtuelle_Maschine_Password) : xxxx
(VM-1)# ifconfig eth1 10.0.0.1 netmask 255.255.255.0 up


$ ./linux ubd0=cow2.img,mini_vnuml-1.8.img
              con0=xterm eth1=daemon,,,/tmp/vde1/ctl &

(Virtuelle_Maschine_Login) : root
(Virtuelle_Maschine_Password) : xxxx
(VM-2)# ifconfig eth1 10.0.0.2 netmask 255.255.255.0 up

$ unixterm /tmp/mgmt1

vde$ plugin/add /usr/local/lib/vde2/plugins/pdump.so
1000 Success

vde$ pdump/active 1
0000 DATA END WITH '.'
.
1000 Success

vde$
```

Infolgedessen wird eine Datei namens ***vde_dump.cap*** im aktuellen Directory erstellt. Diese Datei ähnelt einer Log-Datei, in der der ganze Switch-Verkehr aufgezeichnet wird. Mit einem Tool wie Wireshark ist es möglich, den Inhalt dieser Datei in einer grafischen Oberfläche auszugeben.

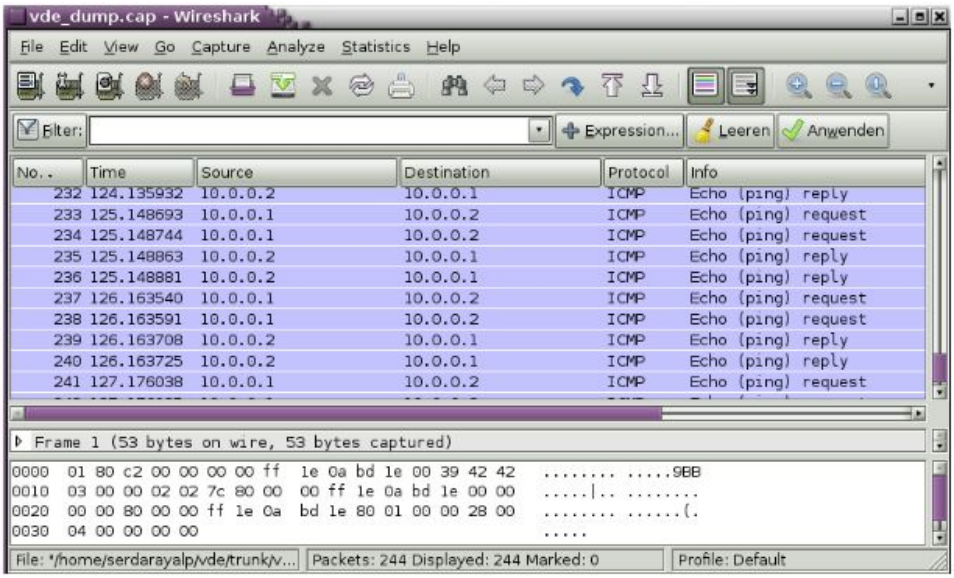

Abbildung 47: ARP-Request und ARP-Reply Pakete von VM-1 zu VM-2

6.4 VLANs und VDE

Serdar Ayalp

Virtuelles LAN oder VLAN ist eine Methode zur Erzeugung mehrerer logischer Netzwerke innerhalb eines physischen Netzes. Obwohl VDE kein reales Netz ist, ist es möglich, das virtuelle Netz in logische VLANs zu unterteilen. Hinzu kommt, dass es auch möglich ist, die Segmente eines VLANs, welche an unterschiedlichen VDE_Switches angeschlossen sind, miteinander zu verbinden. Diese Art der Verbindung der Netz-Segmente über mehrere Switches wird als ***VLAN-Trunking*** bezeichnet.

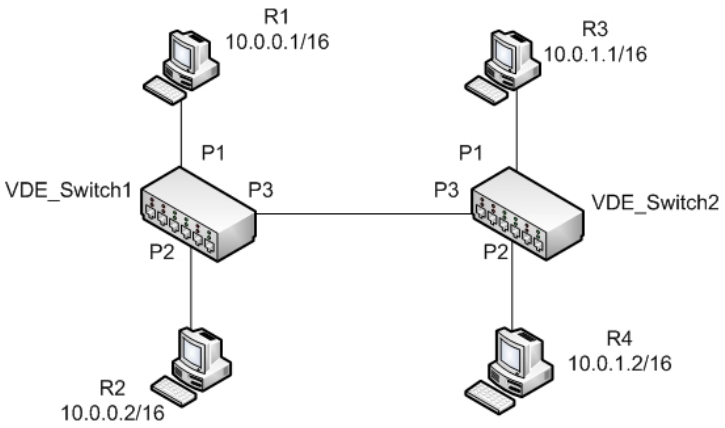

Abbildung 48: Beispiel-Netz zur VLAN-Erklärung

Jetzt werden wir mit Hilfe des obigen Beispiels die Einzelheiten von VLANS bei VDE Switches Schritt für Schritt erklären. Im Beispiel werden wir zwei VDE- Switches, die mit einem *dpipe* miteinander verbunden sind, erzeugen und vier virtuelle Rechner anschließen.

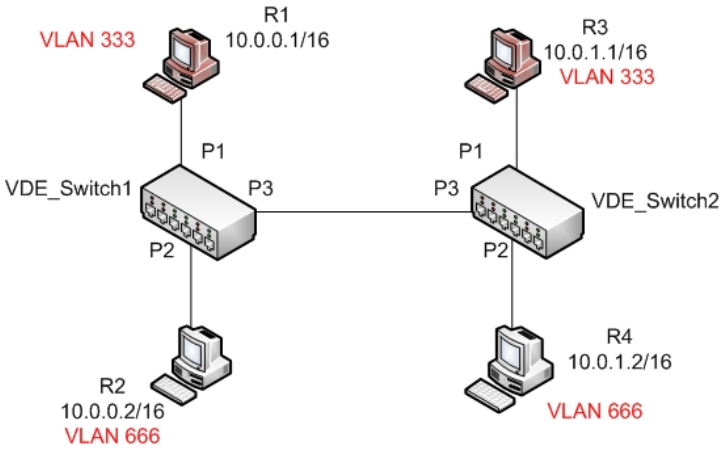

Abbildung 49: Ziel-Zustand zu unserem Beispiel

1. ***Erzeuge zwei VDE_Switches:***

```
~/vde/trunk/vde-2$ vde_switch -d -s /tmp/vde1 -M /tmp/mgmt1
~/vde/trunk/vde-2$ vde_switch -d -s /tmp/vde2 -M /tmp/mgmt2
```

2. ***Erzeuge vier virtuellen Maschine***:

```
~/vde/trunk/vde-2$ ./linux ubd0=cow1.img,mini_vnuml-1.8.img
                   con0=xterm eth1=daemon,,,/tmp/vde1/ctl &
~/vde/trunk/vde-2$ ./linux ubd0=cow2.img,mini_vnuml-1.8.img
                   con0=xterm eth1=daemon,,,/tmp/vde1/ctl &
~/vde/trunk/vde-2$ ./linux ubd0=cow3.img,mini_vnuml-1.8.img
                   con0=xterm eth1=daemon,,,/tmp/vde2/ctl &
~/vde/trunk/vde-2$ ./linux ubd0=cow4.img,mini_vnuml-1.8.img
                   con0=xterm eth1=daemon,,,/tmp/vde2/ctl &
```

3. ***Zwei VDE_Switches werden miteinander verbunden***:

```
~/vde/trunk/vde-2$ dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde2 &
[5] 7557
```

R1 und R3, gehören zu VLAN 333 und die anderen zwei zu VLAN 666. Die Portzustände in den VDE_Switches sehen wie folgt aus:

VDE_Switch 1:

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts         17          bytes                1386
 OUT: pkts         17          bytes                1386
  -- endpoint ID 0003 module unix prog   :
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts         10          bytes                 788
 OUT: pkts          6          bytes                 404
  -- endpoint ID 0008 module unix prog   :
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: serdarayalp Access Control: (User: NONE - Group: NONE)
 IN:  pkts          7          bytes                 598
 OUT: pkts          8          bytes                 640
  -- endpoint ID 0010 module unix prog   : vde_plug:
           user=serdarayalp PID=7557  SOCK=/tmp/vde1/.07557-00000
.
1000 Success

vde$
```

VDE_Switch 2:

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt2
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          9          bytes                   746
 OUT: pkts         11          bytes                   830
  -- endpoint ID 0003 module unix prog   :
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts         10          bytes                   788
 OUT: pkts          6          bytes                   404
  -- endpoint ID 0008 module unix prog   :
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: serdarayalp Access Control: (User: NONE - Group: NONE)
 IN:  pkts          8          bytes                   640
 OUT: pkts          7          bytes                   598
  -- endpoint ID 0010 module unix prog   : vde_plug:
             user=serdarayalp PID=7558  SOCK=/tmp/vde2/.07558-00000
.
1000 Success

vde$
```

In jedem Switch ist zu erkennen, dass der erste und der zweite Port an eine virtuelle Maschine angeschlossen sind. Der dritte Port wird für die Verbindung zwischen den beiden Switches verwendet (*dpipe*). Folgendermaßen lassen sich in jedem Switch zwei VLANs (*VLAN 333* und *VLAN 666*) erzeugen:

```
vde$ vlan/create 333
1000 Success

vde$ vlan/create 666
1000 Success

vde$
```

Nach der Erzeugung zweier VLANs sehen die VLAN-Zustände und die zugehörigen Ports so aus:

VDE_Switch 1:

```
vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0001 tagged=0 active=1 status=Forwarding
 -- Port 0002 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=0 active=1 status=Forwarding
VLAN 0333
VLAN 0666
.
1000 Success

vde$
```

und ***VDE_Switch 2***:

```
vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0001 tagged=0 active=1 status=Forwarding
 -- Port 0002 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=0 active=1 status=Forwarding
VLAN 0333
VLAN 0666
.
1000 Success
vde$
```

Wie in Abb. 49 zu sehen ist, sollen der erste und der zweite Port entsprechend an VLAN 333 und VLAN 666 zugewiesen werden:

VDE_Switch 1:

```
vde$ port/setvlan 1 333
1000 Success

vde$ port/setvlan 2 666
1000 Success

vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0003 tagged=0 active=1 status=Forwarding
VLAN 0333
 -- Port 0001 tagged=0 active=1 status=Forwarding
VLAN 0666
 -- Port 0002 tagged=0 active=1 status=Forwarding
.
1000 Success
vde$
```

und ***VDE_Switch 2***:

```
vde$ port/setvlan 1 333
1000 Success

vde$ port/setvlan 2 666
1000 Success

vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0003 tagged=0 active=1 status=Forwarding
VLAN 0333
 -- Port 0001 tagged=0 active=1 status=Forwarding
VLAN 0666
 -- Port 0002 tagged=0 active=1 status=Forwarding
.
1000 Success

vde$
```

Nun sollte eigentlich ein Ping vom R1 nach R3 klappen :

```
root@router:~# ping -c 4 10.0.1.1
PING 10.0.1.1 (10.0.1.1): 56 data bytes

--- 10.0.1.1 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
root@router:~#
```

Wie oben zu sehen, klappt ein Ping immer noch nicht, denn immer wenn ein Paket Switch 2 erreicht, kann nicht entschieden werden, zu welchem VLAN eigentlich dieses Paket gehören soll. Um dieses Problem zu lösen, wird jedem Paket beim Verlassen des jeweiligen Switches ein zusätzlicher *Header*, der die Zugehörigkeit des Paketes verdeutlichen soll (engl. ***tagging***), hinzugefügt. Infolgedessen ist Port 3 für jeden Switch, als *Trunking-Port*, wie folgt zu kennzeichnen.

VDE_Switch 1:

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ vlan/addport 333 3
1000 Success

vde$ vlan/addport 666 3
1000 Success
```

```
vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0003 tagged=0 active=1 status=Forwarding
VLAN 0333
 -- Port 0001 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=1 active=1 status=Forwarding
VLAN 0666
 -- Port 0002 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=1 active=1 status=Forwarding
.
1000 Success

vde$
```

und ***VDE_Switch 2***:

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt2
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ vlan/addport 333 3
1000 Success

vde$ vlan/addport 666 3
1000 Success

vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0003 tagged=0 active=1 status=Forwarding
VLAN 0333
 -- Port 0001 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=1 active=1 status=Forwarding
VLAN 0666
 -- Port 0002 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=1 active=1 status=Forwarding
.
1000 Success

vde$
```

Nun sollte, wie erwartet, R3 von R1 aus angepingt werden können.

```
root@router:~# ping -c 4 10.0.1.1
PING 10.0.1.1 (10.0.1.1): 56 data bytes
84 bytes from 10.0.1.1: icmp_seq=0 ttl=64 time=1.6 ms
84 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=0.6 ms
84 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=0.6 ms
84 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=0.5 ms
```

```
--- 10.0.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.8/1.6 ms
```

R2 sollte nicht erreichbar sein, weil sich R1 und R2 nicht im selben VLAN befinden.

```
root@router:~# ping -c 4 10.0.0.2
PING 10.0.0.2 (10.0.0.2): 56 data bytes

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
root@router:~#
```

Als nächstes werden wir unser Beispiel erweitern. Was würde passieren, wenn wir noch ein VLAN namens 999 erzeugen und R1 und R2 ihm zuweisen. Könnte man dann von R1 aus R2 anpingen? Unser Ziel-Netz sollte wie folgt aussehen.

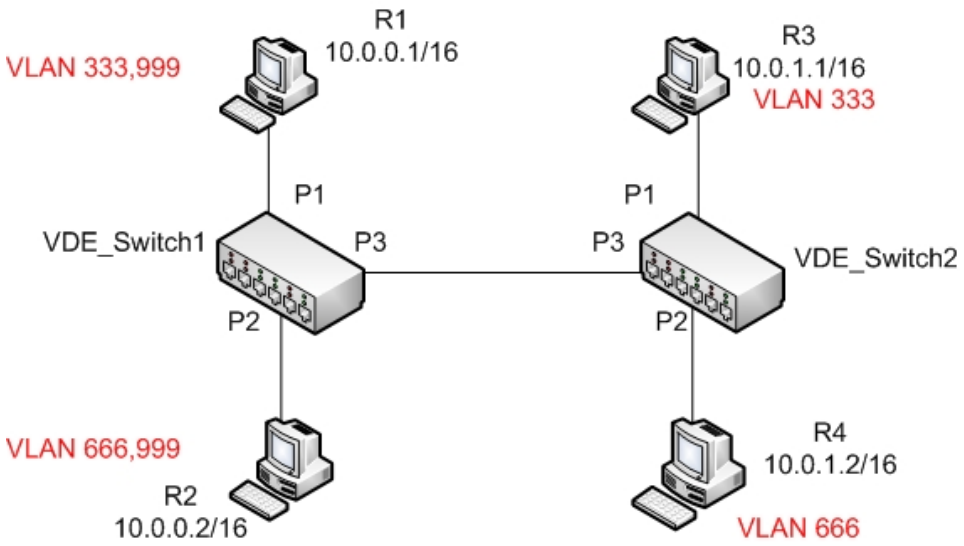

Abbildung 50: Neuer Ziel-Zustand für unser Beispiel

Darum loggen wir uns noch einmal beim VDE_Switch1 ein und erzeugen ein neues VLAN 999. Anschließend werden die Ports 1 und 2 dem VLAN 999 zugewiesen.

```
~vde/trunk/vde-2# unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2
```

```
vde$ vlan/create 999
1000 Success

vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
 -- Port 0003 tagged=0 active=1 status=Forwarding
VLAN 0333
 -- Port 0001 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=1 active=1 status=Forwarding
VLAN 0666
 -- Port 0002 tagged=0 active=1 status=Forwarding
 -- Port 0003 tagged=1 active=1 status=Forwarding
VLAN 0999
.
1000 Success

vde$

vde$ vlan/addport 999 1
1000 Success

vde$ vlan/addport 999 2
1000 Success


vde$ vlan/allprint
0000 DATA END WITH '.'
VLAN 0000
VLAN 0333
 -- Port 0001 tagged=0 active=1 status=Forwarding
VLAN 0666
 -- Port 0002 tagged=0 active=1 status=Forwarding
VLAN 0999
 -- Port 0001 tagged=1 active=1 status=Forwarding
 -- Port 0002 tagged=1 active=1 status=Forwarding
.
1000 Success

vde$
```

Würde ein Ping von R1 nun zu R2 gehen?

```
root@router:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2): 56 data bytes

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

```
root@router:~#
```

Ein Ping zu R2 erreicht das Ziel immer noch nicht. Der Grund dafür ist, dass zwischen virtuellen Maschinen und dem VDE_Switch eine *untagged*-Kommunikation stattfindet und darum VDE_Switch sich nicht entscheiden kann, zu welchem VLAN ein ankommendes Paket gehören und anschließend an welche Maschine es weitergeleitet werden soll. Im Gegensatz zu physikalischen Cisco-Switches unterstützt derzeit VDE_Switch die Zuweisung eines Ports zu verschiedenen VLANs leider nicht (engl. *multiport*). R1 erreicht R2 erst dann, wenn R1 und R2 als Router konfiguriert werden. Dafür werden in jeder virtuellen Maschine mit Hilfe der Linux-Module ***modprobe*** und ***vconfig*** ein virtuelles Interface erzeugt und mit dem Interface der virtuellen Maschine verbunden. Zu den neuen Interfaces (dargestellt *Name des Interfaces der VM.Name des VLANs*) sind auch entsprechende IP-Adressen zuzuordnen.

VM-1:

```
root@router:~# modprobe 8021q
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>

root@router:~# vconfig add eth1 999
root@router:~# ifconfig eth1.999 20.0.0.1
root@router:~#
```

VM-2:

```
root@router:~# modprobe 8021q
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>

root@router:~# vconfig add eth1 999
root@router:~# ifconfig eth1.999 20.0.0.2
root@router:~#
```

Nun klappt ein Ping zu R2:

VM-1:

```
root@router:~# ping -c 4 20.0.0.2
PING 20.0.0.2 (20.0.0.2): 56 data bytes
84 bytes from 20.0.0.2: icmp_seq=0 ttl=64 time=0.4 ms
84 bytes from 20.0.0.2: icmp_seq=1 ttl=64 time=0.4 ms
84 bytes from 20.0.0.2: icmp_seq=2 ttl=64 time=0.4 ms
84 bytes from 20.0.0.2: icmp_seq=3 ttl=64 time=0.4 ms

--- 20.0.0.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.4/0.4 ms
```

6.5 FSTP und VDE

Serdar Ayalp

FSTP wurde in das VDE-Projekt eingebracht, um Loops zu vermeiden und redundante Links in Netzen zu bewerkstelligen. Dieses Protokoll ist für die Rekonfiguration der Netzwerk-Topologie zuständig, wenn ein Link oder ein Switch im Netz ausfallen sollte. VDE_Switch bietet mehrere Befehle zum Management und zum Monitoring des FSTP-Status und seiner Parameter an. Für die Erklärung des *Fast Spanning Tree Protokolls* in VDE werden wir das Beispiel von der VDE-Homepage [8] nutzen.

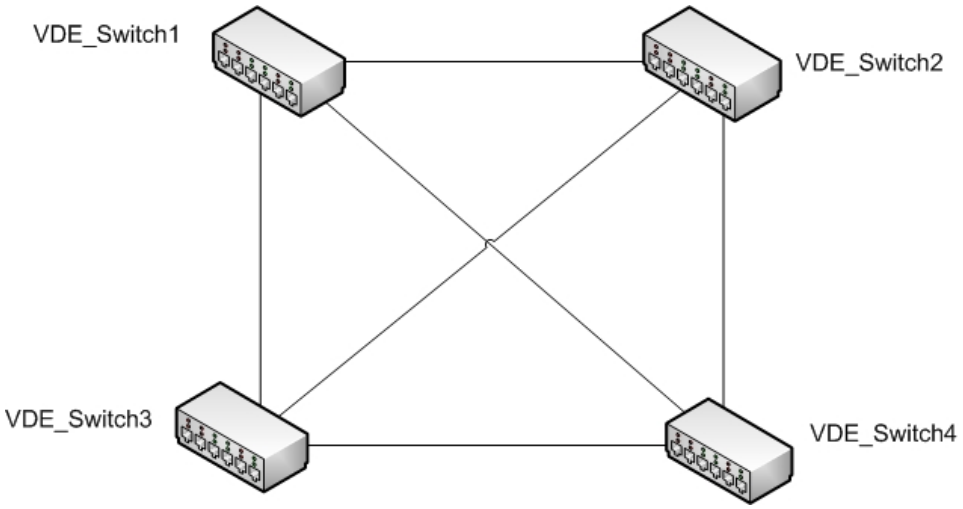

Abbildung 51: Beispiel Topologie für FSTP

In der Abb. 51 wird jeder der vier VDE_Switches,durch einen Link, miteinander verbunden. Wie zu sehen, gibt es viele Loops und redundante Links. Die Aufgabe des FSTPs ist die Berechnung eines *Span*-Baums und die Blockierung der Links, die nicht Teil des Baums sind.
Um dieses Netz aufzubauen, erstellen wir vier VDE_Switches und verbinden sie wie folgt miteinander:

```
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde1 -M /tmp/mgmt1
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde2 -M /tmp/mgmt2
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde3 -M /tmp/mgmt3
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde4 -M /tmp/mgmt4
~/vde/trunk/vde-2#

~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde2 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde3 &
```

```
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde4 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde2 = vde_plug /tmp/vde3 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde2 = vde_plug /tmp/vde4 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde3 = vde_plug /tmp/vde4 &
~/vde/trunk/vde-2#
```

Standardmäßig ist FSTP auf keinem Switch aktiviert und darum sind alle erstellten Links und die zugehörigen Ports aktiv. FSTP-Informationen sind auf jedem Switch mit Hilfe des Befehls *fstp/print* anzuzeigen. Wie in den Ausgaben unten zu sehen, denkt jeder Switch am Anfang, dass er selbst der Root-Switch ist.

```
~/vde/trunk/vde-2# unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH FSTP IS DISABLED
 ++ root 80:00:00:ff:2a:04:e8:55
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 539 bonusport 0000 bonuscost 0
.
1000 Success

vde$
```

```
~/vde/trunk/vde-2# unixterm /tmp/mgmt2
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH FSTP IS DISABLED
 ++ root 80:00:00:ff:0f:c0:8d:08
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 599 bonusport 0000 bonuscost 0
.
1000 Success

vde$
```

```
~/vde/trunk/vde-2# unixterm /tmp/mgmt3
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
```

```
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH FSTP IS DISABLED
 ++ root 80:00:00:ff:4f:d8:c1:09
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 646 bonusport 0000 bonuscost 0
.
1000 Success

vde$


~/vde/trunk/vde-2# unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH FSTP IS DISABLED
 ++ root 80:00:00:ff:5d:ab:ea:49
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 678 bonusport 0000 bonuscost 0
.
1000 Success

vde
```

Nach der Einschaltung des STP-Protokolls auf jedem Switch werden die Schritte, die im Kapitel *Spanning Tree Protokoll* eingeführt worden sind, verfolgt [28]. In unserem Beispiel wurde der VDE_Switch 3 zum Root-Switch und alle seine Ports wurden zu *Designated Ports*. Diese geben Daten an die anderen Netz-Segmente weiter. Auf jedem Switch wurden alle direkt an den Root-Switch verbundenen Ports und die mit den kleinsten Kosten zum Root-Switch, zu Root-Ports. Alle anderen Ports sind als Alternate/Backup (*Blocking*-Port in STP) gekennzeichnet, bis zu einer Änderung der Topologie bzw. zum Ausfall eines Switches oder Links. Der FSTP-Status auf jedem Switch und der Zustand der Topologie sehen nach dem Einschalten des FSTPs wie folgt aus.

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated 80:00:00:ff:16:e0:8d:07
```

[28](siehe Kapitel 5.3)

```
 ++ rootport 0002 cost 20000000 age 1 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$

~/vde/trunk/vde-2$ unixterm /tmp/mgmt2
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated 80:00:00:ff:16:e0:8d:07
 ++ rootport 0002 cost 20000000 age 1 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

~/vde/trunk/vde-2$ unixterm /tmp/mgmt3
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 173 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Designated
 -- Port 0003 tagged=0 portcost=20000000 role=Designated
.
1000 Success

vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
```

```
FST DATA VLAN 0000
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated 80:00:00:ff:16:e0:8d:07
 ++ rootport 0002 cost 20000000 age 2 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$
```

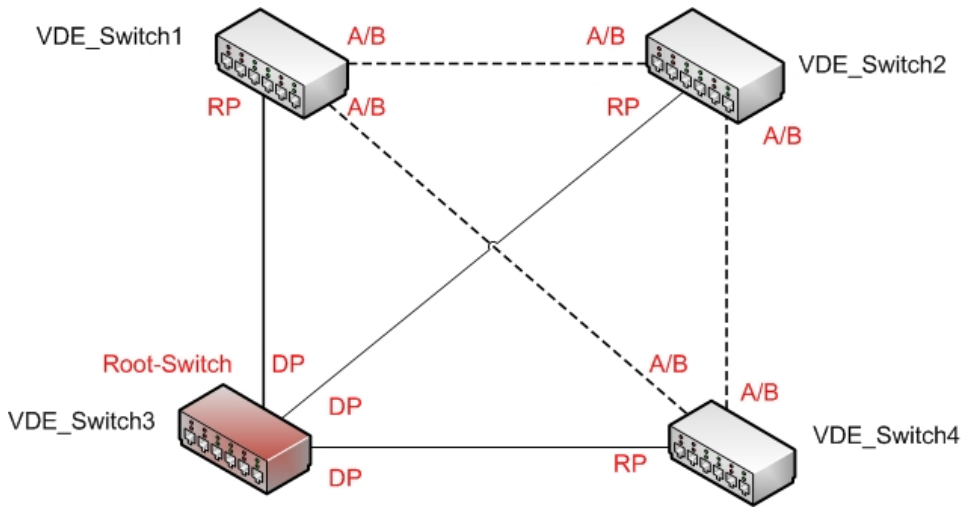

Abbildung 52: Zustand nach dem Einschalten des FSTPs.

Es ist auch möglich einen Link-Ausfall bzw. eine Link-Störung zu simulieren, indem man einen Link auswählt und die mit ihm verbundenen VDE_Plugs abschaltet bzw. den passenden Prozess killt. Was würde z.B erfolgen, wenn wir den Link zwischen VDE_Switch 3 und 4 abschalten würden. Dafür sind erst die passenden VDE_Plugs auf den jeweiligen Switches zu identifizieren. Weil der Root-Port den VDE_Switch 4 mit VDE_Switch 3 verbindet, ist es einfach den passenden VDE_Plug zu finden und anschließend den Link lahm zu legen. Loggen wir uns in Switch 4 ein und suchen die Prozess-ID des VDE_Plugs.

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2


vde$ port/allprint
0000 DATA END WITH '.'
```

```
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: serdarayalp Access Control: (User: NONE - Group: NONE)
 IN:  pkts        106          bytes                5618
 OUT: pkts        104          bytes                5512
  -- endpoint ID 0003 module unix prog   : vde_plug: user=serdarayalp
                               PID=9237  SOCK=/tmp/vde4/.09237-00000
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: serdarayalp Access Control: (User: NONE - Group: NONE)
 IN:  pkts      26806          bytes              1420718
 OUT: pkts        104          bytes                5512
  -- endpoint ID 0008 module unix prog   : vde_plug: user=serdarayalp
                              PID=9239  SOCK=/tmp/vde4/.09239-00000
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: serdarayalp Access Control: (User: NONE - Group: NONE)
 IN:  pkts        101          bytes                5353
 OUT: pkts      26806          bytes              1420718
  -- endpoint ID 0010 module unix prog   : vde_plug: user=serdarayalp
                              PID=9233  SOCK=/tmp/vde4/.09233-00000
.
1000 Success

vde$
```

In diesem Fall ist es die Nummer 9239 (*PID=9239*). Um den Link abzuschalten, reicht es aus, dass man einen Plug des jeweligen Links abschaltet. Durch einen Linux-Befehl ***kill*** lässt sich dies wie folgt durchführen.

```
~/vde/trunk/vde-2$ kill -9 9239
```

Nach dem Ausfall des Links wird die Topologie per FSTP noch einmal konfiguriert. Die Port-Zustände sind:

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated 80:00:00:ff:16:e0:8d:07
 ++ rootport 0002 cost 20000000 age 1 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Designated
.
1000 Success
```

```
vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt2
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated 80:00:00:ff:16:e0:8d:07
 ++ rootport 0002 cost 20000000 age 0 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt3
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 514 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Designated
.
1000 Success

vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:16:e0:8d:07
 ++ designated 80:00:00:ff:89:07:f8:63
```

```
 ++ rootport 0003 cost 40000000 age 4 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0003 tagged=0 portcost=20000000 role=Root
.
1000 Success

vde$
```

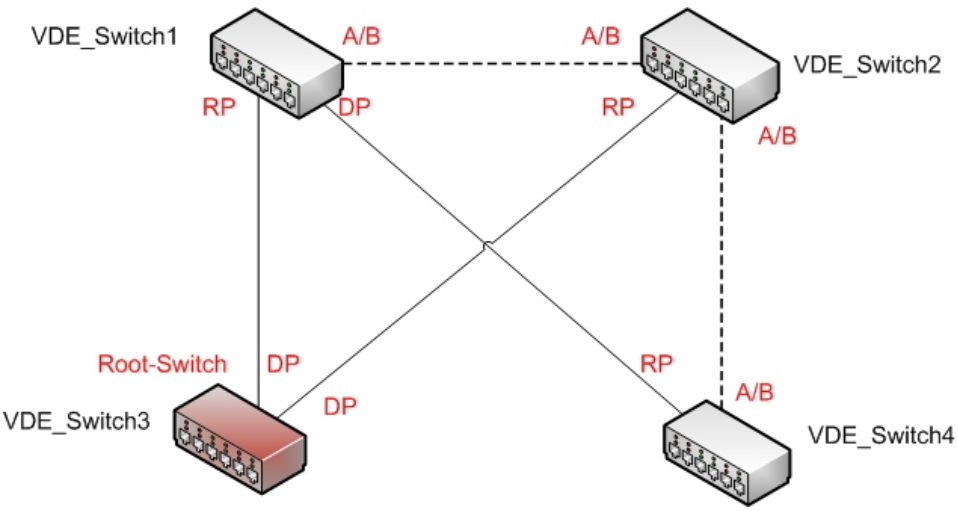

Abbildung 53: Topologie nach Abschaltung der Verbindung zwischen Switch 3 und 4

Standardmäßig hat jeder Port eines Switches die selben Kosten(in VDE, 20M=20000000). Aber es ist auch möglich (für jedes VLAN) einem Port, genannt *Bonus-Port*, *Bonus-Kosten* (engl. bonus cost) zuzuordnen. Ein Bonus-Port ist anders als die normalen Ports, weil von seinen Standard-Kosten der Betrag der Bonuskosten, welche vom User eingegeben wurden, subtrahiert wird. Wir werden wieder das selbe Beispiel verwenden, um die Port-Kosten zu manipulieren und zu kontrollieren, ob die Topologie dem entsprechend konfiguriert werden kann. Wir müssen aber das ganze Szenario von Anfang an starten, weil die Manupilation der Port-Kosten vor dem Einschalten des FSTPs gemacht werden muss. Dafür sind wieder vier VDE_Switches und ihre Verbindungen, wie oben beschrieben, zu erstellen (siehe 6.5)- ohne FSTP-Unterstützung. Bevor das FSTP eingeschaltet wird, sind für jeden VDE_Switch der Bonus-Port und die Bonus-Kosten zu definieren. Damit ist man in der Lage, die Topologie nach eigenen Wüschen zu ändern bzw. zu rekonfigurieren. Es ist auch zu beachten, dass VDE nach jedem Neustart, die Switch-Zustände ändern kann(z.B. Root-Switch). Es ist leider derzeit in VDE schwierig eigene Konfigurationen, wie z.B. die Bestimmung des Root-Switches, vorzunehmen. In unserem Fall wird Switch 2 zum Root-Switch.

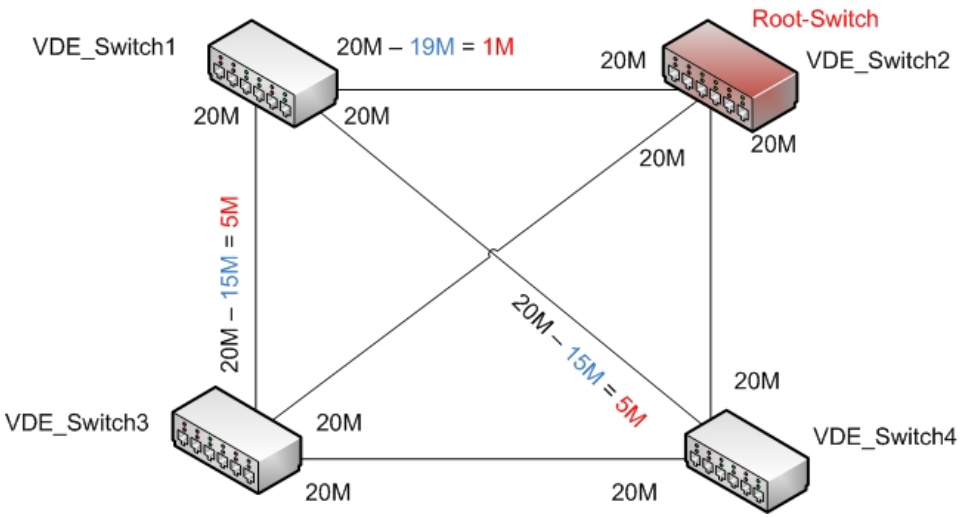

Abbildung 54: Manipulation der Port-Kosten

Die blauen Nummern in der Abbildung stellen die Bonus-Kosten für den jeweiligen Port und die roten Nummern die tatsächlichen Port-Kosten da.
Konfigurationsschritte:

1. VDE_Switches erzeugen und miteinander verbinden.

```
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde1 -M /tmp/mgmt1
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde2 -M /tmp/mgmt2
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde3 -M /tmp/mgmt3
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde4 -M /tmp/mgmt4
~/vde/trunk/vde-2#

~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde2 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde3 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde4 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde2 = vde_plug /tmp/vde3 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde2 = vde_plug /tmp/vde4 &
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde3 = vde_plug /tmp/vde4 &
~/vde/trunk/vde-2#
```

2. In die VDE_Switches 1, 3, und 4 einloggen und den Bonus-Port und die Bonus-Kosten bestimmten.
   Syntax: ***fstp/bonus*** *VLAN Port-Nummer Bonus-Kosten.*

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2
```

```
vde$ fstp/bonus 0 1 19000000
1000 Success

vde$

~/vde/trunk/vde-2$ unixterm /tmp/mgmt3
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/bonus 0 1 15000000
1000 Success

vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/bonus 0 1 15000000
1000 Success

vde$
```

3. Bei jedem Switch einloggen und FSTP aktivieren, wie z.B. :

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/setfstp 1
0000 DATA END WITH '.'
.
1000 Success

vde$
```

4. Verifizieren der Port-Zustände.

```
~/vde/trunk/vde-2$ unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:31:0f:1c:43
 ++ designated 80:00:00:ff:31:0f:1c:43
 ++ rootport 0001 cost 1000000 age 2 bonusport 0001 bonuscost 19000000
 -- Port 0001 tagged=0 portcost=20000000 role=Root
```

```
 -- Port 0002 tagged=0 portcost=20000000 role=Designated
 -- Port 0003 tagged=0 portcost=20000000 role=Designated
.
1000 Success

vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt2
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH
 ++ root 80:00:00:ff:31:0f:1c:43
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 1016 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt3
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:31:0f:1c:43
 ++ designated 80:00:00:ff:f2:44:77:6f
 ++ rootport 0001 cost 6000000 age 0 bonusport 0001 bonuscost 15000000
 -- Port 0001 tagged=0 portcost=20000000 role=Root
 -- Port 0002 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$


~/vde/trunk/vde-2$ unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2
```

```
vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:31:0f:1c:43
 ++ designated 80:00:00:ff:f2:44:77:6f
 ++ rootport 0001 cost 6000000 age 3 bonusport 0001 bonuscost 15000000
 -- Port 0001 tagged=0 portcost=20000000 role=Root
 -- Port 0002 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$
```

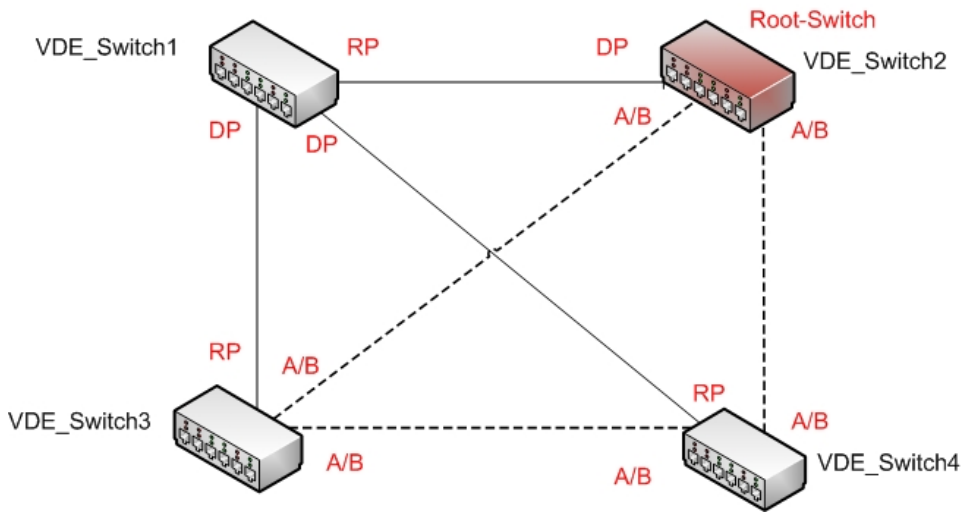

Abbildung 55: Die Wege mit den niedrigsten Kosten werden bevorzugt

Wie in der Abbildung 55 zu sehen ist, wurden nicht die jenigen Ports zu Root-Ports, die den Non-Root-Switch direkt mit dem Root-Switch verbinden, sondern die jenigen, durch die ein Paket den Root-Switch mit minimalen Kosten erreicht. In diesem Fall nimmt ein Paket, das den VDE_Switch 4 verlässt, nicht den direkten Weg (Kosten= 20M) sondern den Weg über VDE_Switch 1 (Kosten = 5M + 1M = 6M).

6.6 Verbindung mit einem physikalischen Switch einrichten

Serdar Ayalp

Ein physikalischer Switch lässt sich mit einem virtuellen Switch verbinden. In der Abb 56 sind zwei VDE_Switches zu sehen, die durch eine virtuelle Bridge mit einem physikalischen Switch verbunden sind. Es ist auch zu beachten, dass sich zwischen jedem VDE_Switch und der virtuellen Bridge ein virtuelles Tun-Tap-Interface befindet. Diese Interfaces sind vor dem Starten jedes VDE_Switches zu erstellen und danach direkt dem VDE_Switch anzubinden.

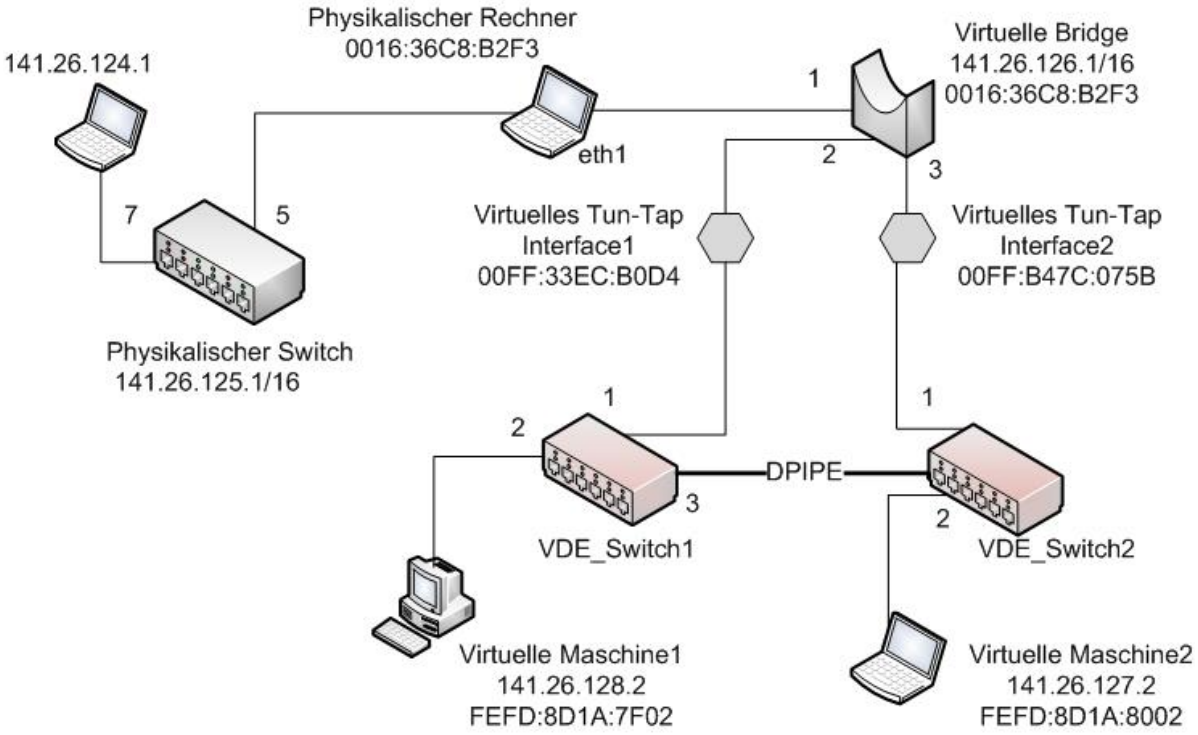

Abbildung 56: Verbindung mit einem physikalischen Switch einrichten

Im Folgenden werden die für die Verbindung notwendigen Schritte ausführlich beschrieben.

1. Zuerst konfigurieren wir die IP-Adresse unseres Rechners auf 0.0.0.0 (In diesem Fall Interface *eth1*). ***Standardmäßig müssen alle zu einer virtuellen Bridge***

anzubindenden Geräte eine IP von 0.0.0.0 haben. Hinzu kommt, dass als Empfangsmodus der *Promiscuous Mode* zu verwenden ist[29]

```
~/vde/trunk/vde-2# ifconfig eth1 0.0.0.0 promisc up
```

2. Erstelle eine virtuelle Bridge namens MeineBridge:

```
~/vde/trunk/vde-2# brctl addbr MeineBridge
```

3. Die Verweildauer in den *listening*- und *learning*-Zuständen werden auf 0 gesetzt.

```
~/vde/trunk/vde-2# brctl setfd MeineBridge 0
```

4. Bestimmumg, wie oft die *Hello-Pakete* auszusenden sind.

```
~/vde/trunk/vde-2# brctl sethello MeineBridge 0
```

5. STP wird eingeschaltet bzw. ausgeschaltet.

```
~/vde/trunk/vde-2# brctl stp MeineBridge off
```

6. Der Bridge wird eine IP und eine Netzwerkmaske zugewiesen.

```
~/vde/trunk/vde-2# ifconfig MeineBridge 141.26.126.1
                        netmask 255.255.0.0 up
```

7. Interface des physikalischen Rechners(*eth1*) mit der Bridge verbinden.

```
~/vde/trunk/vde-2# brctl addif MeineBridge eth1
```

8. Für jeden VDE_Switch ein Tun-Tap-Interface erstellen.

```
~/vde/trunk/vde-2# tunctl -u root -t MeinTunTap1
Set 'MeinTunTap1' persistent and owned by uid 0
~/vde/trunk/vde-2# tunctl -u root -t MeinTunTap2
Set 'MeinTunTap2' persistent and owned by uid 0
```

9. Weil Tun-Taps auch mit der Bridge zu verbinden sind, müssen auch sie eine IP von 0.0.0.0 haben.

```
~/vde/trunk/vde-2# ifconfig MeinTunTap1 0.0.0.0 promisc up
~/vde/trunk/vde-2# ifconfig MeinTunTap2 0.0.0.0 promisc up
~/vde/trunk/vde-2# brctl addif MeineBridge MeinTunTap1
~/vde/trunk/vde-2# brctl addif MeineBridge MeinTunTap2
```

[29] **Zitat** : Der Promiscuous Mode bezeichnet einen bestimmten Empfangsmodus für netzwerktechnische Geräte. In diesem Modus liest das Gerät den gesamten ankommenden Datenverkehr an die in diesen Modus geschaltete Netzwerkschnittstelle mit und gibt die Daten zur Verarbeitung an das Betriebssystem weiter. Geräte, die diesen Modus benutzen, können Kombinationen aus Switch und Router, Netzwerktester oder auch normale Computer sein.[13]

10. VDE_Switches erstellen und mit Tun-Taps verbinden.

```
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde1
                    -M /tmp/mgmt1 -t MeinTunTap1
~/vde/trunk/vde-2# vde_switch -d -s /tmp/vde2
                    -M /tmp/mgmt2 -t MeinTunTap2
~/vde/trunk/vde-2#
```

11. Als Letztes sind zwei virtuelle Maschinen zu erstellen und mit den VDE_Switches zu verbinden. Mit dem *route*-Befehl in jeder virtuellen Maschine wird der sämtliche Datenverkehr zur Bridge weitergeleitet.

```
~/vde/trunk/vde-2#./linux ubd0=cow1.img,mini_vnuml-1.8.img
        con0=xterm eth1=daemon,,,/tmp/vde1/ctl &

(VM-1)# ifconfig eth1 141.26.128.2 netmask 255.255.0.0 up
(VM-1)# route add default gw 141.26.126.1


~/vde/trunk/vde-2#./linux ubd0=cow2.img,mini_vnuml-1.8.img
con0=xterm eth1=daemon,,,/tmp/vde2/ctl &

(VM-2)# ifconfig eth1 141.26.127.2 netmask 255.255.0.0 up
(VM-2)# route add default gw 141.26.126.1
```

Nun kann man von der VM1 aus die VM2 und die Bridge anpingen.

```
root@router:~# ping -c 4 141.26.126.1
PING 141.26.126.1 (141.26.126.1): 56 data bytes
84 bytes from 141.26.126.1: icmp_seq=0 ttl=64 time=0.2 ms
84 bytes from 141.26.126.1: icmp_seq=1 ttl=64 time=0.2 ms
84 bytes from 141.26.126.1: icmp_seq=2 ttl=64 time=0.2 ms
84 bytes from 141.26.126.1: icmp_seq=3 ttl=64 time=0.2 ms

--- 141.26.126.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
root@router:~#


root@router:~# ping -c 4 141.26.127.2
PING 141.26.127.2 (141.26.127.2): 56 data bytes
84 bytes from 141.26.127.2: icmp_seq=0 ttl=64 time=0.2 ms
84 bytes from 141.26.127.2: icmp_seq=1 ttl=64 time=0.2 ms
84 bytes from 141.26.127.2: icmp_seq=2 ttl=64 time=0.2 ms
84 bytes from 141.26.127.2: icmp_seq=3 ttl=64 time=0.2 ms

--- 141.26.127.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0.2/0.2/0.2 ms
root@router:~#
```

Was würde passieren, wenn wir die beiden VDE_Switches direkt miteinander verbinden und die VM2 wieder anpingen.

```
~/vde/trunk/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde2 &
[3] 7562

root@router:~# ping -c 4 141.26.127.2
PING 141.26.127.2 (141.26.127.2): 56 data bytes
84 bytes from 141.26.127.2: icmp_seq=0 ttl=64 time=0.8 ms
84 bytes from 141.26.127.2: icmp_seq=0 ttl=64 time=1.0 ms (DUP!)
84 bytes from 141.26.127.2: icmp_seq=0 ttl=64 time=1.0 ms (DUP!)
84 bytes from 141.26.127.2: icmp_seq=1 ttl=64 time=0.2 ms
84 bytes from 141.26.127.2: icmp_seq=2 ttl=64 time=0.2 ms
84 bytes from 141.26.127.2: icmp_seq=3 ttl=64 time=0.2 ms

--- 141.26.127.2 ping statistics ---
4 packets transmitted, 4 packets received, 2 duplicates, 0% packet loss
round-trip min/avg/max = 0.2/0.5/1.0 ms
```

Wie man sieht, kommen Pakete doppelt an. Weil wir im fünften Schritt (siehe 5) das STP abgeschaltet haben, kommen nach einer Weile keine Pakete mehr von der VM1 zur VM2.

```
root@router:~# ping -c 4 141.26.127.2
PING 141.26.127.2 (141.26.127.2): 56 data bytes

--- 141.26.127.2 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

Ein Ping geht erst dann wieder, wenn man auf der Bridge STP aktiviert:

```
~/vde/trunk/vde-2# brctl stp MeineBridge on
~/vde/trunk/vde-2#

root@router:~# ping -c 4 141.26.127.2
PING 141.26.127.2 (141.26.127.2): 56 data bytes
84 bytes from 141.26.127.2: icmp_seq=0 ttl=64 time=0.7 ms
84 bytes from 141.26.127.2: icmp_seq=1 ttl=64 time=0.3 ms
84 bytes from 141.26.127.2: icmp_seq=2 ttl=64 time=0.3 ms
84 bytes from 141.26.127.2: icmp_seq=3 ttl=64 time=0.3 ms

--- 141.26.127.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.4/0.7 ms
root@router:~#
```

Es ist auch in der MAC-Tabelle der Bridge zu sehen, dass Port 3 gesperrt wurde. Die Bridge erreicht die VM2 nicht über Port 3, sondern über Port2.

```
~/vde/trunk/vde-2# brctl showmacs MeineBridge
portno     mac addr              is local?     ageing timer
  1        00:16:36:c8:b2:f3     yes           0.00
  2        00:ff:33:ec:b0:d4     yes           0.00
  3        00:ff:b4:7c:07:5b     yes           0.00
  2        fe:fd:8d:1a:7f:02     no            37.81
  2        fe:fd:8d:1a:80:02     no            88.96
~/vde/trunk/vde-2#
```

Jetzt verbinden wir unseren physikalischen Rechner mit einem physikalischen Switch. Zusätzlich ist dem physikalischen Switch ein anderer Rechner mit der IP 141.26.124.1 angeschlossen. Ein Ping von der VM1 zum physikalischen Switch und Rechner sollte jetzt gehen:

```
root@router:~# ping -c 4 141.26.125.1
PING 141.26.125.1 (141.26.125.1): 56 data bytes
84 bytes from 141.26.125.1: icmp_seq=0 ttl=255 time=7.1 ms
84 bytes from 141.26.125.1: icmp_seq=1 ttl=255 time=21.4 ms
84 bytes from 141.26.125.1: icmp_seq=2 ttl=255 time=27.9 ms
84 bytes from 141.26.125.1: icmp_seq=3 ttl=255 time=36.4 ms

--- 141.26.125.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 7.1/23.2/36.4 ms
root@router:~#


root@router:~# ping -c 4 141.26.124.1
PING 141.26.124.1 (141.26.124.1): 56 data bytes
84 bytes from 141.26.124.1: icmp_seq=0 ttl=255 time=11.6 ms
84 bytes from 141.26.124.1: icmp_seq=1 ttl=255 time=15.8 ms
84 bytes from 141.26.124.1: icmp_seq=2 ttl=255 time=11.8 ms
84 bytes from 141.26.124.1: icmp_seq=3 ttl=255 time=12.2 ms

--- 141.26.124.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 11.6/12.8/15.8 ms
root@router:~#
```

Der letzte Zustand der Bridge MAC-Tabelle:

```
~/vde/trunk/vde-2# brctl showmacs MeineBridge
portno   mac addr             is local?  ageing timer
  1      00:05:dd:3e:ce:80    no         164.83
```

```
  1       00:05:dd:3e:ce:85    no          0.34
  1       00:06:53:77:0f:40    no          72.65
  1       00:16:36:c8:b2:f3    yes         0.00
  2       00:ff:33:ec:b0:d4    yes         0.00
  3       00:ff:b4:7c:07:5b    yes         0.00
  2       fe:fd:8d:1a:7f:02    no          185.41
  2       fe:fd:8d:1a:80:02    no          72.65
~/vde/trunk/vde-2#
```

7 Fazit

Suat Algin

Durch die Entwicklung der Switchtechnologie, ist man in der Lage, Aufgaben performanter und effizienter zu erledigen. Wie im ersten Teil der Diplom-Arbeit zu sehen ist, kann der Administrator sich die Arbeit durch die Clusterisierung seines Netzwerks sehr erleichtern. Einerseits ist es ihm möglich, Fehlerfälle direkt anhand der grafischen Darstellung zu erkennen und zu beheben, andererseits aber auch Fehlerquellen im vorhinein festzustellen, um vorbeugende Maßnahmen zu treffen. Auch die inzwischen etablierte Methode der Aufteilung von Netzwerken in virtuelle Teilnetze ist ein essenzielles Mittel des Administrators. Durch solch eine VLAN-Gruppierung kann man mit ganz einfachen Mitteln, den Zugriff auf bestimmte Teilnetze limitieren oder erweitern. Besonders Erwähnenswert sind in diesem Zusammenhang Multi-VLANs. Diese Art der Virtualisierung erlaubt dem Netzteilnehmer auf zwei oder mehr VLANs gleichzeitig zuzugreifen. Hierdurch können bestimmten Clients mehr Rechte zugeordnet werden, um eine größere Erreichbarkeit zu erreichen. Diese Zuordnung wird meist in Unternehmen und größeren Netzwerken(Uni-Netzwerk) genutzt, um bestimmte Gruppen oder Abteilungen von anderen abzuheben. In diesem Zusammenhang ist natürlich auch das Spanning Tree Protokoll, welches sich in den letzten Jahren sehr weiterentwickelt hat, zu erwähnen. Durch die Einführung des Rapid Spanning Tree Protokolls, welches auch Fast Spanning Tree Protokoll genannt wird, hat mein nun eine realisierbare Technik, welche redundante Pfade in einem Netzwerk zwar aufrecht erhält, aber gleichzeitig auch Loops im Netz verhindert. Diese Methode gleicht dem ursprünglichen Spanning Tree Protokoll, bis auf die Ausnahme , dass durch eine schnellere Abarbeitung der STP-Phasen, eine schnellere Erreichbarkeit ausgefallener Teilnetze ermöglicht wird. Hier ist aber zu beachten, dass die Option "PORT-FAST"des Catalyst 3500XL nicht gleichzusetzen ist, mit dem RSTP bzw. dem FSTP. Wie im letzten Teil der Diplom-Arbeit ausführlich behandelt wurde, entwickelt Virtual Square ein sehr nützliches Tool namens Virtual Distributed Ethernet. Dieses Tool ermöglicht jedem Linux-Nutzer durch recht einfache Mittel virtuelle Netzwerke auf seinem Rechner zu erzeugen. Diese Netzwerke können dann reale Szenerien nachstellen und so Fehlerquellen oder Netzwerkprobleme verdeutlichen. So ist es möglich, ohne den Einsatz von teurer Hardware, Netzwerke zu simulieren und Administrierungsaufgaben nachzuempfinden. Nichts desto trotz muss aber auch erwähnt werden, dass die Technik in einigen Aspekten noch nicht sehr ausgereift ist und so eine Weiterentwicklung voraussetzt. Diese Aspekte sind zum Beispiel das einrichten von Multi-VLANs auf mehr als einem VDE-Switch und in diesem Zusammenhang, die Erreichbarkeit von Netzwerkteilnehmern, welche Pakete üngetagged"über einen Inter-Switch-Link senden. Trotz dieser Probleme ist VDE, für jeden der sich für die Administration von kleinen und größeren Netzwerken interessiert, absolut empfehlenswert. Dadurch, dass VDE frei zugänglich ist und von jedem weiterentwickelt werden kann, ist davon auszugehen, dass diese fehlerhaften Eigenschaften in den nächsten Jahren behoben werden.

8 Aufgaben

Serdar Ayalp, Suat Algin

1. **Aufgabe**: Erläutern Sie das Prinzip eines Clusters!
2. **Aufgabe**: Erklären Sie den Unterschied zwischen Single-VLAN Membership und Multi-VLAN Membership!
3. **Aufgabe**: Konfigurieren Sie den Cisco Catalyst 3500 XL so, dass folgendes Netz erzeugt wird!

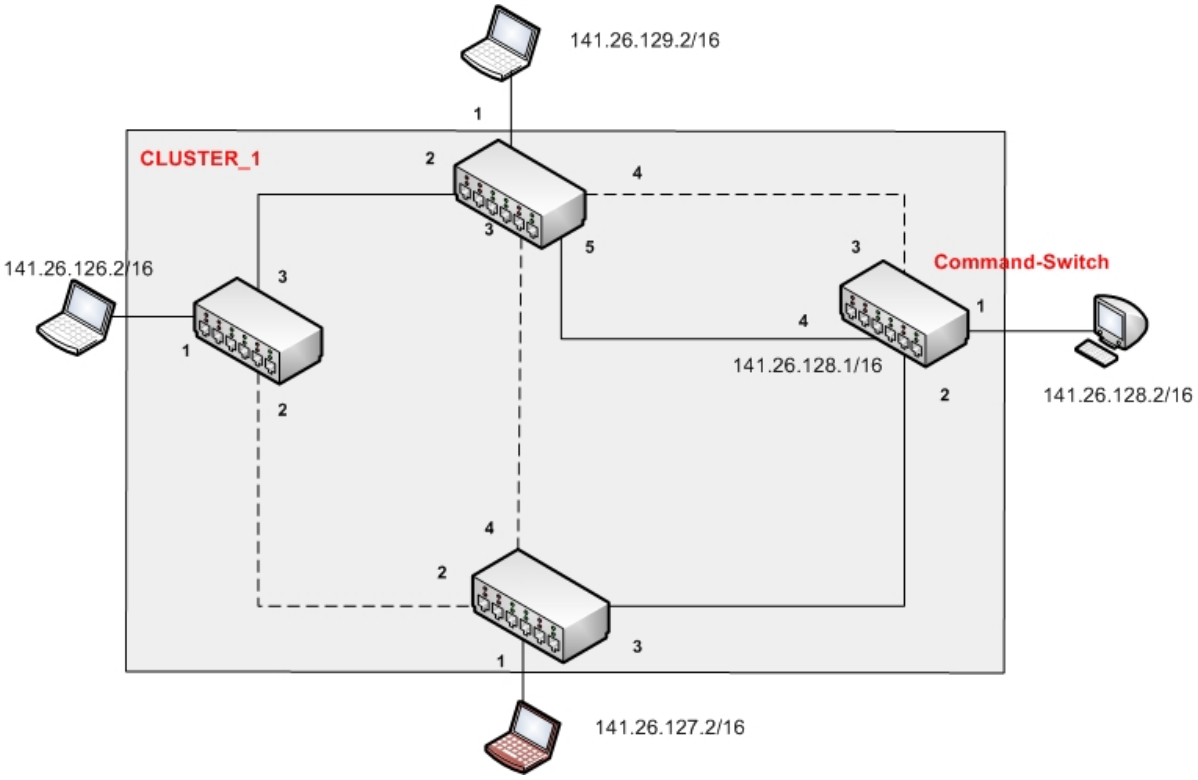

Abbildung 57: Netzwerk-Topologie(Cisco)

4. **Aufgabe**: Erstellen Sie das folgende Netz mit VDE und achten Sie darauf, dass die Port-Nummern übereinstimmen.

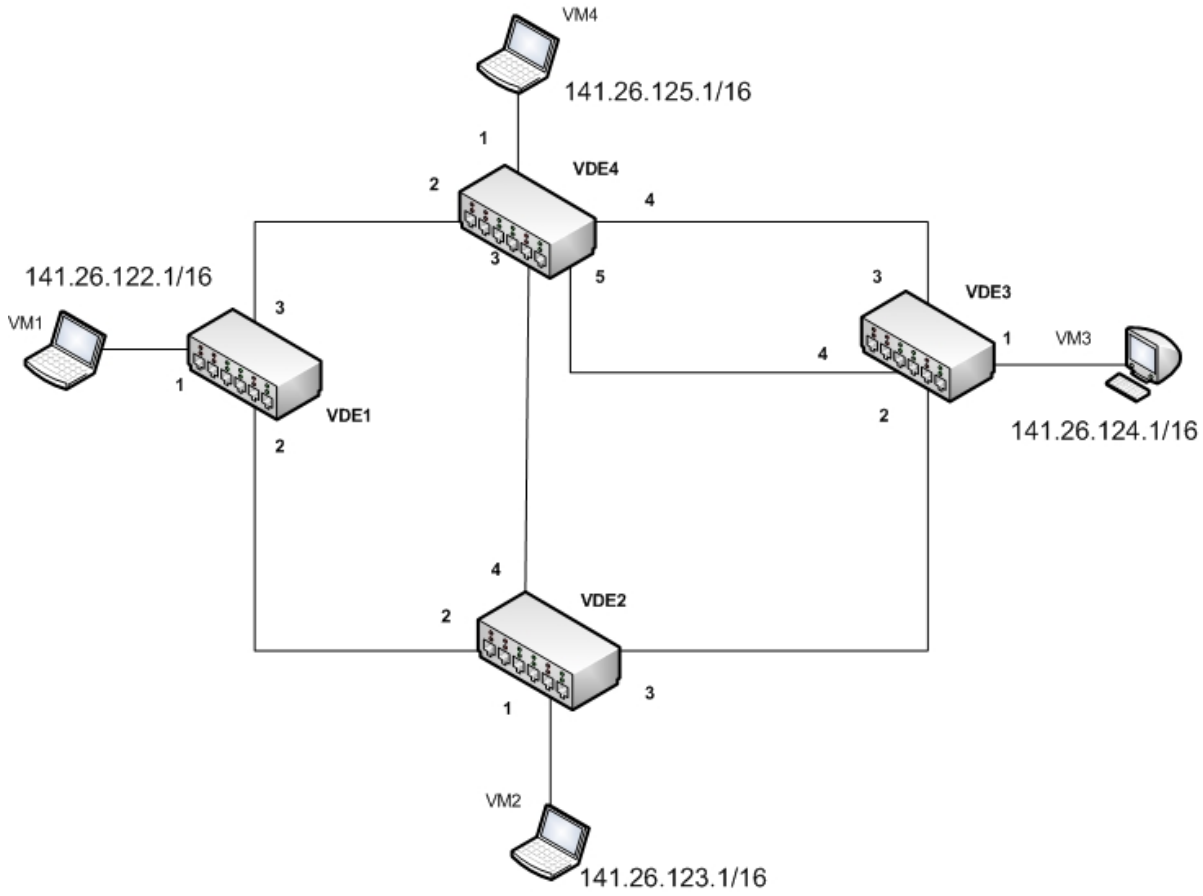

Abbildung 58: Netzwerk-Topologie(VDE)

9 Lösungen

Serdar Ayalp, Suat Algin

1. **Aufgabe:**
   Ein Cluster setzt sich aus verschiedenen Switches, die miteinander verbunden sind zusammen. Hierbei werden die Switches zu einer Gruppe zusammengefasst und über einen einzigen Switch administriert. Der große Vorteil eines Clusters ist, dass man nur einem Switch eine IP-Adresse vergibt und dann die Konfiguration der anderen Geräte anhand dieses Switches vornimmt. Auch in einem Fehlerfall, ist der Administrator nicht verpflichtet auf jeden Switch gesondert zu zugreifen, sondern nimmt seine Fehlerbehebung über den sogenannten Root-Switch vor.

2. **Aufgabe:**
   Ein VLAN ist ein virtuelles Netz, welches innerhalb eines physischen Netzes gebildet wird. Im Gegensatz zu früher ist man in der Lage verschiedene Geräte, die sich physikalisch in dem gleichen Netz befinden voneinander zu trennen und so eine Kommunikation zu unterbinden. Hierbei bedeutet Single-VLAN Membership, dass das Endgerät nur zu einem VLAN gehören darf. Das heißt es wird einem bestimmten VLAN zugewiesen und kann auch nur innerhalb dieses VLANs kommunizieren. Im Gegensatz dazu, kann man beim Multi-VLAN Membership zu mehreren VLANs zugehören. Dies hat zur Folge, dass man eine größere Zugriffsberechtigung hat. Diese Technik wird meistens dazu genutzt, um in Unternehmen bestimmten Personen(Administrator, Abteilungsleiter usw.) den Zugriff auf mehrere virtuelle Netze zu gewährleisten. Wie in Kapitel 5.2 zu sehen ist, kann man dadurch auch den Zugriff auf das Internet limitieren.

3. **Aufgabe:**
   Voraussetzung für die Konfigurierung ist der erfolgreiche Verbindungsaufbau:

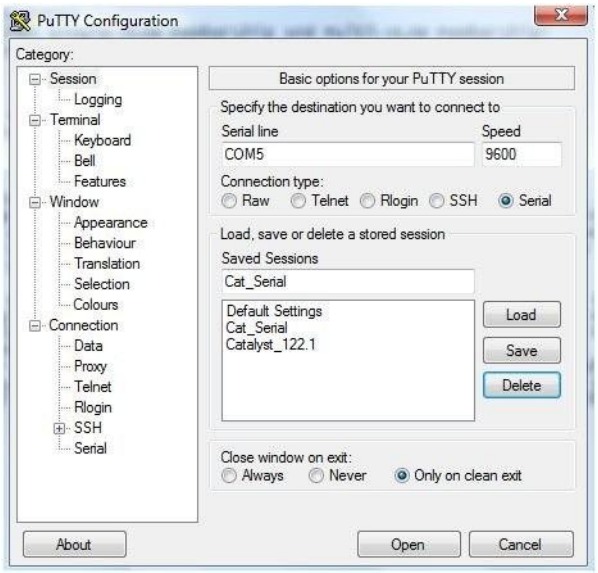

Abbildung 59: Verbindungsaufbau unter Vista per TuTTY

Nach dem Verbindungsaufbau, konfiguriert man die einzelnen Parameter:

```
Cat_3500>enable
Password:
Cat_3500#setup

         --- System Configuration Dialog ---

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Continue with configuration dialog? [yes/no]: y
Enter IP address: 141.26.128.1
Enter IP netmask: 255.255.0.0
Would you like to enter a default gateway address? [yes]: n
Enter host name [141.26.124.1]: Catalyst_1

The enable secret is a one-way cryptographic secret used
instead of the enable password when it exists.

Enter enable secret: xxx

Would you like to configure a Telnet password? [yes]: n
```

```
The following configuration command script was created:

ip subnet-zero
interface VLAN1
ip address 141.26.128.1 255.255.0.0
hostname 141.26.128.1
enable secret 5 $1$PGia$ZPMf2njeBeWCCedA61VaU1
line vty 0 15
no password
snmp community private rw
snmp community public ro
!
end

Use this configuration? [yes/no]: y
Building configuration...
[OK]
Use the enabled mode 'configure' command to modify this configuration.

Press RETURN to get started.


141.26.128.1#
```

Als nächstes verbindet man sich über einen Browser auf den konfigurierten Switch. Dazu gibt man einfach die IP-Adresse(141.26.128.1) in der Adressleiste ein und erzeugt ein Cluster:

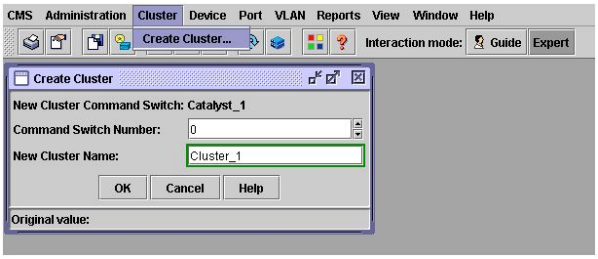

Abbildung 60: Cluster erzeugen

Als nächstes nehmen wir die Switches in das Cluster auf:

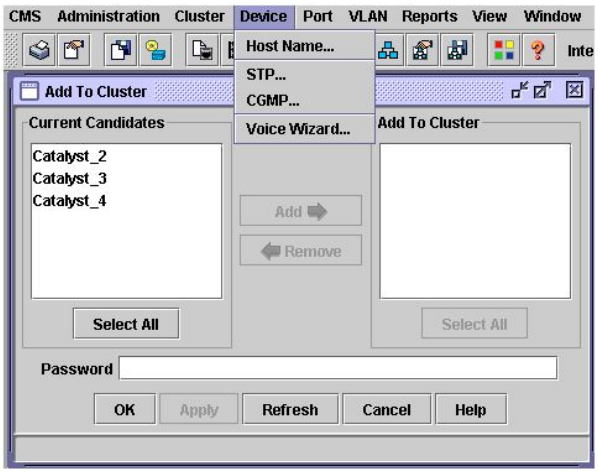

Abbildung 61: Aufnahme in das Cluster

Als letzten Schritt, vergibt man die Portkosten, so dass die gewünschte Topologie entsteht:

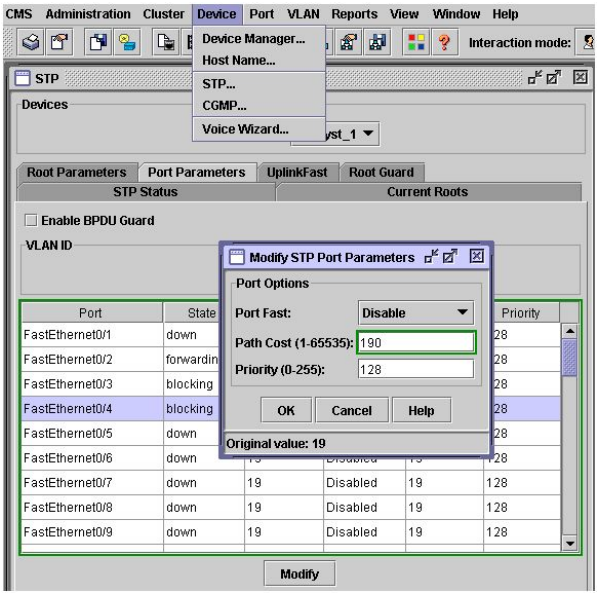

Abbildung 62: Vergabe der STP-Kosten

Die IP-Adressen der Rechner werden manuell an den Rechnern selbst eingegeben. Nun erhalten wir folgende Topologie:

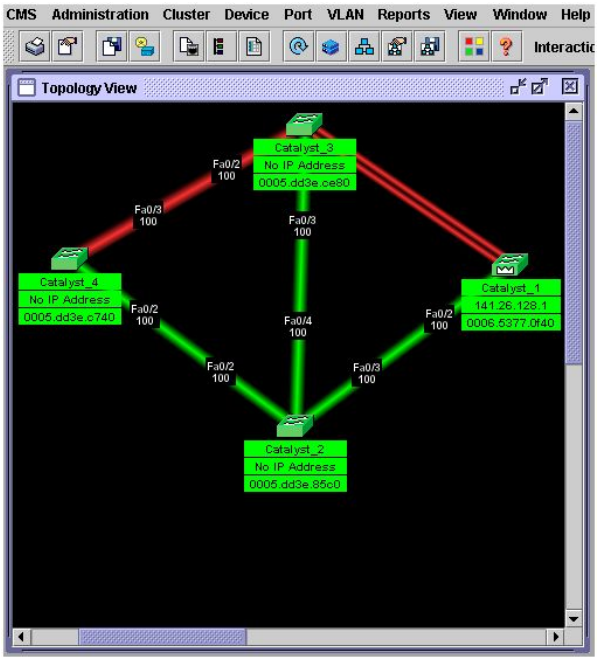

Abbildung 63: Die endgültige Topologie

4. **Aufgabe:**

   - Vier VDE_Switches erstellen.

```
~/vde-2#  vde_switch -d -s /tmp/vde1 -M /tmp/mgmt1
~/vde-2#  vde_switch -d -s /tmp/vde2 -M /tmp/mgmt2
~/vde-2#  vde_switch -d -s /tmp/vde3 -M /tmp/mgmt3
~/vde-2#  vde_switch -d -s /tmp/vde4 -M /tmp/mgmt4
```

   - VMs erstellen und jeweils mit einem Switch verbinden.

```
~/vde-2# ./linux ubd0=cow1.img,mini_vnuml-1.8.img
            con0=xterm eth1=daemon,,,/tmp/vde1/ctl &

// Virtuelle Maschine 1
login : root
password : xxxx
root@router:~# ifconfig eth1 141.26.122.1 netmask 255.255.0.0 up


~/vde-2# ./linux ubd0=cow2.img,mini_vnuml-1.8.img
            con0=xterm eth1=daemon,,,/tmp/vde2/ctl &
```

```
// Virtuelle Maschine 2
login : root
password : xxxx
root@router:~# ifconfig eth1 141.26.123.1 netmask 255.255.0.0 up


~/vde-2# ./linux ubd0=cow3.img,mini_vnuml-1.8.img
             con0=xterm eth1=daemon,,,/tmp/vde3/ctl &

// Virtuelle Maschine 3
login : root
password : xxxx
root@router:~# ifconfig eth1 141.26.124.1 netmask 255.255.0.0 up


~/vde-2# ./linux ubd0=cow4.img,mini_vnuml-1.8.img
             con0=xterm eth1=daemon,,,/tmp/vde4/ctl &

// Virtuelle Maschine 4
login : root
password : xxxx
root@router:~# ifconfig eth1 141.26.125.1 netmask 255.255.0.0 up
```

- In die Switches einloggen und FSTP auf jedem Switches wie folgt aktivieren.

```
~/vde-2# unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/setfstp 1
0000 DATA END WITH '.'
.
1000 Success

vde$
```

- Switches miteinander nach ihrer Reihenfolge verbinden.

```
//VDE1 mit VDE2 verbinden
~/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde2 &
[5] 8022

//VDE1 mit VDE4 verbinden
~/vde-2# dpipe vde_plug /tmp/vde1 = vde_plug /tmp/vde4 &
[6] 8031

//VDE2 mit VDE3 verbinden
~/vde-2# dpipe vde_plug /tmp/vde2 = vde_plug /tmp/vde3 &
[7] 8037
```

```
//VDE2 mit VDE4 verbinden
~/vde-2# dpipe vde_plug /tmp/vde2 = vde_plug /tmp/vde4 &
[8] 8042

//VDE3 mit VDE4 verbinden
~/vde-2# dpipe vde_plug /tmp/vde3 = vde_plug /tmp/vde4 &
[9] 8047

//Noch einmal VDE3 mit VDE4 verbinden
~/vde-2# dpipe vde_plug /tmp/vde3 = vde_plug /tmp/vde4 &
[10] 8059
```

- In die Switches einloggen und Port-Zustände verifizieren.

```
//VDE-1
~/vde/trunk/vde-2# unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          6          bytes                  468
 OUT: pkts        208          bytes                11024
  -- endpoint ID 0003 module unix prog   :
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        157          bytes                 8321
 OUT: pkts        146          bytes                 7738
  -- endpoint ID 0008 module unix prog   : vde_plug: user=root
                         PID=8022 SOCK=/tmp/vde1/.08022-00000
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        137          bytes                 7261
 OUT: pkts        138          bytes                 7314
  -- endpoint ID 0010 module unix prog   : vde_plug: user=root
                         PID=8031 SOCK=/tmp/vde1/.08031-00000
.
1000 Success

vde$
```

```
//VDE-2
~/vde/trunk/vde-2# unixterm /tmp/mgmt2
VDE switch V.2.2.3
```

```
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          6          bytes                 468
 OUT: pkts        225          bytes               11925
  -- endpoint ID 0003 module unix prog   :
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        166          bytes                8798
 OUT: pkts        176          bytes                9328
  -- endpoint ID 0008 module unix prog   : vde_plug: user=root
                          PID=8023 SOCK=/tmp/vde2/.08023-00000
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        154          bytes                8162
 OUT: pkts        160          bytes                8480
  -- endpoint ID 0010 module unix prog   : vde_plug: user=root
                          PID=8037 SOCK=/tmp/vde2/.08037-00000
Port 0004 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        151          bytes                8003
 OUT: pkts        149          bytes                7897
  -- endpoint ID 0012 module unix prog   : vde_plug: user=root
                          PID=8042 SOCK=/tmp/vde2/.08042-00000
.
1000 Success

vde$
```

```
//VDE-3
~/vde/trunk/vde-2# unixterm /tmp/mgmt3
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          6          bytes                 468
 OUT: pkts        234          bytes               12402
  -- endpoint ID 0003 module unix prog   :
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
```

```
 IN:  pkts        171          bytes                 9063
 OUT: pkts        165          bytes                 8745
  -- endpoint ID 0008 module unix prog   : vde_plug: user=root
                          PID=8038 SOCK=/tmp/vde3/.08038-00000
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        152          bytes                 8056
 OUT: pkts        152          bytes                 8056
  -- endpoint ID 0010 module unix prog   : vde_plug: user=root
                          PID=8047 SOCK=/tmp/vde3/.08047-00000
Port 0004 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        149          bytes                 7897
 OUT: pkts        149          bytes                 7897
  -- endpoint ID 0012 module unix prog   : vde_plug: user=root
                          PID=8059 SOCK=/tmp/vde3/.08059-00000
.
1000 Success

vde$
```

```
//VDE-4
~/vde/trunk/vde-2# unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ port/allprint
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: NONE Access Control: (User: NONE - Group: NONE)
 IN:  pkts          6          bytes                  468
 OUT: pkts        241          bytes                12773
  -- endpoint ID 0003 module unix prog   :
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        180          bytes                 9540
 OUT: pkts        178          bytes                 9434
  -- endpoint ID 0008 module unix prog   : vde_plug: user=root
                          PID=8032 SOCK=/tmp/vde4/.08032-00000
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        171          bytes                 9063
 OUT: pkts        173          bytes                 9169
  -- endpoint ID 0010 module unix prog   : vde_plug: user=root
                          PID=8043 SOCK=/tmp/vde4/.08043-00000
```

```
Port 0004 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        163          bytes                 8639
 OUT: pkts        162          bytes                 8586
  -- endpoint ID 0012 module unix prog   : vde_plug: user=root
                          PID=8048 SOCK=/tmp/vde4/.08048-00000
Port 0005 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
 Current User: root Access Control: (User: NONE - Group: NONE)
 IN:  pkts        160          bytes                 8480
 OUT: pkts        159          bytes                 8427
  -- endpoint ID 0014 module unix prog   : vde_plug: user=root
                          PID=8060 SOCK=/tmp/vde4/.08060-00000
.
1000 Success

vde$
```

- Die Verbindung der VMs testen.

```
//VM-1
root@router:~# ping -c 4 141.26.123.1
PING 141.26.123.1 (141.26.123.1): 56 data bytes
84 bytes from 141.26.123.1: icmp_seq=0 ttl=64 time=21.1 ms
84 bytes from 141.26.123.1: icmp_seq=1 ttl=64 time=0.4 ms
84 bytes from 141.26.123.1: icmp_seq=2 ttl=64 time=0.4 ms
84 bytes from 141.26.123.1: icmp_seq=3 ttl=64 time=0.4 ms

--- 141.26.123.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.4/5.5/21.1 ms

root@router:~# ping -c 4 141.26.124.1
PING 141.26.124.1 (141.26.124.1): 56 data bytes
84 bytes from 141.26.124.1: icmp_seq=0 ttl=64 time=21.4 ms
84 bytes from 141.26.124.1: icmp_seq=1 ttl=64 time=0.5 ms
84 bytes from 141.26.124.1: icmp_seq=2 ttl=64 time=0.4 ms
84 bytes from 141.26.124.1: icmp_seq=3 ttl=64 time=0.4 ms

--- 141.26.124.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.4/5.6/21.4 ms

root@router:~# ping -c 4 141.26.125.1
PING 141.26.125.1 (141.26.125.1): 56 data bytes
84 bytes from 141.26.125.1: icmp_seq=0 ttl=64 time=22.0 ms
84 bytes from 141.26.125.1: icmp_seq=1 ttl=64 time=0.4 ms
84 bytes from 141.26.125.1: icmp_seq=2 ttl=64 time=0.5 ms
84 bytes from 141.26.125.1: icmp_seq=3 ttl=64 time=0.4 ms

--- 141.26.125.1 ping statistics ---
```

```
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.4/5.8/22.0 ms
root@router:~#
```

- Verifizierung der Port-Rollen in FSTP.

```
//VDE-1
~/vde/trunk/vde-2# unixterm /tmp/mgmt1
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated 80:00:00:ff:43:e6:62:0a
 ++ rootport 0002 cost 20000000 age 2 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$
```

```
//VDE-2
~/vde/trunk/vde-2# unixterm /tmp/mgmt2
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 1618 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Designated
 -- Port 0003 tagged=0 portcost=20000000 role=Designated
 -- Port 0004 tagged=0 portcost=20000000 role=Designated
.
1000 Success

vde$
```

```
//VDE-3
~/vde/trunk/vde-2# unixterm /tmp/mgmt3
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated 80:00:00:ff:43:e6:62:0a
 ++ rootport 0002 cost 20000000 age 0 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0004 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$



//VDE-4
/vde/trunk/vde-2# unixterm /tmp/mgmt4
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated 80:00:00:ff:43:e6:62:0a
 ++ rootport 0003 cost 20000000 age 2 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0003 tagged=0 portcost=20000000 role=Root
 -- Port 0004 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0005 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$
```

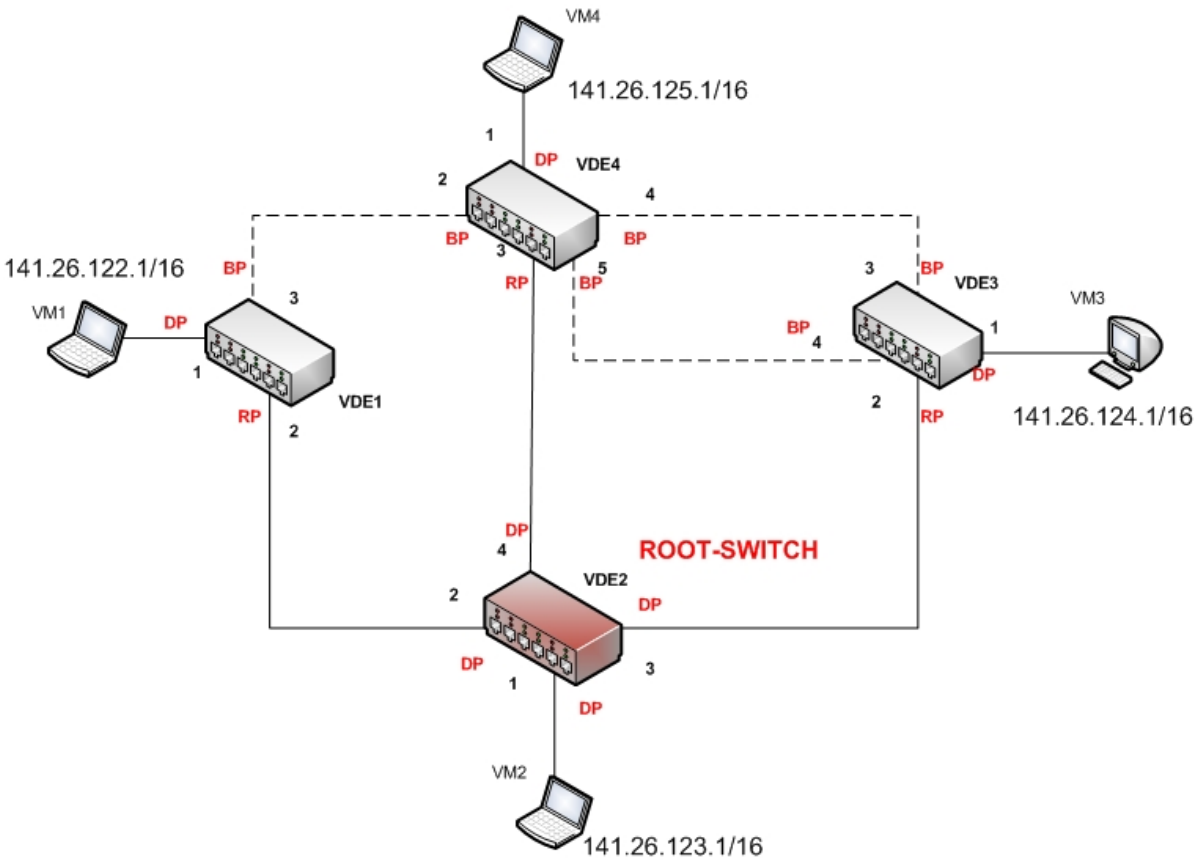

Abbildung 64: Port-Rollen in FSTP

- Die Verbindung zwischen VDE1 und VDE2 fällt aus. PID (Prozess ID) der Verbindung ist *8023*.

```
kill -9 8023
```

- Verifiziere noch einmal die Port-Rollen in FSTP.

```
//VDE-1
vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated 80:00:00:ff:a2:f0:9c:76
 ++ rootport 0003 cost 40000000 age 3 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0003 tagged=0 portcost=20000000 role=Root
.
```

```
1000 Success

vde$



//VDE-2
vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000 ROOTSWITCH
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated ff:ff:ff:ff:ff:ff:ff:ff
 ++ rootport 0000 cost 0 age 3217 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0003 tagged=0 portcost=20000000 role=Designated
 -- Port 0004 tagged=0 portcost=20000000 role=Designated
.
1000 Success

vde$



//VDE-3
vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated 80:00:00:ff:43:e6:62:0a
 ++ rootport 0002 cost 20000000 age 3 bonusport 0000 bonuscost 0
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Root
 -- Port 0003 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0004 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$



//VDE-4
vde$ fstp/print
0000 DATA END WITH '.'
FST DATA VLAN 0000
 ++ root 80:00:00:ff:43:e6:62:0a
 ++ designated 80:00:00:ff:43:e6:62:0a
 ++ rootport 0003 cost 20000000 age 3 bonusport 0000 bonuscost 0
```

```
 -- Port 0001 tagged=0 portcost=20000000 role=Designated
 -- Port 0002 tagged=0 portcost=20000000 role=Designated
 -- Port 0003 tagged=0 portcost=20000000 role=Root
 -- Port 0004 tagged=0 portcost=20000000 role=Alternate/Backup
 -- Port 0005 tagged=0 portcost=20000000 role=Alternate/Backup
.
1000 Success

vde$
```

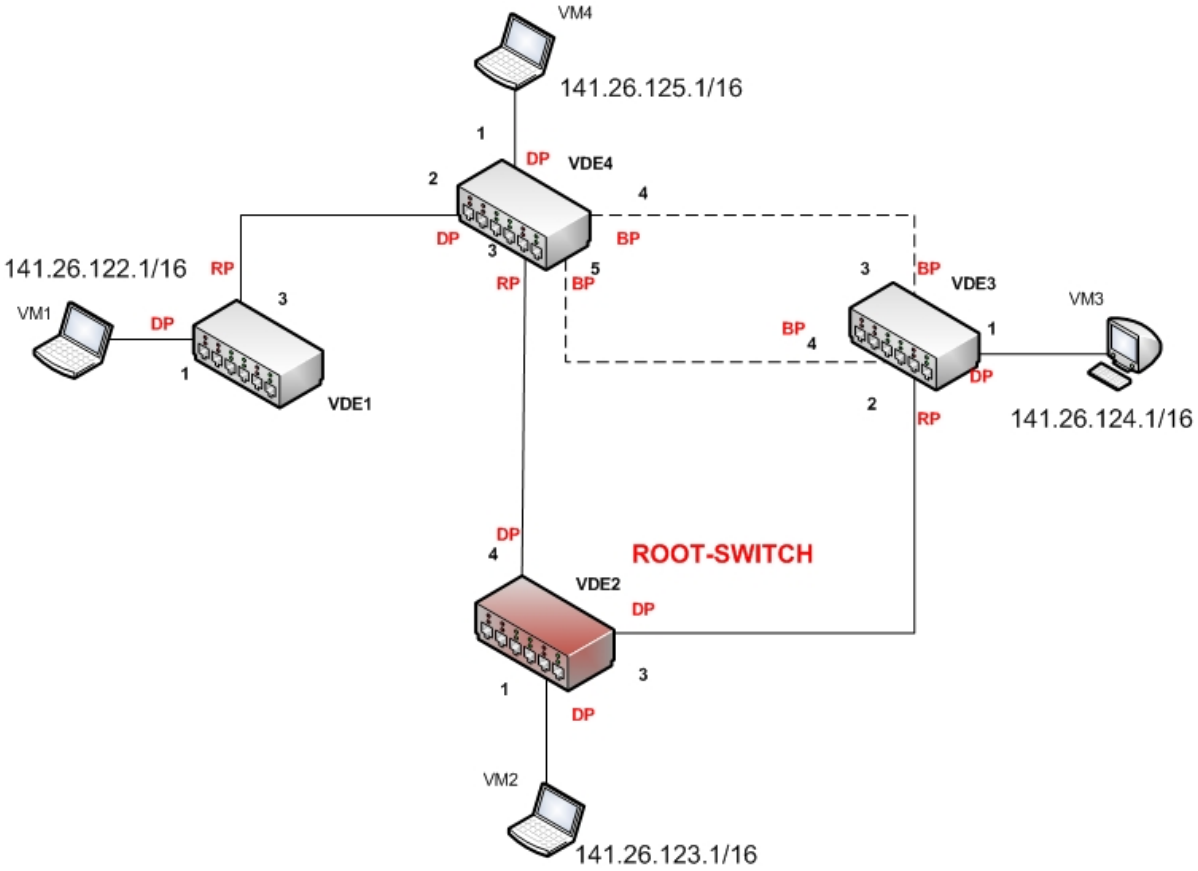

Abbildung 65: Port-Rollen nach dem Ausfall der Verbindung

Abbildungsverzeichnis

Literatur

[1] Anonym. *802.1D Spanning Tree Protocol*. http://networkninja.co.za/?p=34.

[2] Cisco. *Cisco Systems*. http://www.cisco.com.

[3] Dan DiNicolo. *Line Configuration Mode*. http://www.2000trainers.com/cisco-ccna-07/ccna-line-configuration-mode.

[4] Sebastian Hein. *Spanning Tree Algorithmus*. http://www.all-about-security.de/security-artikel/netzwerk-sicherheit/nac-network-access-control/artikel/2691-spanning-tree-algorithmus/.

[5] Wolfgang Schulte. *Spanning Tree*. http://funkschau.info/heftarchiv/pdf/2003/fs1603/fs0316055.pdf.

[6] Petra Treubel. *Einführung in die Konfiguration von Cisco Switches*. http://www.treubel.net/lade_daten.php?download=101.

[7] Suat Algin und Serdar Ayalp. *Switchkonfiguration und Management am Bespiel des Cisco Catalyst 5500*. Verlag Dr. Müller, 1 edition, 2008.

[8] Virtualsquare-Projekt. *Fast Spanning Tree Protocol*. http://wiki.virtualsquare.org/index.php/Fast_Spanning_Tree_Protocol.

[9] 18.08.2009 Wikipedia, Die freie Enzyklopädie. *Boot-Loader*. http://de.wikipedia.org/wiki/Boot-Loader.

[10] 18.08.2009 Wikipedia, Die freie Enzyklopädie. *Hot Standby Router Protocol*. http://de.wikipedia.org/wiki/Hot_Standby_Router_Protocol.

[11] 18.08.2009 Wikipedia, Die freie Enzyklopädie. *Internetwork Operating System*. http://de.wikipedia.org/wiki/Internetwork_Operating_System.

[12] 18.08.2009 Wikipedia, Die freie Enzyklopädie. *Maximum Transmission Unit*. http://de.wikipedia.org/wiki/Maximum_Transmission_Unit.

[13] 18.08.2009 Wikipedia, Die freie Enzyklopädie. *Promiscuous Mode*. http://de.wikipedia.org/wiki/Promiscuous_Mode.

[14] 18.08.2009 Wikipedia, Die freie Enzyklopädie. *Trivial File Transfer Protocol*. http://de.wikipedia.org/wiki/Trivial_File_Transfer_Protocol.

[15] 18.08.2009 Wikipedia, Die freie Enzyklopädie. *User Mode Linux*. http://de.wikipedia.org/wiki/User_Mode_Linux.

[16] Markus Wilthaner. *Segmentierung mit Bridges oder Switches*. http://school.wilth.net/prru/kapitel4.htm#_Toc29916037.