TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik
Professur Rechnernetze und verteilte Systeme

Studienarbeit

"Klassenbasierter Anschluss im Chemnitzer Studenten Netz"

Bearbeitet von: Sebastian Junge

Betreuer: Ronald Schmidt

Chemnitz, 31.07.2005

Aufgabenstellung

"Klassenbasierter Anschluss im Chemnitzer Studenten Netz"

Im Rahmen dieser Studienarbeit soll die Bedeutung klassenbasierter Anschlüsse im CSN untersucht und Tools bereit gestellt werden, welche eine weitgehend automatisierte Anlage/Verwaltung solcher Netzklassen ermöglichen. Hiermit soll ein Werkzeug geschaffen werden um die Möglichkeiten der Netznutzung für einzelne Nutzer-Ports differenziert gestalten zu können, unter den wichtigen Aspekten der Sicherheit und Transparenz für die Nutzer. Beispiele für Netzklassen wären ein Quarantäne-Netz für wurminfizierte Rechner mit beschränktem Internetzugang und ein Anmelde-Netz für neue Nutzer zur Vereinfachung der CSN-Anmeldung.

In den theoretischen Betrachtungen ist auf die Bedeutung von Netzklassen im CSN einzugehen und sinnvolle Netzklassen vorzustellen. Verschiedene Möglichkeiten der Implementierung (z.B. 802.1q, 802.1x, Subnetting) und deren Zusammenwirken (z.B. in den Bereichen Authentifizierung und Kapselung der Netzklassen) sind im Rahmen der CSN-Netzstruktur auf ihre Vor- und Nachteile zu prüfen. Desweiteren soll neben der Erläuterung der verwendeten Technologien ein Ausblick gegeben werden, welche Möglichkeiten durch den erweiterten Einsatz modernster Netzwerktechnik im CSN zur Verfügung stehen.

Als praktische Realisierung ist eine Webapplikation für die Anlage/Verwaltung der Netzklassen zu erstellen, welche die Einrichtung entsprechender Netzklassen auf allen Switches und Management der Nutzer-Ports erlaubt. Für Squid als transparentenProxy und Iptables für Filtering, NAT und Redirection sollen ebenfalls Konfigurationen erzeugt werden.

Selbstständigkeitserklärung

Hier erkläre ich, daß ich die vorliegende Arbeit mit dem Titel “Klassenbasierter Anschluss im Chemnitzer Studentennetz” selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt, keine anderen als die angegeben Hilfsmittel benutzt und wörtliche sowie sinngemäße Zitate als solche gekennzeichnet habe.

Chemnitz, 31.07.2005

Sebastian Junge

Inhaltsverzeichnis

1. Einleitung

1.1. Motivation

Diese Arbeit soll Möglichkeiten untersuchen, wie man verschiedene Arten der Netznutzung auf einem Netz realiseren kann und eine Lösung angepasst auf die Infrastruktur des Chemnitzer Studenten Netzes [1] liefern. An der Aufgabe reizt mich besonders die aktuell existierende Notwendigkeit, daß der CSN-Zugang differenzierter gestaltet werden muß und das als Ergebnis eine praktische Implementation entstehen soll, die sich täglich im Einsatz befinden wird.

Bisher bieten sich für die Nutzer des CSN grob gesagt nur 2 Möglichkeiten der Netznutzung, entweder vollständig oder garnicht. Wird zum Beispiel ein Rechner im Falle eines Wurmbefalls gesperrt, so hat der Nutzer meist keine Möglichkeit, seinen Rechner selbständig wieder zu säubern. Dies muß ein Mitarbeiter des CSN übernehmen, was in Zeiten massiver Wurmangriffe zu starker Arbeitsbelastung führt.
Dies ist nur ein Beispiel, wo eine Einführung einer neuen Nutzungsklasse, nämlich in diesem Fall einer Klasse für wurminfizierte Rechner mit beschränktem Internetzugang, sinnvoll wäre.

In Kapitel 2 werde ich genauer auf die aktuellen Probleme eingehen, die verschiedenen Arten der Netznutzungsmöglichkeiten im CSN abzugrenzen und den Begriff “Netzklasse” im Allgemeinen definieren. Kapitel 3 gibt eine Übersicht über die für das Thema interessanten Technologien. Im Kapitel 4 wird die Anwendbarkeit der Technologien auf Basis der CSN-Netzstruktur untersucht und bewertet.
Meine eigene Implementation werde ich in Kapitel 5 darstellen. Der Anhang gibt einen Überblick über die wichtigste Funktionalität, welche zur Konfiguration der Netzklassen benötigt wird.

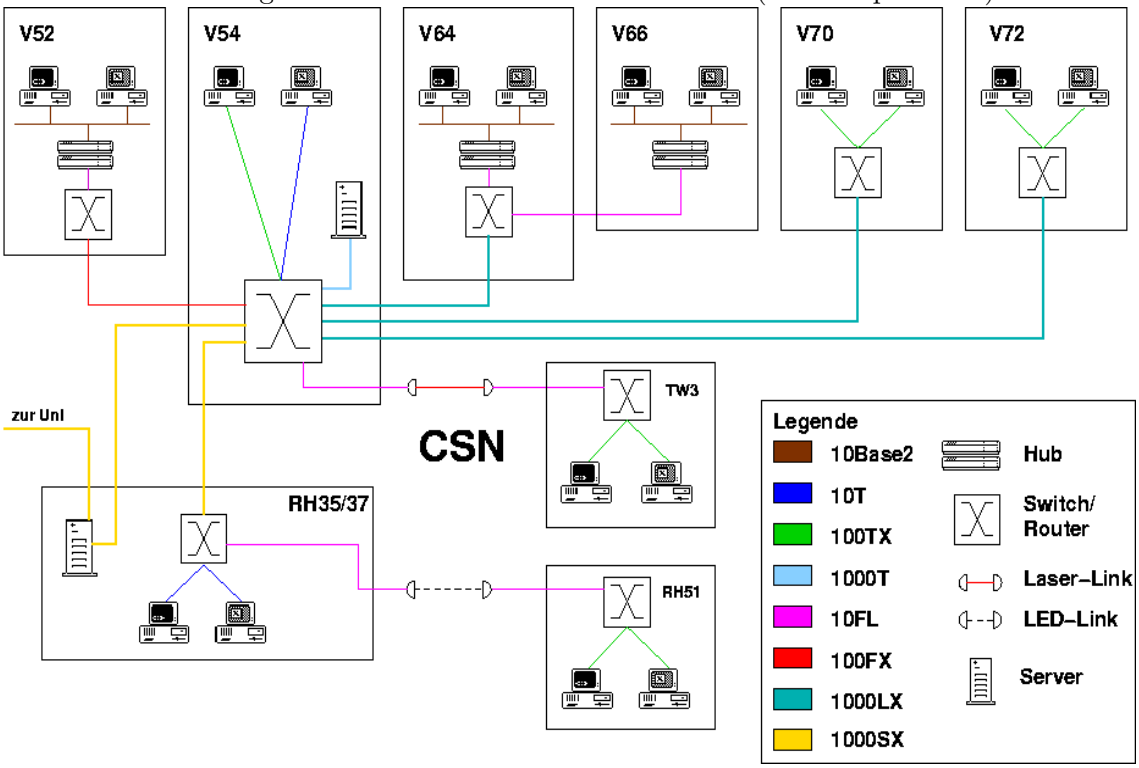
1.2. Überblick über das Chemnitzer Studenten Netz

Das Chemnitzer Studenten Netz wurde 1994 gegründet und ermöglicht Studenten, welche in den Wohnheimen der Technischen Universität Chemnitz wohnen, ihren Rechner mit dem Campusnetz zu verbinden. Dieses Netz ist ein Forschungsnetz mit einer heterogenen Twisted-Pair/BNC Struktur und einem weitgehend automatisierten Netzmanagement. Aktuell sind ca. 1800 Rechner in den 9 Wohnheimen angeschlossen. Verwaltet wird das CSN ehrenamtlich von Studenten, mit Unterstützung des Studentenwerkes und des Rechenzentrums der TU-Chemnitz.
Im Rahmen dieses Abschnittes werde ich grob auf die aktuelle Netzstruktur und die eingesetzte Technik im CSN eingehen, da diese beiden Punkte eine wichtige Grundlage für spätere Entscheidungen in dieser Arbeit darstellen. Sehr detaillierte Informationen über Struktur und Aufbau des CSN können aus folgendem Dokument [2] bezogen werden, wobei natürlich beachtet werden muß, daß das CSN sich ständig weiterentwickelt.
In naher Zukunft ist der Umbau der letzten beiden per BNC verkabelten Wohnheime auf

Twisted-Pair Verkabelung geplant und das Ersetzen grosser Teile der Switches durch neuere und leistungstärkere Modelle.

1.2.1. Netzstruktur

Abbildung 1.1.: Aktuelle Netzstruktur des CSN (Stand April 2005)



Die wichtigsten Elemente der Netzstruktur sind der CSN-Router, an dem bis auf ein Haus alle Wohnheime direkt angeschlossen sind, und der CSN-Server, welcher unter anderem die Aussenanbindung des CSN darstellt. Dies sind die einzigen Netzelemente, welche auf OSI-Layer 3 arbeiten.
Aktuell existieren “Management-VLAN” für die Konfiguration der Netztechnik und sogenannte ”Haus-VLAN“, welche sich über jeweils ein Haus erstrecken und denen alle Nutzer eines Hauses zugeordnet sind.
Ein Nutzer meldet seinen Rechner einmal am CSN an und danach erhält nurnoch dieser Rechner auf dem jeweiligen Switchport Netzzugang, da Portsecurity auf allen Switches eingesetzt wird. Jedem Rechner wird bei der Anmeldung eine IP-Adresse zugeteilt und dem Nutzer bleibt es dann selbst überlassen, ob er die Konfiguration statisch einstellt oder sich dynamisch per DHCP holt.
Weitere Elemente der Nutzerverwaltung sind der Umzug innerhalb des CSN und der Auszug aus dem Wohnheim. Die entsprechenden Daten werden von einer Datenbank verwaltet.

1.2.2. Eingesetzte Netztechnik

Bis auf die Vettersstrasse 64/66 sind alle Wohnheime mit Twisted-Pair verkabelt und vollgeswitcht mit 10MBit/S und teilweise auch 100MBit/S im Zimmer.

Übersicht über die verwendete Rechentechnik pro Wohnheim:

- Reichenhainer Strasse 35/37
  In jeder zweiten Etage stehen drei 3com1100 Switches, untereinander verbunden über Patchkabel. Beide Wohnheime teilen sich einen Hausswitch 3com3300. Im Keller befindet sich der CSN-Server mit der Aussenanbindung an das Rechenzentrum der TU-Chemnitz und der Uni-Radio Server.

- Reichenhainer Strasse 51
  Ein Stack aus Cisco 3750 versorgt das ganze Wohnheim, angebunden an das CSN über einen LED-Link zur Reichenhainer Strasse 35/37.

- Vettersstrasse 52
  Dieses Wohnheim wurde zuletzt von BNC auf TwistedPair umgerüstet, eingesetzte Rechentechnik sind 2 Stacks aus Cisco 3750.

- Vettersstrasse 54
  Ausrüstung der Etagen mit gestackten 3com1100-3300 Switches, Im Keller befindet sich der CSN-Router, welcher über GigaBit an die V70/72, V64/66 und Rh35/37 angebunden ist.

- Vettersstrasse 64/66
  Beide Wohnheime sind derzeit noch BNC-verkabelt, mit geswitchten Segmenten pro Etage. Ein cisco3550 wird als gemeinsamer Hausswitch eingesetzt.

- Vettersstrasse 70/72
  Eingesetzt werden gestackte 3com3300 und ein Cisco 3750.

- Thüringer Weg 3
  Dieses Wohnheim verfügt über 3 gestackte 3com3300 Switches und ist an das restliche Netz mittels eines Laserlinks angebunden.

1.2.3. IP-Adressvergabe

Die IP-Adresskonfiguration der Nutzerrechner ist auf hausweiten virtuellen Netzen aufgesetzt. Jedes Haus verfügt über ein “Haus-VLAN” mit einem eigenen IP-Subnetz. Aus diesem Adressraum werden den Nutzerrechnern IP-Adressen zugeteilt.
Der für jedes Haus festgelegte Standardgateway ist das entsprechende Interface des CSN-Routers, an welchem das Wohnheim angeschlossen ist. Alle Netztechnikgeräte pro Haus arbeiten ausschliesslich auf OSI-Layer 2.

Tabelle 1.1.: IP-Adresskonfiguration der “Haus-VLAN” im CSN

Wohnheim	Subnetz	Gateway IP
Reichenhainer Str. 35	134.109.80.0 / 255.255.254.0	134.109.81.254
Reichenhainer Str. 37	134.109.84.0 / 255.255.254.0	134.109.85.254
Reichenhainer Str. 51	134.109.114.0 / 255.255.254.0	134.109.115.254
Vettersstrasse 52	134.109.92.0 / 255.255.254.0	134.109.93.254
Vettersstrasse 54	134.109.96.0 / 255.255.254.0	134.109.97.254
Vettersstrasse 64	134.109.116.0 / 255.255.254.0	134.109.117.254
Vettersstrasse 66	134.109.104.0 / 255.255.254.0	134.109.105.254
Vettersstrasse 70	134.109.88.0 / 255.255.254.0	134.109.89.254
Vettersstrasse 72	134.109.108.0 / 255.255.254.0	134.109.109.254
Thüringer Weg 3	134.109.112.0 / 255.255.254.0	134.109.113.254

1.2.4. Konfiguration der Netztechnik

Die einzelnen Switches werden bisher von Hand per Telnet oder Command Line Interface eingerichtet.

Das Konfigurieren der Nutzerports erfolgt in regelmässigen Abständen automatisch durch das Switchconfig-Skript. Dieses nutzt die CSN-Switch-API, welche einen grundlegenden Funktionsumfang zur Konfiguration von Switchports mittels SNMP bereitstellt.
Es können Ports geschlossen/geöffnet, MAC-Adressen für PortSecurity eingetragen und Regeln für die Handhabung des Verletztens der PortSecurity gesetzt werden.

Diese Funktionalität wird in einem Perl Modul gekapselt und ist weitgehend unabhängig vom verwendeten Gerätemodell. Durch die Objektorientierte Programmierweise wird eine dem Gerätetyp 3com/Cisco entsprechende Klasse instanziiert und zurückgegeben.

2. Problembeschreibung und Begriffsdefinition Netzklasse

Im ersten Abschnitt dieses Kapitels werde ich einige Problemfälle erläutern, die sich aktuell im CSN ergeben und auch Anstoss für diese Studienarbeit sind. Danach möchte ich skizzieren, welche verschiedenen Möglichkeiten der Netznutzung sich im CSN unterscheiden lassen und anhand deren wichtigster Eigenschaften den Begriff der Netzklasse erklären und dessen Komponenten spezifizieren.

2.1. Problemfälle

Wie schon in der Einleitung angedeutet, existieren im Moment im CSN für die Rechner von Nutzern nur zwei Arten des Netzzugangs. Wenn der Nutzer freigeschalten ist, dann verfügt er über vollen Netzzugriff, ist er hingegen gesperrt, verfügt er über keinerlei Zugang. Es gibt momentan keine Möglichkeit, verschiedene Nutzungsklassen durch Beschränkungen von Zugriffsrechten zu erschaffen und diesen Klassen dann Nutzerports zuzuteilen. Der Begriff Nutzerports steht für Switchports, an denen Nutzerrechner, also Endgeräte, angeschlossen sind.
Konkrete Problemfälle sind zum Beispiel das Anmelden eines Rechners am CSN und die Quarantäne für wurminfizierte Rechner.
Will jemand einen Rechner am CSN anmelden, so kann er dies nicht vom eigenen Rechner aus tun. Der Rechner hat keinen Netzzugang, da er noch gesperrt ist und erst nach erfolgter Anmeldung freigeschalten wird. Somit ist der Nutzer gezwungen, sich von einem anderen Rechner mit Netzzugang anzumelden (z.B. aus einem Computerpool der Universität).

Ein ähnliches Problem liegt bei der Behandlung wurminfizierter Rechner vor. Sobald der Verdacht auf Wurmbefall vorliegt, werden die befallenen Rechner gesperrt und haben keinerlei Netzzugang mehr. Dies erschwert aber ein selbständiges Säubern des Rechners durch den Nutzer, da dieser nun keine Möglichkeiten mehr hat, direkt auf aktuelle Programme und Updates zuzugreifen. Somit ist der Etagenverantwortliche des CSN gefragt, welcher dann mit entsprechender Software zum Nutzer gehen und den zumeist langwierigen Säuberungsprozess durchführen muß.

Das “Zertifikat Internet Nutzung” ist zwingende Vorraussetzung für Erhaltung des Netzzugangs im CSN. Fehlt einem Nutzer trotz angemeldeten Rechners noch dieses ZIN, hat er keinen Zugang zum Netz. Auch hier würde sich eine differenziertere Netznutzung anbieten, um solchen Nutzern wenigstens einen eingeschränkten Netzzugang zu gewähren. Dieser könnte z.B. auf das interne Netz der TU-Chemnitz beschränkt sein und würde diesen Nutzern die Möglichkeit bieten, praktische Erfahrungen für die ZIN-Prüfung sammeln zu können.

2.2. Möglichkeiten der Netznutzung im CSN

Es ergeben sich aufgrund oben beschriebener Problematik und des Einsatzes neuer Technologie (z.B. WLAN) verschiedene Möglichkeiten der Netznutzung im CSN.
Folgende Eigenschaften sollen für jede der Nutzungsmöglichkeiten beschrieben werden.

- Zugeordnet: Dieser Nutzungsart zugeordnete Nutzer.
- Aufgabe: Kurze Aufgabenbeschreibung dieser Nutzungsart.
- Konfiguration: Die wichtigsten technischen Parameter kurz zusammengefasst.
- Beschränkungen: Die Einschränkungen welchen die Nutzer unterliegen.
- Externer Zugriff: Inwieweit der externe Zugriff eingeschränkt ist.
- HTTP-Zugriff: Auf welche Domains der HTTP-Zugriff erlaubt ist.

Tabelle 2.1.: Anmeldenetz

Zugeordnet	Dem CSN unbekannte Nutzerrechner, also Rechner deren MAC-Adresse nicht in der Datenbank eingetragen ist.
Aufgabe	Vereinfachung der CSN-Anmeldung, diese soll vom eigenen Rechner aus durchgeführt werden können. Verbessertes Heranführen neuer Nutzer an das CSN.
Konfiguration	Firewall zur Kontrolle der Beschränkungen benötigt. DHCP Einsatz notwendig.
Beschränkungen	Komplette Isolation des Anmeldenetzes von den anderen Netzen.
Ext. Zugriff	Kein Routing nach ausserhalb des CSN.
HTTP-Zugriff	Nur Zugriff auf die Webseiten des CSN.

Tabelle 2.2.: Quarantänenetz

Zugeordnet	Nutzerrechner, bei welchen der Verdacht auf Wurm-/Virenbefall besteht.
Aufgabe	Schaffung einer minimalen Internetkonnektivität, welche dem Besitzer des Rechners den Zugriff auf neueste Updates und Virenscanner erlaubt. Gleichzeitig soll durch Isolation befallener Rechner eine Ausbreitung von Schädlingen verhindert werden.
Konfiguration	Firewall zur Kontrolle der Beschränkungen benötigt. Bisher genutzte IP-Adressen weiterhin nutzbar.
Beschränkungen	Komplette Isolation des Quarantäne-Netzes von anderen Netzen. Isolation der befallenen Rechner untereinander. Deaktivieren einzelner Netzwerkprotokolle (z.B. SMTP).
Ext. Zugriff	Nur HTTP-Zugriff auf erlaubte Domains.
HTTP-Zugriff	Zugriff auf die Webseiten von Anti-Viren-Software Herstellern und Microsoft.

Tabelle 2.3.: CSN-Light Netz

Zugeordnet	Nutzerrechner, deren Besitzern noch das “Zertifikat Internet Nutzung” fehlt.
Aufgabe	Netzzugang auf das interne Netz der TU-Chemnitz für Nutzer mit fehlendem ZIN.
Konfiguration	Firewall zur Kontrolle der Beschränkungen benötigt.
Beschränkungen	Intern im CSN keinerlei Einschränkungen, keine Isolation notwendig.
Ext. Zugriff	Nur Zugang auf das Campusnetz der TU-Chemnitz.
HTTP-Zugriff	Zugriff auf CSN und TU-Chemnitz Webseiten.

Tabelle 2.4.: CSN-Vollzugriff Netz

Zugeordnet	Nutzer-Rechner, die keiner der anderen Klassen zugeordnet sind.
Aufgabe	Uneingeschränkter Netzzugriff.
Konfiguration	Netzzugang direkt über den CSN-Router.
Beschränkungen	Keine.
Ext. Zugriff	Keine Einschränkungen.
HTTP-Zugriff	Keine Einschränkungen.

Tabelle 2.5.: CSN-WLAN Netz

Zugeordnet	Per WLAN verbundene und nicht am CSN angemeldete Rechner.
Aufgabe	Eingeschränkter Netzzugang für WLAN-Nutzung mit nicht am CSN angemeldeten Rechnern. Dieses Netz soll die Nutzer unterstützen, um CSN-Vollzugriff mittels WLAN zu erhalten.
Konfiguration	Firewall zur Kontrolle der Beschränkungen benötigt. DHCP-Einsatz notwendig.
Beschränkungen	Komplette Isolation des WLAN-Netzes.
Ext. Zugriff	Kein Routing nach ausserhalb des CSN.
HTTP-Zugriff	Nur Zugriff auf die CSN Webseiten.

2.3. Begriffsdefinition Netzklasse

Allgemein möchte ich den Begriff Netzklasse als ein "Paket" betrachten, welches eine Art des Netzzugangs definiert. Mit der Festlegung des Netzzugangs sollen auch gleichzeitig die Möglichkeiten der erlaubten Netznutzung beschränkt werden.

Es muß ein "Paket" entstehen, welches es erlaubt, die Bestandteile der einzelnen Netznutzungsmöglichkeiten detailliert beschreiben zu können, und gleichzeitig die Fähigkeit besitzt, die notwendigen technischen Konfigurationen vornehmen zu können, um die Vorraussetzungen für den Einsatz einer Netznutzungsmöglichkeit zu schaffen. Ebenfalls dazu gehört eine Nutzerverwaltung, welche aufgrund gewisser Regeln den Nutzern die jeweils passende Netzklasse zuordnet.
Mit der Einführung von Netzklassen im CSN soll die Grundlage geschaffen werden, den CSN-Netzzugang differenzierter gestalten zu können. Je nach Bedarf sollen neue Netzklassen angelegt und alte gelöscht werden können. Alle Arten der Netzzugangs sollen über Netzklassen definiert werden, somit muß z.B. der bisher existierende Vollzugriff einer Netzklasse zugeordnet werden.

Die Eigenschaften der Netznutzungsmöglichkeiten dienen als Basis für einen Grossteil der Komponenten einer Netzklasse. Eine Netzklasse muß somit aus theoretischer Sicht mindestens folgende Eigenschaften haben.

- Name - Identifikation einer Netzklasse über ihren Namen.
- Beschreibung - Eine kurze Beschreibung der Aufgaben dieser Netzklasse.
- Typ - Wie aus der Übersicht über die Möglichkeiten der Netznutzung ersichtlicht ist, muß zwischen zwei Typen von Netzklassen unterschieden werden. Netzklassen vom Typ 1 benötigen intern im CSN keine Isolation untereinander und können innerhalb des CSN gleich behandelt werden. Klassen vom Typ 2 dagegen müssen komplett isoliert werden.
- Technische Informationen - Dienen der Abbildung einer Netzklasse auf die Netzstruktur, also die Implementation der Netzklasse auf der Netztechnik-Ebene des CSN. Es muß unterschieden werden, ob zur Kontrolle des Netzzugangs eine zusätzliche Firewall nötig ist oder ob der Netzzugang direkt über den CSN-Router erlaubt wird.
- Zugriffsrechte - Genaue Definition, welche Möglichkeiten der Netznutzung die dieser Klasse zugeordneten Nutzer haben. Hier muß unterschieden werden zwischen den erlaubten Netzwerkprotokollen und der Kontrolle von HTTP-Zugriffsrechten für den externen Zugriff.
- Nutzer - Dieser Netzklasse zugeordnete Nutzer. Es müssen Möglichkeiten bestehen, die Nutzer einer einschränkenden Netzklasse zu informieren, warum sie sich dort befinden und was ihnen erlaubt ist.

2.4. Netzklassen im CSN

2.4.1. Abschätzung der Aufgaben einer Netzklassenverwaltung

Im CSN ist bei einem System der Netzklassenverwaltung zwischen zwei Aufgaben zu unterscheiden.

Die erste Aufgabe ist das Schaffen der Infrastruktur für eine neue Netzklasse und das Einrichten einer Firewall zur Kontrolle der Zugriffsbeschränkungen. Eine Netzklasse muß sich über alle Wohnheime des CSN erstrecken, um eine Zuordnung der Nutzer unabhängig vom Standort zu erlauben. Es muß auch die Möglichkeit zum Löschen einer Netzklasse existieren.

Zur zweiten Aufgabe gehört das Verwalten der Nutzer, also die Zuordnung der Nutzerrechner zu den Netzklassen. Hierbei ist wichtig, daß das Umschalten zwischen den Netzklassen transparent für den Nutzer erfolgt. Es sollen keine Änderungen an der Netzwerkkonfiguration von Endgeräten notwendig sein.

2.4.2. Anforderungen an den Einsatz von Netzklassen

Neben den Eigenschaften der Netzklassen ergeben sich auch Anforderungen, die vor allem durch den praktischen Einsatz in einem grossen Netzwerk entstehen. Die Aspekte liegen hier vorallem auf Sicherheit und Stabilität, das Verwalten der Netzklassen soll ohne Störung des Netzbetriebs funktionieren.

1. Eine Netzklasse muß alle notwendigen Daten beinhalten, um wiederherstellbar zu sein. Im Falle von Rechentechnik-Problemen oder des Einsatzes neuer Technik, müssen die entsprechenden Konfigurationen jederzeit neu geschrieben werden können.

2. Der zusätzliche Aufwand für die CSN Administratoren sollte minimal sein und das Anlegen der Netzklasse nach Auswahl der nötigen Parameter weitgehend automatisch geschehen.

3. Der Prozess des Anlegens/Konfigurierens einer Netzklasse muß transaktionsbasiert sein, damit im Falle von Problemen mit der Rechentechnik wieder ein konsistenter Status hergestellt wird.

4. Es müssen Kontrollmöglichkeiten existieren, ob die Konfigurationen erfolgreich waren.

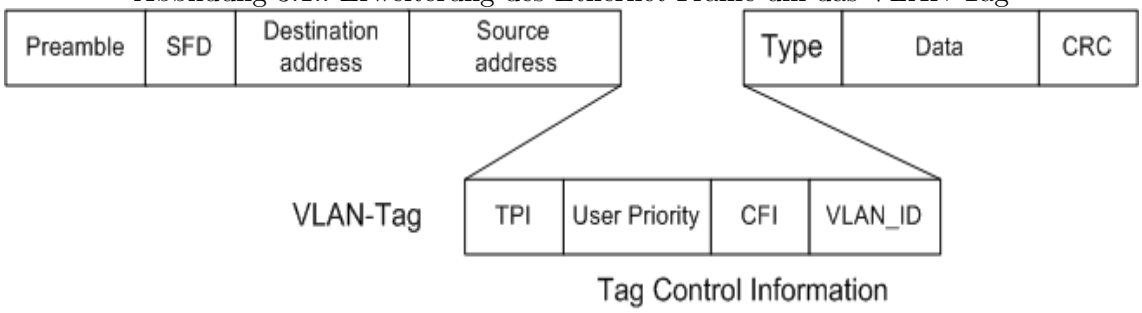
3. Technologische Grundlagen

3.1. Virtual Bridged Local Area Networks - 802.1q

Die Virtual Bridged Local Area Networks (VLAN) haben sich aus einer ganzen Reihe proprietärer und herstellerspezifischer Lösungen entwickelt und wurden 1998 vom Institute of Electrical and Electronics Engineers als IEEE Standard 802.1Q definiert.[3]
Die Aufgabe eines VLAN ist die Trennung der logischen Netzinfrastruktur von der physikalischen Verkabelung, also eine "ortsunabhängige" Zusammenfassung von Rechnern, die miteinander kommunizieren, als ob sie im gleichen Local Area Network (LAN) wären.[5] Mittels VLAN lassen sich über einen oder mehrere Switches hinweg virtuell getrennte Netze betreiben.
VLAN haben die gleichen Eigenschaften wie ein gewöhnliches LAN. Jedes virtuelle Netz stellt eine Broadcast-Domäne dar, es ist kein Multi- oder Broadcasttraffic über die Grenzen des VLAN hinaus möglich. Die Kommunikation innerhalb eines virtuellen Netzes geschieht mittels switching, also in den Switches durch den Vergleich von Ziel-MAC-Adressen der Ethernet Frames und der Forwarding Table. Zur Kommunikation über VLAN Grenzen hinweg muß geroutet werden, der Router muß die VLAN Tags ändern.

Zur Identifikation wird jedem VLAN eine eindeutige Nummer, ein sogenannter VLAN Identifikator (VLAN_ID) zugeordnet. Alle Switchports, denen die gleiche VLAN_ID zugeordnet ist, gehören zu einem virtuellen Netz. Die korrekte Zuordnung der Pakete zu einem VLAN bei der Kommunikation zwischen Switches ist über VLAN Tagging möglich. Dazu wird der OSI-Layer 2 Ethernet Frame Header um ein 4 Byte grosses Feld ergänzt. Man spricht jetzt von einem tagged Paket/Frame.

Abbildung 3.1.: Erweiterung des Ethernet Frame um das VLAN-Tag



Aufbau des VLAN-Tag: [3, 4]

- Tag Protocol Identifier - 16 Bit Ethernet Typ ID für 802.1Q. Mit diesem Eintrag wird ein Paket als tagged markiert
- Tag Control Information hat die folgenden Elemente:

- User Priority - 3 Bit, ermöglicht das Nutzen von bis zu 8 Prioritätsleveln
- Canonical Format Identifier - 1 Bit, Festlegung der Bitorder der Adressinformation im Frame
- VLAN_ID - 12 Bit VLAN Identifikator, dient der Identifizierung zu welchem VLAN dieses Frame gehört

Insgesamt können somit theoretisch 4096 verschiedene VLAN unterschieden werden, da allerdings die VLAN_ID 0 und 4095 reserviert sind, ergeben sich 4094 mögliche virtuelle Netze.[6]

Die Ports, über die VLAN-fähige Switches verbunden sind, werden als "Trunk-Ports" bezeichnet, da auf ihnen mehrere virtuelle Netze anliegen können. Empfängt ein Switch ein tagged Frame, wertet er die VLAN Information aus und sendet das Paket an den entsprechenden Zielport, sofern dieser demselbem virtuellen Netz zugeordnet ist. Handelt es sich bei diesem Port um ein Endgerät, wird das VLAN-Tag entfernt. Versendet ein Endgerät Pakete, dann wird der Switch diesen das VLAN-Tag hinzufügen, um eine eindeutige Identifikation zu ermöglichen.[3, 4]

Für die Zuordnung von untagged Paketen zu VLAN definiert 802.1q-2003 zwei Regeln.[3]
Die "Port Based Tagging Rule" besagt, daß jedes nicht tagged Frame, was von einem Port empfangen wird, dem virtuellen Netz zugeordnet wird, dessen VLAN_ID an diesem Port anliegt.
Bei der "Port and Protocol Based Tagging Rule" erfolgt die Klassifikation durch Auswertung der OSI-Layer 2 bzw. der Protokollinformationen der darüberliegenden Schichten und darüber die Zuordnung eines Pakets zu einem VLAN.
Es ergeben sich mehrere Möglichkeiten der Zuordnung zu virtuellen Netzen, zumeist basierend auf den OSI-Layers:[3, 4]

1. Port Based VLAN / statische VLAN: Jedem physischen Port wird direkt ein virtuelles Netz zugeordnet. Es wird somit durch den Switchport festgelegt, zu welchem virtuellen Netz die angeschlossenen Geräte gehören.

2. MAC Based VLAN / dynamische VLAN: Basiert auf OSI-Layer 2, die MAC-Adressen der angeschlossenen Geräte bestimmen die Zuordnung zum VLAN. Es bedarf einer zentralen Verwaltung aller MAC-Adressen und der Zuordnung zu den virtuellen Netzen. Für dieses Management ist wiederum ein eigenes Protokoll nötig (z.B. Cisco VLAN Trunk Protocol).

3. Protocol Based VLAN: Basiert auf Protokoll-Auswertung der OSI-Layer 3. Dazu gehören z.B. die Zuordnung der VLAN_ID durch Auswertung der Protokolltypen und die Zuordnung ganzer IP-Subnetze in ein VLAN.

Die Möglichkeiten der Klassifikation sind noch wesentlich umfangreicher, z.B. durch Kombinaton einzelner Verfahren.

Es exisitieren auch nicht standardisierte und zumeist herstellerspezifische Verfahren, die den Austausch von VLAN-Informationen zwischen Switches ermöglichen und den Frame-Header nicht verändern. Beim Zeitmultiplexverfahren wird der Backbone zwischen den Switches in Zeitscheiben unterteilt, die fest den einzelnen virtuellen Netzen zugeordnet sind. Eine weitere

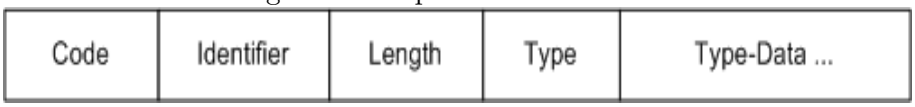
Möglichkeit ist der Austausch von Adresstabellen zwischen den Switches, die um die VLAN-Information für jeden Port ergänzt wurden. Dieser Austausch sollte in kurzen Abständen erfolgen, damit Änderungen schnellstmöglich bekanntgemacht werden.[7]

3.2. Port Based Network Access Control - 802.1x

Der IEEE Standard 802.1x stellt eine generelle Methode für die Autorisierung und Authentifizierung von Endgeräten in Netzwerken bereit.[8]
Die Benutzeridentifikation wird bereits am Netzzugangsport vorgenommen, dies kann ein logischer Zugangsport im WLAN oder ein physischer Port im LAN sein. Der Switchport wird erst freigegeben, wenn der Nutzer authentifiziert ist. Somit stellt 802.1x eine wirksame Methode bereit, den Zugang zum lokalen Netzwerk direkt am Switchport zu begrenzen.
Folgende Elemente werden unterschieden: ein Supplicant, welcher den anzumeldenden Rechner darstellt, der Authenticator, welcher entweder ein Access Point bei der Nutzung von WLAN oder ein Switch im LAN ist, und ein Authentication Server, dieser verwaltet die Authentifizierungsdaten (zB. RADIUS - Remote Authentication Dial In User Service).[8, 9]

Mittels des Extensible Authentication Protocol (EAP) können der Supplicant und der Authentication Server über ein einfaches request/response Verfahren aushandeln, welches Authentifizungverfahren genutzt werden soll. EAP dient der Einkapselung eines beliebigen Authentifizierungsprotokolls.[9, 10]

Abbildung 3.2.: Komponenten einer EAP Pakets



Aufbau der request und response Pakete:[10]

- Code - Gibt den Typ des EAP Pakets an (1 - request, 2 - response)
- Identifier - Dient der eindeutigen Zuordnung der request und response Pakete zueinander.
- Length - Enhält die Gesamtlänge des Pakets.
- Type - Code für das zu nutzende Authentifikationsverfahren (4 - MD5, 13 - TLS, 14 - TTLS). mit dem Code 3 kann dem Kommunikationspartner mitgeteilt werden, daß eine andere Methode genutzt werden soll.
- Type-Data - Feld variabler Grösse und stellt den eigentlichen Inhalt dar.

Der Authenticator selbst kennt keinerlei nutzerspezifische Informationen, sondern leitet einfach die request- und response-Pakete weiter. Die Kommunikation zwischen Supplicat und Authenticator läuft entweder über EAPoW (EAP over Wireless) bei Nutzung von WLAN oder über EAPoL (EAP over LAN).[9]
Verläuft die Authentifizierung des Supplicant erfolgreich, wird eine Nachricht vom Authentication Server zum Authenticator gesendet und dieser schaltet den entsprechenden Port frei.

Hierbei kann übermittelt werden, welchem VLAN der Supplicant zuzordnen ist.

Das Verfahren TLS (Transport Layer Security Protocol) erfordert eine gegenseitige zertifikatbasierte Authentifizierung von Supplicant und Authentication Server.
Bei der Nutzung von TTLS (Tunneled Transport Layer Security) wird mittels Serverzertifikat ein sicherer TLS-Tunnel aufgebaut und der Benutzer authentifiziert sich durch Nutzername und Passwort. Bei dieser Methode ist kein Client-Zertikikat notwendig.[9, 10]

3.3. Simple Network Management Protocol - SNMP

Das Simple Network Management Protocol wurde mit dem RFC 1157 als verbindlicher Standard SNMPv1 definiert. Dieses Protokoll dient der Verwaltung und Überwachung von Netzwerkkomponenten. Als Transportprotokoll für SNMP wird UDP genutzt, vorallem aufgrund der geringen Komplexität, aber nach Definition kann SNMP auch auf anderen Protokollen (z.B. TCP,IPX) eingesetzt werden.[11, 13]

Das Architekturmodell von SNMP unterscheidet zwischen zwei Komponenten, dem Managementsystem und den Agenten. Der Manager ist zumeist eine spezielle Anwendung, die Agenten sind die einzelnen Netzwerkelemente (z.B Switches).
Es werden Management-, Alarm- und Statusinformationen zwischen Netzwerkelementen und Managementsystemen ausgetauscht. Dieser Austausch kann in beide Richtungen erfolgen. In der Regel schickt der Manager eine Request-Message an den Agenten und dieser antwortet mit einer Response-Message.
Im umgekehrten Fall sendet der Agent eine unaufgeforderte Nachricht, eine sogenannte SNMP-Trap, an den Manager. Damit können z.B. unvorhergesehene Ereignisse und Alarmmeldungen dem Managementsystem übermittelt werden.[11, 13]

Die Definition der verfügbaren Informationen der Netzelemente erfolgt in den Management Information Bases (MIB). Diese enthalten die Definitionen der Management-Variablen, welche in einer Hierarchie abgelegt und durch eindeutige Object-Identifier gekennzeichnet sind.[11]

Folgende Befehlsfunktionalität steht dem SNMP zur Verfügung:[11, 12]

- get-request - Dient dem Abrufen der Werte von Variablen. Es können mehrere Variablen mit einem Befehl angefordert werden.
- getnext-request - Dient dem Abrufen des lexikographischen Nachfolgers einer Variablen. Mit diesem Befehl können mehrere Datensätze nacheinander abgerufen werden, sinnvoll für das Auslesen von Tabellen.
- set-request - Dient dem Setzen von Management-Variablen auf neue Werte. Hierbei ist speziell auf den Typ der Variablen zu achten.
- get-response - Dient dem Zurückgeben der Werte von Variablen und wird vom Agenten an den Manager verschickt als Antwort auf get-request, getnext-request und set-request Kommandos.
- trap-request - Dient der Information des Managers über unvorhergesehene Ereignisse.

Abbildung 3.3.: Komponenten einer SNMP Message (ausser trap-request)

Aus folgenden Komponenten besteht eine SNMP-Message:[12]

- Version - Gibt die zu nutzende SNMP Version an (0 - SNMPv1, 1 - SNMPv2, ...)
- Community - Management Community Passwort, zur Absicherung von Zugriffsbeschränkungen
- Typ - Message-Typ: get-request, getnext-request, set-request oder get-response
- Identifikator - Eindeutiger Identifiktor der Message, der zur Abstimmung der request- und response-Message benötigt wird.
- Fehlerstatus - Liefert Nachricht über Erfolg oder Fehler bei SNMP-Operation, sechs verschiedene Werte vordefiniert (0 - NoError, 1 - tooBig, 2 - noSuchName, 3 - badValue, 4 - readOnly, 5 - genErr)
- Fehlerindex - Einzelne Ziffer, welche den Index der Variable bestimmt, wo der Fehler aufgetreten ist.
- Variablen-Bindungen - Folge von Object Identifier und Variablenwerten.

SNMPv1 bietet keine Sicherheitsmechanismen und versendet das Community-Passwort im Klartext im Protokoll. Neuere SNMP-Versionen (SNMPv2p,SNMPv3) erlauben neben einer kryptografischen Absicherung u.a. grössere Informationsmengen pro Nachricht und die Kommunikation zwischen Managern.[12]

3.3.1. VLAN Verwaltung mittels SNMP

Wie aus dem Überblick über die eingesetzte Technik im CSN erkennbar ist, gibt es zwei Typen von Switches, die Geräte von 3com und Cisco. Beide nutzen jeweils Herstellerspezifische MIB zur Konfiguration der Geräte.

3com

Für eine VLAN-Konfiguration auf 3com-Geräten werden die 3CVLAN-MIB und die IF-MIB benötigt. Die VLAN und die 802.1q Kapselung werden intern auf virtuelle Interfaces abgebildet. Die Geräte unterscheiden nicht explizit zwischen Trunk-Ports und normalen Ports.

Das Anlegen eines VLAN besteht aus 3 Schritten.
Mittels der Tabelle a3ComVlanIfTable kann ein neues VLAN initialisiert werden, es müssen die VLAN_ID und der Name gesetzt werden. Hierfür ist der Interface Index eines neuen virtuellen Interfaces notwendig, welchen man sich vom Objekt a3ComNextAvailableVirtIfIndex erzeugen lassen kann, diese Nummer repräsentiert jetzt geräteintern das VLAN.
Im Zweiten Schritt muß ein virtuelles Interface für die 802.1q Kapselung erstellt werden. Dies

geschieht mittels der a3ComVlanEncapsIfTable, in welcher sich die VLAN_ID und die Kapselung 802.1x setzen lässt. Wiederum muß die Nummer eines neuen virtuellen Interfaces erzeugt werden.
Zur Aktivierung des VLAN muß eine Zeile in der ifStackTable erzeugt werden mit der Zuordnung <virtuelles VLAN Interface>.<virtuelles Interface der Kapselung>.
Das Löschen eines VLAN geschieht durch Entfernen der Tabelleneinträge, genau in der umgekehrten Reihenfolge des Anlegens.

Die Zuordnung eines virtuellen Netzes zu Switchports erfolgt ebenfalls in der ifStackTable. Soll das VLAN untagged auf einen Port anliegen, so ist die Zuordnung <virtuelles VLAN Interface>.<Portnummer> notwendig. Wenn das VLAN tagged am Port anliegen soll, muß stattdessen das virtuelle Interface der Kapselung genutzt werden.

Cisco

Die VLAN Konfiguration von Cisco-Geräten erfolgt über die CISCO-VTP-MIB und die CISCO-VLAN-MEMBERSHIP-MIB.
Standardmässig nutzt Cisco das VLAN Trunking Protocol (VTP) zur Propagierung der VLAN-Konfigurationen zwischen den Switches einer Managementdomäne. Diese Funktionalität kann deaktiviert werden.

Zur Konfiguration von VLAN-Einstellungen verwendet Cisco die vtpVlanEditTable als Edit-Buffer. Dieser muß über das Objekt vtpVlanEditOperation aktiviert werden. Danach kann dieser editiert und die entsprechenden Einstellungen (VLAN_ID, Name, Typ) gesetzt werden. Zurückgeschrieben wird der Buffer wiederum über die vtpVlanEditOperation. Im vtpVlanApplyStatus kann geprüft werden, ob die Änderungen erfolgreich in die Running-Config übernommen wurden. Das Löschen eines VLAN geschieht auf gleichem Wege.
Da diese Konfiguration nicht Beim Aktivieren der vtpVlanEditTable wird ein Timer gestartet, ist dieser abgelaufen, bevor die Änderungen zurückgeschrieben wurden, verfallen diese.

Cisco unterscheidet explizit zwischen normalen Ports und Trunk-Ports, also Ports, denen tagged VLAN zugeordnet werden und die sozusagen die Uplink-Ports der Switches darstellen.
Trunk-Ports werden über die vlanTrunkPortTable konfiguriert, ihnen wird eine 1024 Bit grosse Bitmaske zugewiesen, in welcher jedes Bit für eine VLAN_ID steht. Ist ein Bit gesetzt, liegt das entsprechende VLAN tagged am Port an. Gespeichert ist diese Bitmaske als Hexstring.

Normalen Ports, also Ports, an denen Endgeräte hängen, wird über das Objekt vmVlan ein untagged VLAN zugewiesen.

3.4. Kurzbeschreibungen

3.4.1. Iptables

Iptables ist ein Linuxprogramm, welches den Zugriff auf die netfilter Module des Kernels erlaubt. Dies ermöglicht das Erstellen einer komplexen Firewall mittels einer Folge von Befehlen. Iptables ist fähig, verbindungsorientierte Protokolle zu handhaben und unterstützt Network

Address Translation.[14]

Iptables definiert standardmässig 3 Tabellen und ordnet diesen jeweils einige Regelketten zu. Die filter-Tabelle dient der Paketfilterung und verfügt über die Ketten INPUT, FORWARD und OUTPUT. Die nat-Tabelle mit den Ketten PREROUTING und POSTROUTING wird genutzt für Network Address Translation, Masquerading und Redirection. Die mangle-Tabelle dient der "Bearbeitung" von Paketen.[15]

Diese Ketten können durch selbstdefinierte Regeln gefüllt werden, desweiteren können eigene Ketten definiert werden. Eine Regel besteht jeweils aus Bedingungen und einem Ziel. Das Ziel bestimmt, was mit dem IP-Paket geschehen soll, wenn es auf die Bedingungen passt. Einige vordefinierte Ziele:[14]

- ACCEPT - Paket an der aktuellen Kette akzeptieren und zur nächsten Kette weiterleiten
- DROP - Paket verwerfen
- REJECT - Paket verwerfen und den Absender informieren
- RETURN - Paket sofort ans Ende der Kette weiterleiten

Erreicht ein Paket eine Regelkette, wird es nacheinander an den einzelnen Regeln geprüft. Trifft eine Regel zu, bestimmt das Ziel dieser Regel über das Schicksal des Pakets. Passt keine Regel auf das Paket, wird entsprechend der definierten POLICY der Kette das Paket akzeptiert oder verworfen.[13]

Eine ausführliche Einführung zum Thema iptables gibt die Studienarbeit "Firewall mit nutzerspezifischen Regeln" von Heiko Jehmlich. [22]

3.4.2. Dynamic Host Configuration Protocol - DHCP

Das Dynamic Host Configuration Protocol ermöglicht die dynamische Zuweisung von Netzwerkkonfigurationen an Rechner in einem lokalen Netzwerk. Es basiert auf dem Bootstrap Protocol (BOOTP) und erweitert es um neue Parameter.
DHCP ermöglicht die Einbindung neuer Endgeräte in einem bestehenden Netzwerk, ohne auf ihnen eine Konfiguration der Netzwerkparameter vornehmen zu müssen. Es muß lediglich das automatische Beziehen der Netzwerkeinstellungen aktiviert sein, also ein DHCP-Client genutzt werden.[16]

Der DHCP-Server verwaltet die wichtigsten Netzwerkparameter (z.B. Gateway, Netzmaske, DNS-Server und WINS-Server) und teilt diese dem DHCP-Client mit. Befindet sich der DHCP-Server nicht im selben Subnetz wie der Client, ist ein DHCP Relay Agent notwendig, welcher die Nachrichten an einen DHCP-Server weiterleitet und diesem mitteilt, aus welchem Subnetz er die Konfigurationen vergeben soll.[16]

Die Kommunikation läuft standardmässig über die Ports 67 und 68. Ablauf der Adressvergabe: [16, 17]

1. Der DHCP-Client schickt eine DHCPDISCOVER-Nachricht als Broadcast in sein Subnetz. Er kennt seine IP-Adresse noch nicht und setzt das Feld auf "0.0.0.0".

2. Der DHCP-Server antwortet mit einer DHCPOFFER-Nachricht direkt an den Client und schlägt diesem eine IP-Adresse vor.

3. Der Client kann jetzt entscheiden, ob er die vorgeschlagene IP-Adresse annimmt. Falls er die IP-Adresse behalten will, wird eine DHCPREQUEST-Nachricht an den Server gesendet. Dieser bestätigt die Konfiguration durch eine DHCPACK-Nachricht und übermittelt damit weitere Netzwerkparameter.
   Falls der Client die Adresse nicht akzeptiert, sendet er eine DHCPDECLINE-Nachricht an den Server und die Prozedur beginnt von neuem.

4. Bewertung

Ich werde in diesem Kapitel untersuchen, wie die im letzten Kapitel vorgestellten Technologien im CSN eingesetzt werden können. Entscheidendes Kriterium sind die Einschränkungen durch die verfügbare Netztechnik und die Forderung, daß sich die Lösung in die bisherige CSN-Struktur gut integrieren muß.
Von vorne herein lässt sich sagen, daß sich mit den gegebenen Mittels mit Sicherheit nicht die optimale Lösung der Theorie in die Praxis übernehmen lässt. Das Ziel muß sein, die bestmögliche Lösung für die aktuelle Situation zu schaffen. Dabei soll aber auch ein Ausblick auf Möglichkeiten gegeben werden, welche bei ausreichender Ausstattung mit neuer Netzwerktechnik verfügbar sind. Das CSN entwickelt sich ständig weiter.
Am Ende des Kapitels soll feststehen, wie die in Kapitel 2 eingeführten Komponenten der Netzklassen angepasst auf das CSN auszusehen haben.

4.1. Betrachtung möglicher Alternativen

4.1.1. Kommerzielle und Open Source Alternativen

Eine Betrachtung kommerzieller Alternativen (z.B. HP OpenView, Cisco Works) kommt für diese Studienarbeit nicht in Frage, da keine Geldmittel für das Anschaffen von Netzmanagement-Software vorgesehen sind. Das CSN setzt konsequent auf die Nutzung von Open Source Software (OSS) und eigenen Lösungen.

Aus dem Bereich der Open Source Software existieren eine ganze Reihe von Netzwerk-Management-Programmen. Die Studienarbeit “Abstraktion von Managementaufgaben aktiver Netzkomponenten“ von Thomas Kuschel [18] stellt bereits einen guten Überblick über die wichtigsten Eigenschaften der OSS Netdisco, Nedi, Scotti, “Big Brother” und “The Multi Traffic Grapher” (MRTG) dar.

Zu den in Kapitel 2 definierten Aufgaben einer Netzklassenverwaltung gehört als erstes die Schaffung der benötigten Infrastruktur. Hierfür ist die Betrachtung der Konfigurationsfähigkeiten der einzelnen Tools nötig. Dies wurde von Thomas Kuschel bereits unter dem Begriff der “Gerätekonfiguration” zusammengefasst. Die betrachteten Programme verfügen maximal über rudimentäre Fähigkeiten einer Konfiguration von Netzelementen mittels SNMP.
Allgemein lässt sich sagen, daß die betrachteten Programme mehr auf Monitoring, Visualisierung und Überwachung abzielen und keine Alternative für die Konfiguration eines grossen Netzes sind.

Eine Grundfunktionalität für die Konfiguration von Netzelementen mittels SNMP stellt, wie schon beschrieben, die Switch-API des CSN bereit. Eine auf dieser SNMP-Bibliothek basierende eigene Lösung hat ganz klar den Vorteil, daß sie sich bestmöglich in die aktuelle Netztopologie, die Nutzerverwaltung und die Datenbank einpassen kann.

Die Mitarbeiter des CSN sind selbst nur Studenten. Es herrscht ein regelmäßiger Wechsel innerhalb des CSN-Teams, welches die höchste administrative Ebene darstellt.
Dies birgt bei eigenen Lösungen die Gefahr, daß der Weggang von Teammitgliedern einen grossen Wissensverlust bedeutet. Von daher sind eine vernünftige Dokumentation und das rechtzeitige Einarbeiten des "Nachwuchses" unbedingt notwendig.

4.1.2. Netzverwaltung des Rechenzentrums der TU-Chemnitz

Das Rechenzentrum verwaltet das Campusnetz der TU-Chemnitz und bietet umfassende Dienste (z.B. WWW, E-Mail, AFS, FTP, ...) für die Nutzer der Universität an.
Aktuell wird das gesamte Netzwerk auf Basis des Projekts "Campusnetz II" [19] umstrukturiert. Die Ziele sind u.a. das Ersetzen alter Technik, der Aufbau einer ausfallsicheren Topologie im Backbone, eine Verbesserung des Netzmanagements und die Einführung von "Voice over IP" (VoIP).
Es werden mindestens 5 seperate Routingbereiche eingeführt und mittels dynamischen Routing können bei Ausfall einzelner Strecken binnen kurzer Zeit Ersatzstrecken aktiviert werden. Eine Nutzung campusweiter VLAN, wie sie bisher noch existieren, ist nichtmehr möglich.

Die VLAN-Konfiguration der Netzelemente erfolgt aktuell per VLAN Trunking Protocol (VTP), einer Cisco spezifischen Technologie. Diese ermöglicht die Propagierung von VLAN Informationen zwischen den Switches, ohne jeden einzelnen konfigurieren zu müssen. Unterschieden werden muß zwischen Servergeräten, welche die VLAN Einstellungen verteilen und Clientgeräten. Der Server mit der grössten Revisionsnummer stellt die initiale Konfiguration bereit.
Desweiteren wird das kommerzielle Netzmanagementprogramm HP Openview eingesetzt, welches aber mehr zum Monitoring genutzt wird.

VTP bietet die Möglichkeit, eine VLAN-Konfiguration zentral zu verwalten und automatisch durch ein Netzwerk zu verteilen. Die Nutzung von VTP im CSN ist zur Zeit nicht möglich, da dieses Protokoll nur von den Geräten der Marke Cisco unterstützt wird und die Geräte vom Typ 3com nicht damit umgehen können. Desweiteren birgt die Nutzung von VTP neue Risiken. Ein einziges falsch eingestelltes Netzelement kann die gesamte netzweite VLAN-Konfiguration zerstören. Es muß nur als VTP Server konfiguriert sein und eine höhere Stelligkeit als der aktuelle Server haben. Somit verteilen sich die möglicherweise fehlerhaften VLAN-Einstellungen über das ganze Netz.
Da im CSN öfters Technik per Hand getauscht wird, sei es nur zu Versuchszwecken, ist dies eine nicht zu unterschätzende Gefahr. Zusätzlich dazu entsteht generell ein höheres Trafficaufkommen zwischen den Netzelementen zwecks Verteilung der Konfigurationen.

4.2. Einsatzmöglichkeiten der Technologien im CSN

4.2.1. Port Based Network Access Control

Durch die Benutzeridentifikation direkt am Zugangsport der Switches stellt 802.1x ein für das CSN interessantes Verfahren zur Kontrolle des Netzzugangs bereit.
Bisher wird Portsecurity zur Beschränkung des Netzzugangs eingesetzt. Dieses Verfahren ist jedoch sehr anfällig, da nur auf eine gültige MAC geprüft wird. Liegt eine nicht auf dem

Switchport angemeldete MAC an, wird dieser geschlossen. Mittels 802.1x könnte aufgrund der Benutzeridentifikation eine wesentlich bessere Absicherung des Netzzugangs erreicht werden. Zusätzlich dazu könnte direkt vom Authentication Server festgelegt werden, in welches VLAN der Nutzerport zu schalten ist.

Das Hauptproblem hierbei ist wieder die aktuell im CSN verfügbare Technik. Die eingesetzten Switches vom Typ 3Com arbeiten nicht mit 802.1x. Desweiteren existiert kein Authentication Server im Studentennetz.
Es ist eine Software auf dem Supplicant, also dem anzumeldenden Rechner, erforderlich. Der Microsoft 802.1x Supplicant, welcher bei Windows XP schon standardmässig integriert ist, steht nicht für alle älteren Versionen von Microsoft Windows bereit. Dies ist ebenfalls ein problematischer Punkt.

Nichtsdestotrotz stellt 802.1x in Kombination mit der VLAN-Technologie eine durchaus interessante Alternative für eine zukünftige Kontrolle des Netzzugangs im CSN dar.

4.2.2. Virtual Bridged Local Area Networks

Die Technologie 802.1q bietet die Möglichkeit, ohne zusätzlichen Hardwareaufwand weitere virtuelle Netze zu schaffen, welche sich über mehrere Switches erstrecken können. Einzelne Switchports können unterschiedlichen virtuellen Netzen zugewiesen werden, und somit kann eine neue Netzstruktur aufgebaut werden, ohne jegliche Änderung der physikalischen Verkabelung oder des Einsatzes zusätzlicher Hardware.
Alle im CSN eingesetzten Netztechnik-Geräte sind fähig, mit dem Standard 802.1q zu arbeiten und wie im Kapitel 1 angedeutet werden virtuelle Netze im CSN bereits genutzt. Dies sind ein “Management-VLAN”, über welches die Steuerung der Netzwerktechnik erfolgt, und pro Wohnheim ein “Haus-VLAN”, dem alle Nutzerrechner des Hauses zugeordnet sind.

Tabelle 4.1.: Übersicht über die wichtigsten VLAN_IDs im CSN

Wohnheim	“Haus-VLAN” ID	“Management-VLAN” ID	Native VLAN ID
Reichenhainer Str. 35	3	1	12
Reichenhainer Str. 37	2	1	12
Reichenhainer Str. 51	114	1	12
Vettersstrasse 52	93	103	103
Vettersstrasse 54	9	1	12
Vettersstrasse 64	7	1	12
Vettersstrasse 66	8	1	12
Vettersstrasse 70	6	1	12
Vettersstrasse 72	5	1	12
Thüringer Weg 3	4	1	12

Das VLAN 12 ist ein "Ersatz-VLAN", über welches kein Traffic läuft. Es dient lediglich zur Interoperabilität zwischen den Geräten der verschiedenen Typen, da auf den Cisco-Switches immer mindestens ein virtuelles Netz untagged anliegen muß.
Die Nutzung des VLAN 1 als "Management-VLAN" ist nicht optimal, da bei Fehlern im VLAN-Tag ein Paket automatisch dem VLAN 1 zugeordnet wird und somit ein "Überspringen" dieser Grenzen ermöglicht. Der Traffic zum Steuern der verfügbaren Geräte der Marke 3com ist jedoch nur über VLAN 1 möglich, da auf diesem virtuellen Netz die IP-Adresse des Gerätes standardmässig liegt und sich nicht in ein anderes VLAN verlegen lässt.
Das "Management-VLAN" dient auch der "Absicherung" der Kommunikation mit den Netzelementen, da bei der genutzten SNMP Version 1 die Community-Strings im Klartext im Protokoll stehen. Diesem VLAN sind keine Nutzerrechner zugeordnet.
Eine detaillierte Problembeschreibung kann [2] entnommen werden.

Trotz dieser Nachteile der aktuellen Konfigurationen ist die Technologie 802.1q für das CSN die beste Möglichkeit, weitere virtuelle Netze ohne grossen Aufwand zu erzeugen und zu verwalten. Aus der Übersicht der eingesetzten Technik im CSN in Kapitel 1 ist erkennbar, daß der größte Teil der eingesetzen Switches vom Typ 3com1100-3300 sind. Somit bestimmen die Möglichkeiten dieser Geräte die Art des Einsatzes von virtuellen Netzen.
Dieser Switchtyp kann intern gleichzeitig bis zu 16 verschiedene VLAN verwalten. Dies ist eine sehr geringe Anzahl und stellt eine starke Einschränkung für eine umfassende Nutzung von VLAN bei der Netzklassenverwaltung dar. Es muß darauf geachtet werden, sparsam mit der Anzahl der virtuellen Netze umzugehen, um Platz für möglichst viele Netzklassen zu lassen.

Wie bereits beschrieben, existiert in jedem Wohnheim genau ein "Haus-VLAN", welches sich über alle Netztechnikgeräte des Wohnheims erstreckt. Bei der Planung weiterer virtueller Netze ist somit eine Beibehaltung der Unterteilung nach Wohnheimen sinnvoll, da aufgrund der zentralisierten Netzstruktur alle Wohnheime am CSN-Router zusammenlaufen. Ein weiterer Vorteil wäre eine problemlose Integration der bereits genutzten virtuellen Netze in das Netzklassenkonzept, ohne Änderungen an der Netztechnik vornehmen zu müssen.
Dies würde ebenfalls genug Spielraum für eine ausreichend große Anzahl an Netzklassen lassen. Nach den Möglichkeiten der Netznutzung aus Kapitel 2 würden sich mindestens 5 verschiedene Netzklassen ergeben (CSN-Vollzugriff Netz, CSN-Light Netz, Anmeldenetz, Quarantänenetz, CSN-WLAN Netz und das Managementnetz). Dies würde pro Haus nur 6 VLAN_IDs belegen und ermöglicht somit noch genügend Platz für weitere Klassen.
Nach bisherigen Erfahrungen mit Trafficaufkommen bezüglich Broad- und Multicasts in hausweiten VLAN sind gut, es entsteht keine übermässige Belastung in den Netzen.

In Kapitel 2 wird auch gefordert, daß die Nutzer unabhängig von Ihrem aktuellen Standort einer Netzklasse zugeordnet werden können müssen. Somit muß sich eine Netzklasse über alle Wohnheime des CSN erstrecken und benötigt in jedem dieser Häuser eine VLAN_ID.
Aufgrund der zentralisierten Netzstruktur des CSN muß bei der Vergabe der VLAN_IDs berücksichtigt werden, daß diese im ganzen Netz gelten.

4.3. IP-Adresskonfiguration

Wie im Kapitel 1 schon beschrieben, bekommt jeder Nutzerrechner bei der Anmeldung eine feste IP-Adresse aus dem Adressraum des CSN zugeteilt. Der Nutzer hat die Möglichkeit, die Netzwerkeinstellungen entweder statisch einzustellen oder dynamisch per DHCP zu konfigurieren.
Dies ergibt für die Verwendung von Netzklassen ein Problem, denn Rechnern mit statisch eingesteller Konfiguration können keine neuen Adressen zugewiesen werden. Eine nicht geringe Anzahl an Nutzerrechnern ist nach aktuellen Schätzungen statisch konfiguriert. Das Erzwingen der Nutzung von DHCP zur Netzwerkkonfiguration ist aktuell nicht möglich.

In Kapitel 2 wird gefordert, daß das Umschalten zwischen den Netzklassen für den Nutzer transparent zu erfolgen hat. Somit muß eine allgemeine Lösung angestrebt werden, die eine Verwendung statischer Konfigurationen erlaubt. Dies kann nur erreicht werden durch ein Beibehalten der dem Rechner fest zugeordneten IP-Adresse, egal in welcher Netzklasse er sich befindet.
Netzklassen, welchen am CSN angemeldete Rechner zugeteilt werden sollen, müssen somit die gleichen IP-Subnetze pro Haus bilden, wie sie aktuell in den "Haus-VLAN" eingestellt sind. Der IP-Gateway eines VLAN muß entsprechend konfiguriert werden.

Dabei ist zu beachten, daß VLAN an Nutzerports untagged anliegen müssen, also das VLAN-Tag in jedem Ethernetframe vom Switch entsprechend der Senderichtung hinzugefügt bzw. entfernt werden muß.

4.4. Wohnheime Vettersstrasse 64/66

Die Wohnheime Vettersstrasse 64/66 werden bei der weiteren Betrachtung nicht berücksichtigt, da aufgrund der aktuellen Technikausrüstung ein Einsatz der vorgeschlagenen Technologien nicht möglich ist.
Aufgrund der BNC-Verkabelung ist eine Konfiguration für einzelne Nutzerports nicht vernünftig machbar, das praktisch einzige konfigurierbare Gerät ist der Hausswitch der beiden Wohnheime. Eine Isolation der Netzklassen untereinander kann bei der vorhandenen Technik nicht sinnvoll funktionieren. Dem Nutzer bleiben zu viele Ausweichmöglichkeiten.

Theoretisch könnte man die Virtual Private Network Technologie (VPN) nutzen und somit unabhängig von der zugrundeliegenden Verkabelung einen sicheren und nutzerbasierten Netzugang über einen VPN-Concentrator schaffen. Praktisch ist dies nicht gewollt, da durch Nutzung von VPN neuer Overhead in einem Netz entsteht, das im Vergleich zu den anderen Wohnheimen schon langsam ist. Außerdem will man dem Nutzer die explizite Nutzung einer solchen Client-Software nicht vorschreiben.

Diese Entscheidung wurde im Konsens mit den Mitgliedern des CSN-Team getroffen, da der Zeitplan für die Aufrüstung der Vettersstrasse 64/66 auf Twisted-Pair Verkabelung feststeht und diese beiden Wohnheime dann problemlos in das Netzklassen-Konzept integriert werden können.

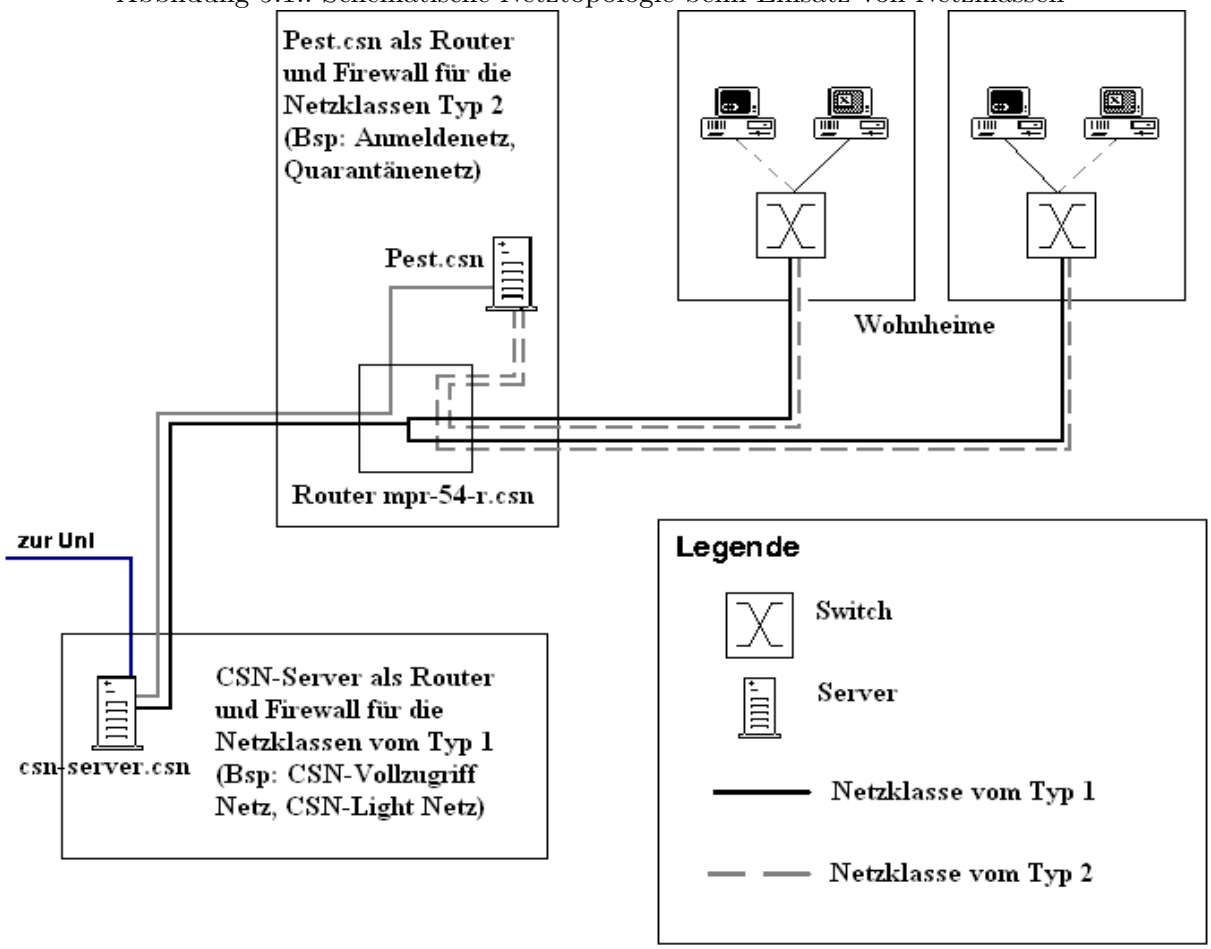
5. Konzept der eigenen Implementation

In diesem Kapitel werde ich meine eigene Implementation darstellen. Zuerst wird ein allgemeiner Überblick über die Struktur gegeben und danach einzelne wichtige Teile konkreter erläutert. Besonderes Augenmerk liegt auf dem Erfüllen der in Kapitel 2 definierten Anforderungen, die durch den Einsatz von Netzklassen in einem grösseren Netz entstehen.
Die zur Implementation gehörenden Programme sind soweit wie möglich in Perl geschrieben, da diese Skriptsprache zum "Standard" innerhalb des CSN erklärt wurde.

5.1. Netzklassenbasierte Topologie

Durch den Einsatz von Netzklassen ergibt sich eine virtuelle Veränderung der Netztopologie.

Abbildung 5.1.: Schematische Netztopologie beim Einsatz von Netzklassen



Bisher wurde aller Traffic standardmässig nur über den Rechner csn-server.csn geroutet und auf diesem von einer Firewall kontrolliert. Zusätzlich zum csn-server.csn kommt jetzt der Rechner pest.csn hinzu, welcher die Aufgaben des Routers und der Firewall für einen Teil der Netzklassen übernimmt.

Bereits in Kapitel 2 wurde zwischen 2 Typen von Netzklassen unterschieden, die sich aus den Beschränkungen und Konfigurationen der Netznutzungsmöglichkeiten ergeben.

Zum ersten Typ gehören alle die Netzklassen, die auf der bisherigen Netztopologie des CSN arbeiten, also denen auf Wohnheim-VLAN-Ebene Interfaces des CSN-Routers als Hausgateways zugeteilt werden und dieser Traffic dann zum csn-server.csn geroutet wird.
Hierbei lassen sich als Basis die bereits existierenden "Haus-VLAN" nutzen, ein Anlegen neuer VLAN beim Erstellen neuer Netzklassen dieses Typs ist nicht notwendig. Es existiert hier keine Isolation der Netzklassen untereinander, Rechner dieser Netzklassen haben innerhalb des CSN Zugriff aufeinander. Lediglich an der Firewall auf dem csn-server.csn wird der Traffic klassenabhängig bewertet.
Beispiele wären das CSN-Vollzugriff und das CSN-Light Netz. Beide nutzen intern im CSN die gleichen VLAN und haben somit Zugriff aufeinander, aber mittels Firewall wird der externe Zugriff für die Nutzer des CSN-Light Netz beschränkt.

Der zweite Typ umfasst alle Netzklassen, bei denen eine Isolation notwendig ist. Es müssen für jede Netzklasse eine eigene virtuelle Netzstruktur und somit die entsprechenden VLAN angelegt werden, welche sich ebenfalls jeweils über ein Wohnheim erstrecken. Der Rechner pest.csn dient als Firewall und Router für diesen Typ. Ein Überschreiten der Grenzen innerhalb des CSN ist für die Nutzer dieser Netzklassen nichtmehr möglich. An der Firewall wird der Traffic ebenfalls klassenabhängig bewertet und es steht ein transparenter HTTP-Proxy zur Kontrolle der HTTP-Zugriffe zur Verfügung.

Tabelle 5.1.: Typen von Netzklassen

Parameter	Netzklasse Typ 1	Netzklasse Typ 2
Netztopologie	gemeinsame Topologie	jeweils eine eigene virtuelle Topologie
Isolation	keine Isolation untereinander	komplette Isolation von anderen Netzklassen
IP-Adressraum	offizielle CSN IP-Adressen	eingeschränkt wählbar
Firewall	klassenabhängig	klassenabhängig
HTTP-Proxy	nein	ja
Beispiele	CSN-Vollzugriff Netz, CSN-Light Netz	Anmeldennetz, Quarantänenetz

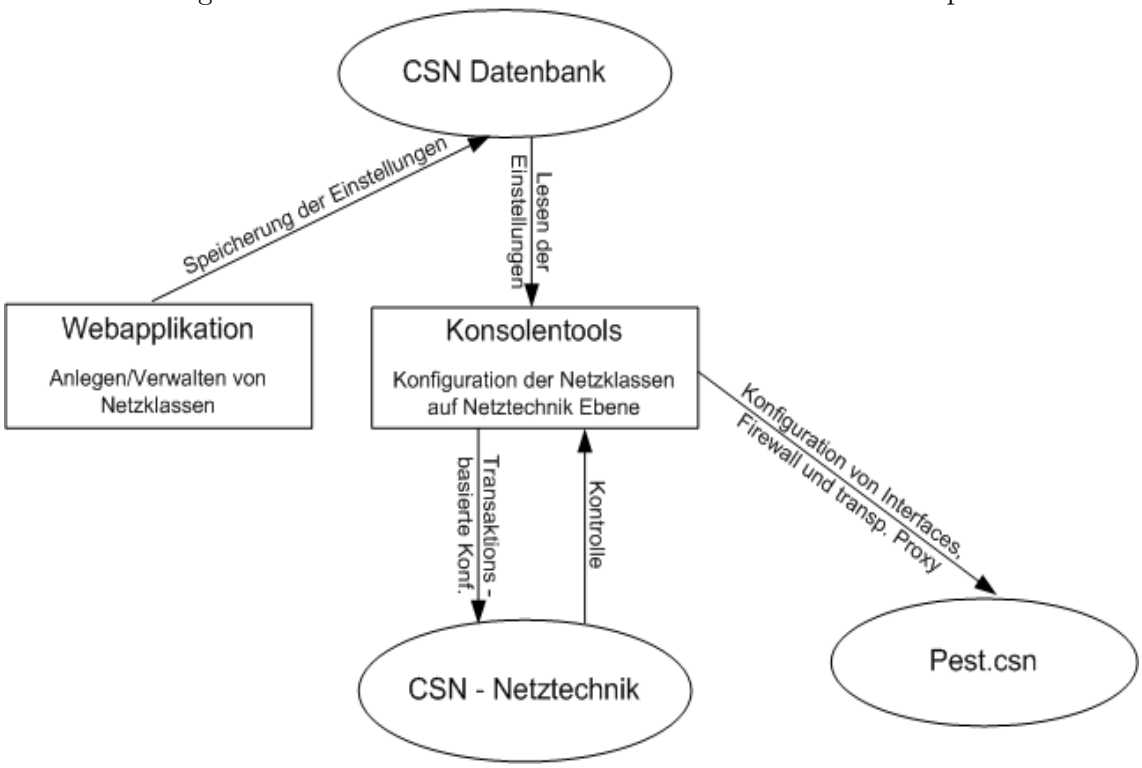
Mit der Nutzung gemeinsamer VLAN für verschiedene Netzklassen des Typ 1 ergeben sich Möglichkeiten für die Nutzer, Netzklassengrenzen zu überspringen. Ein Beispiel hierfür wäre die Nutzung eines HTTP-Proxy Servers, der auf einem Rechner mit CSN-Vollzugriff läuft, durch einen CSN-Light Nutzer.
Bei der Anlage neuer Netzklassen ist abzuwägen, ob es sich um eine starke Einschränkung

der Netznutzungsmöglichkeit handeln soll und somit eine Isolation und stärkere Kontrolle der Zugriffsbeschränkungen notwendig ist, oder ob CSN intern ein uneingeschränkter Zugriff gestattet sein soll.

Ein weiterer wichtiger Faktor ist der IP-Adressraum einer Netzklasse. Wie bereits in Kapitel 4 beschrieben, muß ein im CSN angemeldeter Rechner seine IP-Adresse behalten, egal welcher Netzklasse er zugeordnet ist. Weiterhin existieren Möglichkeiten der Netznutzung, welche im CSN nicht angemeldeten Rechnern offen stehen sollen (z.B. Anmeldenetz, CSN-WLAN Netz).
Somit ist für jede Netzklasse festzulegen, ob der bisherige IP-Adressraum des CSN genutzt werden soll oder ob ein eigener neuer Adressraum definiert wird.
Den Netzklassen vom Typ 1 zugeordnete Rechner müssen am CSN angemeldet sein und die offiziellen CSN IPs nutzen. Es findet keine Adressübersetzung statt. Bei Netzklassen vom Typ 2 ergibt sich die Möglichkeit zwischen offiziellen und privaten IP-Adressen zu wählen. Dies ist davon abhängig, ob dieser Klasse angemeldete Rechner zugeteilt werden sollen oder nicht. Bei Nutzung öffentlicher Subnetze bleibt es dem Nutzer somit überlassen, ob er seinen Rechner statisch konfiguriert oder dynamisch mittels DHCP. Für private Subnetze herrscht DHCP Pflicht.

5.2. Komponenten der Implementation

Abbildung 5.2.: Übersicht des Zusammenwirkens der einzelnen Komponenten



Die Webapplikation soll einen bequemen Weg bieten, neue Netzklassen anzulegen bzw. bestehende Netzklassen zu modifizieren. Genutzt wird hierzu der Wizard aus der CSN-Webseiten-API. Dieser ermöglicht es, dynamisch generierte Webseiten nacheinander abzuarbeiten und in jedem Schritt zu prüfen, ob die benötigten Parameter korrekt gesetzt sind.
Eine Netzklasse wird in folgenden Schritten angelegt: Name und Beschreibung der Netzklasse, Auswahl der VLAN_IDs pro Haus, HTTP-Zugriffsrechte, Grundregeln der Iptables Firewall. Die getroffenen Einstellungen werden anschliessend in der CSN-Datenbank gespeichert.

Die eigentliche Konfiguration der Technik des CSN erfolgt durch einige Konsolentools. Diese beziehen die benötigten Daten aus der CSN-Datenbank. Entsprechend der getroffenen Auswahl zu den VLAN-Einstellungen der Häuser, werden jetzt die Switches der einzelnen Häuser vollautomatisch konfiguriert. Dies soll transaktionsbasiert geschehen, damit bei Konfigurationsfehlern immer ein konsistenter Status wiederhergestellt werden kann.
Es sind Skripte bereitzustellen, welche die Firewall, den transparenten Proxy Squid und die den Netzklassen zugeordneten Interfaces auf der pest.csn konfigurieren.

Die CSN-Datenbank ist um einen Satz von Tabellen und Relationen zu ergänzen. Es müssen die Eigenschaften von Netzklassen abgespeichert und die bisher zu den Nutzern gespeicherten Daten ergänzt werden, um eine Zuordnung des Nutzers zu einer Netzklasse zu ermöglichen.

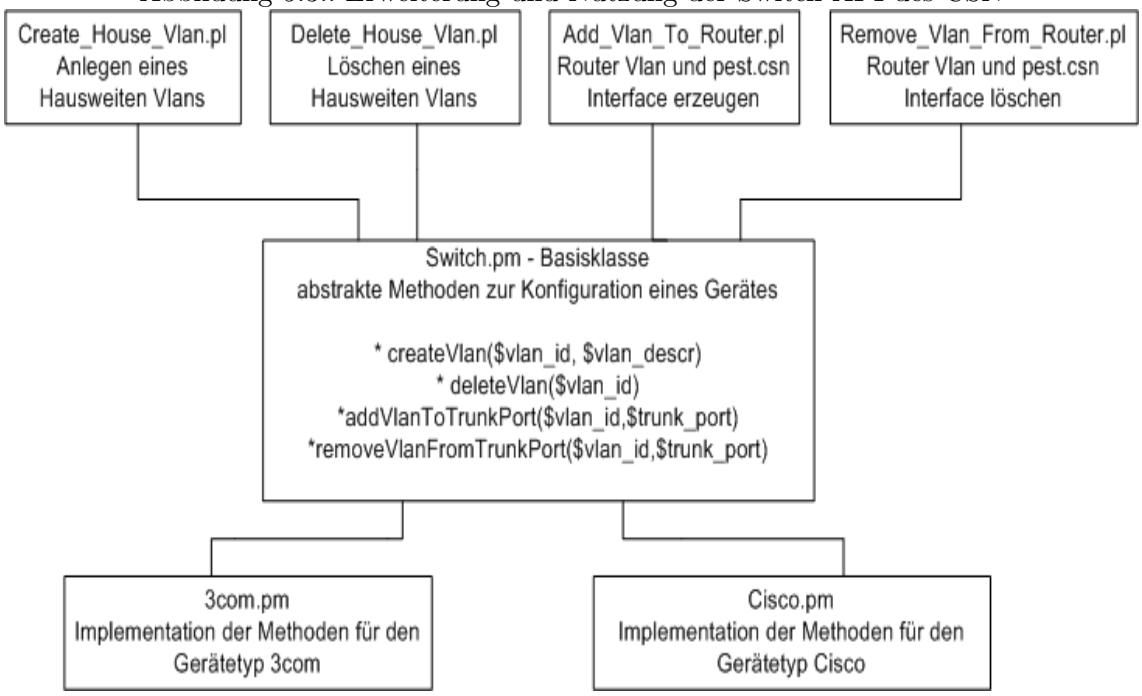
5.3. Konfiguration der Netztechnik

Die Konfiguration der virtuellen Netze soll auf Hausebene erfolgen, also sind Skripte erforderlich, die über allen Switches eines Hauses arbeiten. Ebenfalls muß das VLAN am Router konfiguriert werden. Ss sinnvoll, die Funktionalität in die beiden Schritte der Konfiguration eines Wohnheims und der Konfiguration des Routers und der pest.csn zu zerlegen.

Wie bereits im Kapitel 1 erklärt, erfolgt ein Teil der Konfiguration der Switches durch die Switch-API des CSN. Diese steuert grundlegende Portverwaltung für die beiden Switchtypen Cisco/3com mittels SNMP.
Diese API muß erweitert werden um benötigte Funktionalität für eine VLAN-Verwaltung pro Gerät. Es müssen Methoden zum Erzeugen / Löschen von VLAN und Konfiguration der Trunk-Ports geschaffen werden. Eine Kurzübersicht der Methoden dieser API befindet sich im Anhang.

Es wird jeweils ein Skript für das Anlegen und das Löschen eines VLAN einer Netzklasse auf Hausebene und das Konfigurieren des Routers benötigt. Die für das Management von Netzklassen geforderte transaktionsbasierte Absicherung der Netzwerk-Konfigurationsschritte wird im Abschnitt 5.3.2 erklärt.

Abbildung 5.3.: Erweiterung und Nutzung der Switch-API des CSN



Folgende Tabellen geben einen Überblick über die Aufgabe, notwendige Parameter und Funktionsweise der Skripte.

Tabelle 5.2.: Create_House_Vlan

Aufgabe	Erzeugt ein VLAN einer Netzklasse auf Hausebene.
Parameter	VLAN_ID und Beschreibung des neu zu erzeugenden virtuellen Netzes. Wohnheim-Struktur-Datei, erzeugt bei der Erkennung der Netztopologie (siehe Abschnitt 5.3.1).
Vorbedingungen	VLAN_ID auf allen Geräten des Wohnheims noch frei und noch nicht maximale Anzahl VLAN erreicht.
Funktionsweise	Anlegen des virtuellen Netzes mit der VLAN_ID und konfigurieren der in der Wohnheim-Struktur-Datei genannten Uplink-Ports auf allen Geräten eines Hauses.

Tabelle 5.3.: Delete_House_Vlan

Aufgabe	Entfernt ein VLAN einer Netzklasse auf Hausebene.
Parameter	VLAN_ID des zu löschenden virtuellen Netzes.
Vorbedingungen	Das VLAN exisitiert und darf nicht untagged auf Ports des Hauses anliegen. Es müssen vor dem Löschen eines VLAN alle Nutzerports aus diesem entfernt werden.
Funktionsweite	Löschen des virtuellen Netz mit der VLAN_ID von allen Switches eines Hauses und aus allen Trunk-Ports in denen es gesetzt ist.

Tabelle 5.4.: Add_Vlan_To_Router

Aufgabe	Erzeugt ein VLAN auf dem CSN-Router und konfiguriert ein entsprechendes Interface auf der pest.csn.
Parameter	VLAN_ID und Router Trunk-Ports auf denen das VLAN gesetzt werden soll.
Vorbedingungen	VLAN_ID auf Router noch frei und noch kein Interface auf pest.csn konfiguriert.
Funktionsweite	Legt virtuelles Netz mit VLAN_ID auf Router an, konfiguriert es auf die Trunk-Ports und erzeugt mittels vconfig ein VLAN-Interface auf der pest.csn.

Tabelle 5.5.: Remove_Vlan_From_Router

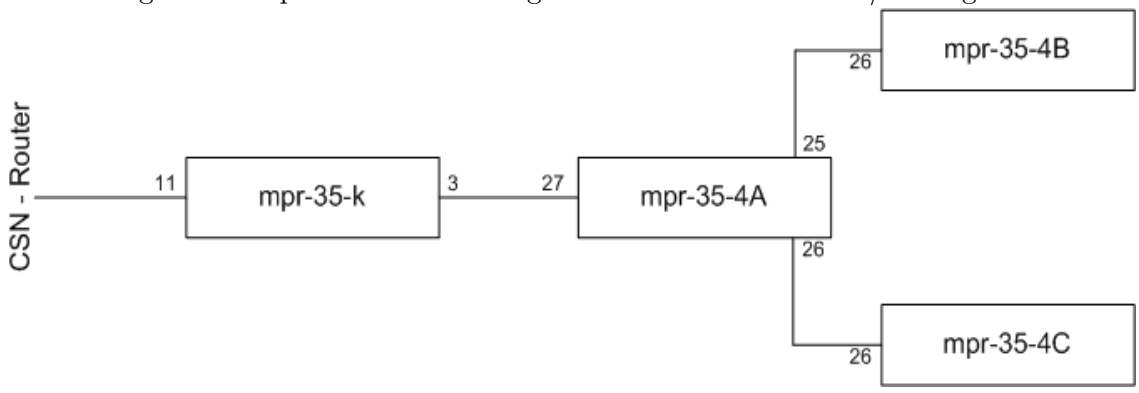
Aufgabe	Entfernt ein VLAN vom CSN Router und löscht das dazugehörige Interface der pest.csn.
Parameter	VLAN_ID.
Vorbedingungen	Das VLAN exisitiert auf dem Router und als Interface auf der pest.csn.
Funktionsweite	Entfernt das virtuelle Netz aus allen Trunk-Ports vom Router und löscht es. Löscht das entsprechende Interface auf pest.csn.

5.3.1. Erkennung der Netztopologie des CSN

Zur vollautomatischen VLAN Konfiguration der Switches ist die genaue Kenntnis der Netztopologie des CSN notwendig, vorallem der Verkabelungen der Switches untereinander in den einzelnen Wohnheimen. So müssen z.B. die Uplinkports dem jeweiligen VLAN der Netzklasse zugeordnet werden, damit der Traffic korrekt weitergeleitet werden kann.

Diese Daten sind bisher nur unvollständig oder gar nicht in irgendeiner Form erfasst worden. Die Konfiguration der Nutzerports erfolgt zwar automatisch per SNMP mit Daten aus der Datenbank, aber dort sind nur Ports erfasst, denen aktuell Endgeräte zugeordnet werden.

Abbildung 5.4.: Beispiel der Verkabelung der 3com Switches der 4./5. Etage der Rh35



Der Ansatzpunkt hierfür ist die dot1d Forwarding Database, in welcher der Switch den Ziel-MAC der Ethernet Frames die Ports zuordnet. Diese Informationen gewinnt er durch Auswertung der eintreffenden Frames an den einzelnen Ports, indem die Quell MAC-Adresse diesem Switchport zuordnet wird. Somit weiss der Switch, daß ein Frame mit der MAC als Zieladdresse über diesen Port weitergeleitet werden muß.
Die Netztopologie eines Wohnheims kann ermittelt werden, indem diese Tabelle per SNMP ausgelesen und nach den bekannten MAC-Adressen der Switches des Hauses und des Routers gesucht wird. Somit erhält man für jeden Switch die Informationen, von welchen Ports er Pakete andere Netztechnik Geräte empfangen hat. Damit lässt sich eine Switchport genaue Topologie abbilden.
Weiterhin ist zu beachten, daß die Switches für jedes VLAN eine Forwarding Tabelle anlegen. Am sinnvollsten ist es, die direkt zum "Management-VLAN" gehörenden Tabellen auszuwerten, da diese relativ klein sind. Die gesamten Nutzerrechner sind diesem VLAN nicht zugeordnet und tauchen in dieser spezifischen Forwarding Table nicht auf. Dabei muß allerdings beachtet werden, daß das "Management-VLAN" nicht in jedem Wohnheim das Default-VLAN 1 ist. Es erfolgt sowohl eine Ausgabe der erkannten Struktur auf dem Bildschirm, als auch als Textdatei, welche als Eingabe für die auf Hausebene operierenden Skripte dient. Dies ist die sogenannten Wohnheim-Struktur-Datei.

5.3.2. Absicherung durch Transaktionen

Das Konfigurieren der VLAN auf den Switches muß nach der Anforderung an das Netzklassen-Management transaktionsbasiert ablaufen. Es muß im Falle des Auftretens von Fehlern sicher gestellt sein, daß immer ein konsistenter Status vorhanden ist.
Nach der Wikipedia-Definition ist eine Transaktion "eine Folge von Verarbeitungsschritten, die entweder als ganzes oder garnicht ausgeführt werden. Transaktionelle Änderungen an Datenbeständen dürfen erst dauerhaft gespeichert werden, wenn die Transaktion vollständig durchlaufen ist".[20]

Es ist mit Nutzung von SNMP zur Konfiguration von Netztechnik nicht möglich, zuerst alle Schritte nur zu testen und erst bei Erfolg die eigentliche Konfiguration vorzunehmen. Man ist gezwungen, einen Schritt nach dem anderen auszuführen und jedesmal auf Erfolg der aktuellen Operation zu testen. Dabei können eine ganze Reihe von Fehlern auftreten:

- SNMP Timeout - Switch antwortet nicht in vorgegebener Zeit auf SNMP-Befehl (Mögliche Ursachen: Switch zu stark belastet; Switch ist ausgefallen; Es wird gerade in der Konfiguration des Switches geschrieben)
- Fehler im SNMP Paket - Das SNMP Protokoll basiert auf dem unzuverlässigen Transportprotokoll UDP.
- Zugriff auf nicht existente Object Identifier.
- Übermittlung falscher Werte bzw. Datentypen in SNMP-Set Operationen

Bei der Netzklassenkonfiguration kann man zwischen zwei Arten von Fehlern unterscheiden, den anwendungs- und den technikbedingten Fehlern.
Die anwendungsbedingten Fehler umfassen all die Fehler, die durch mangelnde Sicherheitsabfragen von der Anwendung selbst ausgelöst werden. Zum Beispiel Operationen zum Anlegen

eines VLAN, obwohl selbiges schon existiert auf dem Gerät. Diese Fehlerart lässt sich weitgehend vermeiden durch genügend Abfragen vor Ausführung von SNMP-Operationen. Dazu gehören die in diesem Kapitel definierten Vorbedingungen der Skripte Create_House_Vlan, Delete_House_Vlan, Add_Vlan_To_Router und Remove_Vlan_From_Router. Hinzu kommen eine Menge weiterer, vorallem Geräte-Typ spezifischer Abfragen.

Technikbedingte Fehler (z.B. Timeout, Fehler im SNMP-Paket) lassen sich von von nicht vornerein abgefangen, denn sie treten während der Ausführung von SNMP-Operationen auf. Hierbei ist wichtig, daß die genutzte SNMP-Bibliothek solche Fehler vernünftigt behandelt und mit einer Fehlermeldung zurückkehrt, anstatt das Programm abzubrechen. Diese Anforderung erfüllt die unter Perl genutzte snmp-lib, indem sie bei Problemen einen String mit der Fehlermeldung füllt.

Perl selbst verfügt über kein standardisiertes Exception-Management, also Möglichkeiten Ausnahmen zu erzeugen und diese dann zu behandeln. Somit musste ein eigenes System implementiert werden, was den Erfolg von Operationen prüft, diese an übergeordnete Funktionen zurückgibt und Fehler behandelt. Diese Anforderung erfüllt die unter Perl genutzte snmp-lib, indem sie bei Problemen einen String mit der Fehlermeldung füllt.

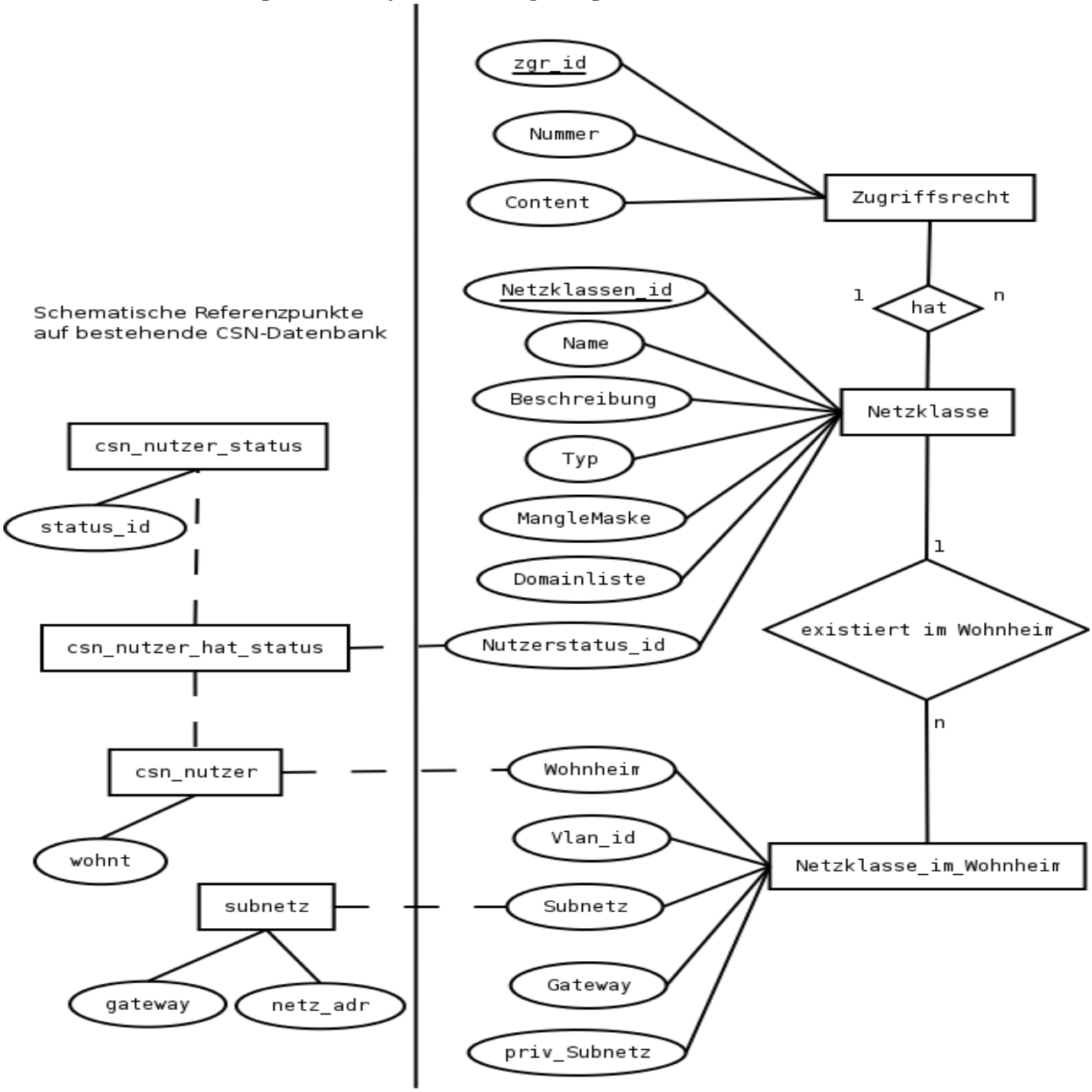
Damit dieses System funktionieren kann, muß zu jedem Zeitpunkt des Konfigurierens der CSN-Technik die Information vorliegen, welche Schritte schon erfolgt sind und wie der aktuelle Status ist. Nur so besteht die Möglichkeit, alle bisher ausgeführten Schritte rückgängig zu machen.
Dieses System beruht auf drei Mechanismen:

- Logfile - Es werden Meldungen über einzelne Schritte (VLAN anlegen, Port konfigurieren, ...) direkt in ein Logfile geschrieben, damit der Ablauf der Operationen besser überwacht werden kann. Fehler selbst werden direkt von der Funktion ausgegeben, wo sie auftreten. Diese Funktion beendet sich danach und gibt einen Fehlercode zurück.
- Rückgabewerte - Jede statusändernde Funktion gibt als ersten Rückgabewert einen Fehlercode zurück. Somit kann jede übergeordnete Funktion prüfen, ob die Operation erfolgreich war. Ist dies nicht der Fall, gibt sie den Fehlercode einfach weiter. Dieser landet dann zwangsläufig in der Transaktionsfunktion.
- Transaktionsfunktion - Diese Funktion wird in den Skripten Create_House_Vlan, Delete_House_Vlan, Add_Vlan_To_Router und Remove_Vlan_From_Router implementiert und überwacht den Ablauf der Operationen. Tritt ein Fehler auf, wird eine Eingabeaufforderung erzeugt und der Admin kann zwischen diesen Möglichkeiten wählen:
  1. abbrechen - Die Konfiguration abbrechen und den Ursprungszustand wiederherstellen.
  2. erneut versuchen - Aktuelle Operation erneut starten.
  3. nächstes Gerät - Mit dem nächsten Gerät fortfahren.
- Funktionen zum Speichern und Wiederherstellen von Konfigurationen auf einzelnen Geräten - Vor Beginn der Konfiguration eines Gerätes werden die aktuellen Parameter gespeichert, damit diese bei Bedarf wiederhergestellt werden können. Realisiert wird dies durch die Methoden “saveVlanConfig” und “restoreVlanConfig”.

5.4. Erweiterung der CSN-Datenbank und Anpassung der Nutzerverwaltung

Die CSN-Datenbank ist um alle notwendigen Tabellen und Beziehungen zu ergänzen, um alle wesentlichen Informationen einer Netzklasse abzupeichern und wieder zugreifbar zu machen. In den Anforderungen in Kapitel 2 wurde bestimmt, daß eine Netzklasse wiederherstellbar sein muß und die wichtigsten Konfigurationen jederzeit wieder neu geschrieben werden können.

Abbildung 5.5.: Entity Relationship Diagramm der Netzklassen



Desweiteren liegt das Augenmerk darauf, die neuen Strukturen möglichst “schonend” in die

bestehende Datenbank einzupassen, da diese bereits eine sehr grosse Komplexität erreicht hat.

Ein CSN-Nutzer kann bis zu zwei Rechner anmelden. Diese beiden Endgeräte teilen sich in den allermeisten Fällen eine Datendose, der Traffic läuft über einen einzigen Switchport. Es steht nur ein einiges native VLAN für beide Rechner bereit und somit scheitert eine Zuordnung der Rechner eines Nutzers in verschiedene Netzklassen vom Typ 2, da jede Klasse eine eigene virtuelle Netztopologie hat und somit unterschiedliche VLAN nutzt.
Eine Zuordnung zur Netzklasse muß nutzerbasiert erfolgen. Am einfachsten geschieht dies durch Einführung eines neuen Nutzerstatus. Wird ein CSN-Nutzer einer Netzklasse zugeordnet, müssen alle auf ihn angemeldeten Rechner einem VLAN zugeteilt werden.

In der Tabelle "Netzklasse" sind die wichtigsten Eigenschaften gespeichert, die direkt einer Netzklasse zugeordnet werden können. Zu den bereits in Kapitel 2 definierten Komponenten gehören der Name und die Beschreibung. Hinzu kommt eine Unterscheidung nach Typ, also ob es sich um eine Klasse vom Typ 1 oder 2 handelt. Die Funktion der Manglemaske wird weiter unten im Abschnitt Firewall beschrieben.
Für jede Netzklasse wird ein Nutzerstatus angelegt, welcher in der Tabelle "csn_nutzer_status" genauer beschrieben wird und CSN-Nutzern in der Tabelle "csn_nutzer_hat_status" zugeteilt wird.

Wie bereits beschrieben besteht eine Netzklasse aus mehreren VLAN, welche sich jeweils über ein Wohnheim erstrecken und jeweils ein eigenes Subnetz bilden. Die Tabelle "Netzklasse_im_Wohnheim" beschreibt diese Eigenschaft und ordnet für jede Klasse die entsprechenden VLAN den Wohnheimen zu.
Bei der IP-Adresskonfiguration ist zu unterscheiden, ob offizielle CSN IP-Adressen oder private Subnetze verwendet werden. Werden offizielle Adressen verwendet, dann wird über die bereits existierende Tabelle "Subnetz" die notwendige Konfiguration für das jeweilige VLAN ermittelt. Diese Tabelle enthält die wichtigsten Netzwerkparameter (z.B. GatewayIP, Subnetzdaten).
Werden jedoch private Subnetze genutzt, dann sind diese direkt in den Attributen "GatewayIP" und "priv_Subnetz" anzugeben. Es werden keine privaten Adressräume in der offiziellen Subnetztabelle gespeichert.
Ist das Attribut "Subnetz" der Tabelle "Netzklasse_im_Wohnheim" auf 0 gesetzt, so werden private Adressräume verwendet, ansonsten offizielle CSN Subnetze.

In Kapitel 2 wurde festgelegt, daß die Zugriffsrechte unterteilt werden müssen in eine iptables-Firewall für die Kontrolle der erlaubten Netzwerkprotokolle und den transparenten Proxy Squid für den HTTP-Zugriff. Die für den HTTP-Zugriff erlaubten Domains lassen sich mit einem Textfeld als Attribut "Domainliste" der Netzklasse direkt zuordnen. Die iptables Regeln werden in der Tabelle "Zugriffsrechte" abgelegt, jeweils bestehend aus dem Regelinhalt als ganzen iptables Befehl und einer Nummer zur Bestimmung der richtigen Reihenfolge.

Die Webformulare zur Nutzerverwaltung beinhalten bereits die Möglichkeit, den Nutzern andere Nutzerstati zuzuteilen. Hiermit können Nutzer per Mausklick anderen Netzklassen zugeordnet werden. Damit diese Änderung auch wirksam wird, muß für den Switchport, dem der Nutzerrechner zugeteilt ist, das in diesem Wohnheim geltende native VLAN der Netzklasse zugewiesen werden.

Die Nutzerverwaltung ist um VLAN-Funktionalität zu ergänzen. Bisher befanden sich die Nutzer standardmässig im "Haus-VLAN", eine Umschaltung in andere virtuelle Netze war nicht vorgesehen. Die Switch-API des CSN muß um die Methode "setNativeVlanOfPort($port,$vlan_id)" erweitert werden, welche unabhängig vom verwendeten Gerätetyp (Cisco oder 3com) das native VLAN eines Nutzerports setzen kann.
Am Nutzerport liegt immer ein native VLAN an, das Setzen bzw Entfernen des VLAN-Tags der Pakete wird vom Switch übernommen.

Wie bereits in Kapitel 1 erläutert, werden zur Zeit jede Stunde die aktuellen Änderungen auf allen Nutzerports mittels des Switchconfig Skriptes konfiguriert, welches die Switch-API nutzt. Dieses Skript muß zusätzlich zu den bisherigen Einstellungen noch das Setzen des native VLAN pro Nutzerport übernehmen. Somit können Nutzer bei Bedarf in zeitlich kurzen Abständen einer neuen Netzklasse zugeordnet werden.

Die Ermittlung des für den einzelnen Nutzer aktuell geltenden VLAN erfolgt ausgehend von der Tabelle "csn_nutzer". Die Netzklasse, welcher der Nutzer zugeordnet ist, wird über die status_id, die den Nutzerstatus beschreibt, ermittelt. Über das Attribut "wohnt" lässt sich das Wohnheim bestimmen, in dem der Nutzer wohnt. Somit kann das für diesen Switchport geltende native VLAN aus der Tabelle "Netzklasse_im_Wohnheim" ermittelt werden.

5.5. Pest.csn

Der Rechner pest.csn ist, wie weiter oben schon beschrieben, der Router und die Firewall für Netzklassen vom Typ 2. Jede dieser Klassen verfügt über eigene virtuelle Netze, welche alle auf diesem Rechner zusammenlaufen müssen. Die Kontrolle der Zugriffsbeschränkungen erfolgt auf OSI-Layer 3 und 4 mittels iptables-Firewall und die HTTP-Zugriffsrechte werden von einem squid als transparentem Proxy geprüft.
Aufgrund der bereits erläuterten Probleme der IP-Adresskonfigurationen der Nutzerrechner ist es notwendig, daß der Rechner pest.csn die Routingfunktionalität des CSN-Routers übernimmt. Die virtuellen Netze dieser Netzklassen dürfen nichtmehr vom CSN-Router auf OSI-Layer 3 geroutet werden, sondern müssen auf OSI-Layer 2 an die pest.csn weitergeleitet werden. Dies geschieht dardurch, indem keine IP-Adresse auf dem jeweiligen VLAN Interfaces im CSN-Router gesetzt wird.

Die pest.csn übernimmt die Funktionalität als Standardgateway für alle Nutzer dieser Netzklassen und es müssen die entsprechenden IP-Adresskonfigurationen für die jeweiligen VLAN angelegt werden. Werden einer Klasse am CSN angemeldete Rechner zugeteilt, müssen die IP-Konfigurationen den bisherigen "Haus-VLAN" entsprechen. Somit geschieht ein Umleiten des Traffics völlig transparent für die Nutzerrechner. Es ist keinerlei Änderung an der Netzwerkkonfiguration der Nutzerrechner notwendig.
Dies hat zur Folge, daß mehrere gleich konfigurierte Interfaces auf der pest.csn existieren können, jeweils eines von jeder Netzklasse mit offiziellen CSN-Adressen pro Wohnheim. Eine sichere Routingentscheidung auf der Basis von Subnetzen kann nichtmehr getroffen werden. Hostspezifische Routen sind notwendig.

Die vom CSN-Router weitergeleiteten VLAN müssen tagged an der pest.csn anliegen, damit über ein einziges reales Interface mit mehreren virtuellen Netzen kommuniziert werden kann.
Es wird für jedes VLAN ein virtuelles Interface benötigt, welches mittels dem Linux-Tool vconfig angelegt wird. Dieses Programm ermöglicht das Erstellen von VLAN-Devices, welche einem realen Netzwerk Interface zugeordnet werden. Genauere Details kann man [21] und [6] entnehmen.
Die pest.csn verfügt zu diesem Zweck über 2 Interfaces, welche beide direkt mit dem CSN-Router verbunden sind. Ein Interface für den ausgehenden Traffic in Richtung csn-server.csn und das andere Interface, zu welchem der Traffic der Netzklassen weitergeleitet wird.

Die Konfiguration der pest.csn erfolgt mittels Secure Shell(ssh) durch das Kopieren von Textdateien und Ausführen von Befehlen. Eine grundlegende API in perl wird bereitgestellt und umfasst z.B. das Erzeugen neuer IP-Listen für die Zuordnung der Nutzerrechner zu entsprechenden Netzklassen.

5.5.1. iptables Firewall

Aktuell wird auf dem csn-server.csn eine iptables Firewall genutzt, diese basiert auf der Studienarbeit “Firewall mit nutzerindividuellen Regeln” von Heiko Jehmlich.[22]
Die Firewall muß um netzklassenspezifische Regeln erweitert werden. Die günstigste Möglichkeit dafür ist das Zusammenfassen der einzelnen Netzklassenregeln, welche in der Datenbank gespeichert sind, in der richtigen Reihenfolge in einer Regelkette. Diese wird dann bei Bedarf angesprungen und durchlaufen. Somit lassen sich sowohl nutzer- als auch netzklassenspezifische Elemente vereinigen.

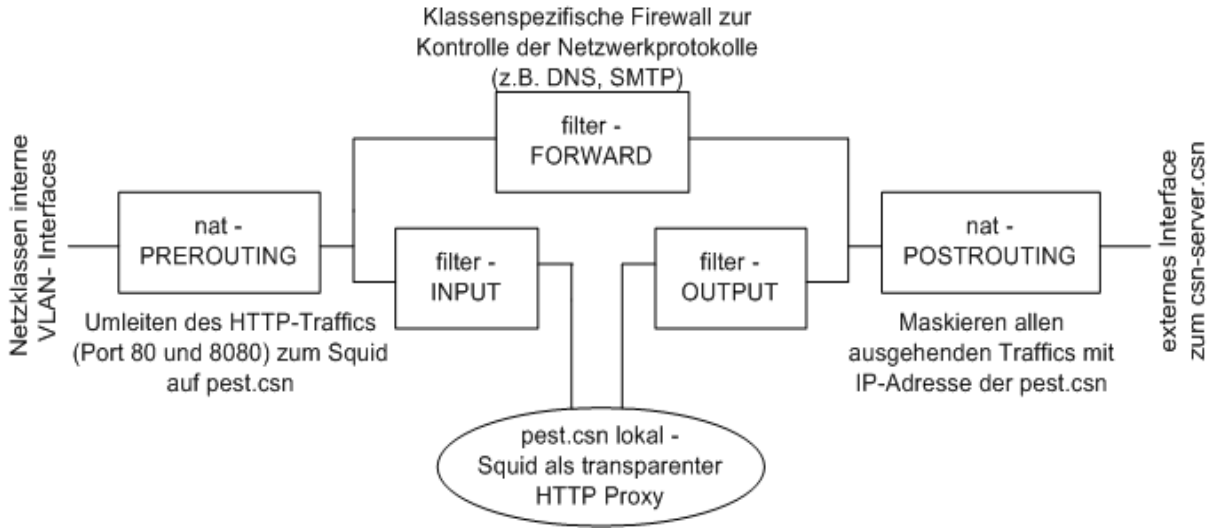
Die beiden Typen der Netzklassen haben als Firewall entweder den csn-server.csn oder die pest.csn. Hier muß eine weitere Unterscheidung getroffen werden.
Auf dem csn-server.csn erfolgt die Zuordnung der einzelnen Pakete durch das Setzen einer 32 Bitmaske, dem sogenannten FWMARKS. Mit dieser Bitmaske wird jedes eingehende Paket markiert.

```
Bits:           31         30   ...     16  15      ....      0
Funktion:  Shaping Bit       Person-ID        Netzklassen-ID
```

Das Shaping Bit signalisiert dem CSN-Shaper, ob der Traffic zu shapen ist oder nicht. Die Person-ID dient zur Zuordnung verschiedener IP-Adressen zu einem CSN-Nutzer. Anhand der Netzklassen-ID, die sich aus dem Datenbank-Attribut “MangleMaske” ergibt, kann nun die klassenspezifische Regelkette angesprungen werden.

Auf der pest.csn wird eine wesentlich einfachere Firewall eingesetzt, welche ausschließlich auf netzklassenspezifische Regeln ausgerichtet ist. Die Zuordnung der Pakete zu den Netzklassen erfolgt nicht über eine Markierung, sondern durch die Klassifikation über VLAN-Interfaces. Die verschiedenen virtuellen Interfaces auf der pest.csn können eindeutig den Netzklassen zugeordnet und somit der Traffic dieser Interfaces in die klassenspezifischen Regelketten weitergeleitet werden.

Abbildung 5.6.: Schematische Darstellung der Funktionsweise der Firewall auf pest.csn



Der die pest.csn in Richtung csn-server.csn verlassende Traffic muß maskiert werden (mittels Network Address Translation), also die Quell IP-Adresse in die der pest.csn geändert werden. Dies ist notwendig, da bei Verwendung offizieller CSN IP-Adressen innerhalb einer Netzklasse vom Typ 2 der gesamte zurückkommende Traffic ansonsten vom CSN-Router in die normalen “Haus-VLAN” und damit in die Struktur der Klassen vom Typ 1 zurückgeroutet werden würde. Die “Alternative” hierzu wären Hostrouten auf dem CSN-Router, was nicht in Betracht kommt.
Weiterhin ermöglicht dies die Nutzung von privaten IP-Adressräumen innerhalb von Netzklassen des Typ 2, womit auch Netznutzungarten für nicht am CSN angemeldete Rechner geschaffen werden können(z.B Anmeldennetz, Wlan-Netz).

5.5.2. DHCP Konfiguration

Bei der DHCP Konfiguration von Nutzerrechnern ist zu unterscheiden, ob eine Netzklasse den offiziellen CSN-Adressraum nutzt oder einen eigenen privaten Adressbereich.
Den Nutzern der ersteren Netzklasse muß die Möglichkeit gegeben werden, ihren Rechner weiterhin per DHCP zu konfigurieren, auch wenn er einer Klasse vom Typ 2 zugeordnet ist und als Standardgateway die pest.csn hat. Nutzerrechner einer Netzklasse mit privaten Adressraum müssen ihre Netzwerkeinstellungen zwangsweise per DHCP konfigurieren.

Hier wird ein DHCP Relay Agent auf der pest.csn benötigt, welcher die geforderte Funktionalität schafft. Es können für Nutzerrechner mit offiziellen IP-Adressen die bisher im CSN laufenden DHCP-Server genutzt werden. Diese müssen sich nicht im aktuellen Subnetz des Endgeräts befinden, der DHCP Relay Agent teilt dem DHCP Server mit, aus welchem Subnetz er die Adresse zu vergeben hat. Der DHCP Relay Agent muß auf den entsprechenden VLAN-Interfaces der Netzklassen mit öffentlichem Adressraum lauschen.
Bei privaten Subnetzen wird einfach der auf der pest.csn laufende lokale DHCP Server genutzt.

5.5.3. squid als transparenter Proxy

Die Kontrolle der HTTP-Zugriffe der Netzklassen vom Typ 2 wird von einem Squid Proxy Server übernommen, welcher im transparenten Modus auf der pest.csn arbeitet. In diesem Modus kann der Proxy den Traffic für den Nutzer weitestgehend unsichtbar kontrollieren.
Der auf den VLAN-Interfaces der Netzklassen ankommende Traffic muß zum Squid umgeleitet werden, damit dieser die HTTP-Requests an der für die Klasse definierten erlaubten Domainliste prüfen kann. Diese Redirection erfolgt mittels iptables Kommandos, aktuell werden die beiden Ports 80 und 8080 zum lokalen Squid weitergeleitet.
Ein Umleiten dieser beiden Ports ist nötwendig, da die Behandlung der HTTP-Requests unabhängig von der Browserkonfiguration der Nutzerrechner funktionieren soll. Dem Nutzer ist es aktuell freigestellt, ob er garkeinen HTTP-Proxy mehr konfiguriert oder die Proxy Server der TU-Chemnitz hinter dem Lastverteiler www-cache nutzt.
Der Squid selbst lässt alle Zugriffe intern aus den Netzklassen zu und überlässt die Kontrolle der Zugriffrechte dem Programm Squidguard.

Für jede Netzklasse wird ein source- und ein dest-Abschnitt in der squidGuard.conf angelegt. Der src-Teil enthält eine Liste von Nutzer IP-Adressen dieser Netzklasse und der dest-Abschnitt eine Liste der erlaubten Domains. Greift ein Nutzer auf eine für seine Netzklasse nicht erlaubte Domain zu, wird er auf eine Informationsseite weitergeleitet.
Diese Webseite dient dazu, dem Nutzer Informationen über den Grund, warum er sich in dieser Netzklasse befindet, und eine Liste der für ihn erlaubten Domains zu geben.

Auszug aus der squidGuard.conf:

```
src quarantaene_clients {
    iplist /etc/squid/quarantaene_clients
}
dest quarantaene_domain {
    domainlist /etc/squid/quarantaene_domain
}
acl {
    quarantaene_clients {
        pass quarantaene_domain csn_domain none
        redirect http://www.csn.tu-chemnitz.de/netzklassen/quarantaene.html
    }
    ...
}
```

Dem Quarantänenetz zugeordnete Clients (IP Liste als Textdatei quarantaene_clients) dürfen nur auf erlaubte Domains (Domainliste als Textdatei quarantaene_domain) und die CSN-internen Webseiten zugreifen, ansonsten werden sie auf eine Informationsseite weitergeleitet.

Der Vorteil dieser Konfiguration ist das einfache Updaten der IP Listen in Textdatein, ohne eine Änderung an der eigentlichen squidGuard.conf vornehmen zu müssen. Mehrere Netzklassen können von einem Squid-Server kontrolliert werden.

Der transparente Proxy selbst kann keine verbindungsorientierten Protokolle behandeln (z.B.

FTP, HTTPS). FTP erfordert neben einer Transferverbindung noch eine Steuerverbindung. HTTPS würde diesen Proxy als “man-in-the-middle-attack“ betrachten.
Die Nutzung von HTTPS muß aber möglich sein (z.B. mail.tu-chemnitz.de, CSN-Webseiten), aus diesem Grund muß in der iptables Firewall der Port 443 für die erlaubten Domains soweit wie nötig freigeschalten werden, damit eine direkte Verbindung ohne transparenten Proxy aufgebaut werden kann.

6. Schlussbetrachtung

6.1. Spezielle Netzklassenprobleme

Das in dieser Arbeit eher allgemein gehaltene Konzept der Netzklassen bietet eine gute Funktionalität, um die Möglichkeiten der Netznutzung differenzierter zu gestalten. Jedoch hat dies den Nachteil, daß spezielle Anforderungen des praktischen Einsatzes einzelner Netzklassen nicht berücksichtigt wurden (z.B. aus technischen Gründen). Aus diesem Grund möchte ich noch kurz auf spezielle Eigenschaften einzelner Klassen eingehen, die Probleme aufwerfen können.
Im Quarantäne-Netz sind die als wurminfiziert vermuteten Nutzerrechner zwar getrennt von allen anderen Rechnern, aber diese können weiterhin innerhalb eines Hauses untereinander kommunizieren. Dies liegt an der Verwendung hausweiter VLAN. Hier ist eine komplette Isolation dieser Rechner wünschenswert. Ein Ansatzpunkt hierfür wäre die Verwendung von minimalen Subnetzen innerhalb des Quarantäne-Netzes, welche nur aus einem inifzierten Endgerät und der pest.csn bestehen. Hierbei muß natürlich die Problematik der fest im Nutzerrechner konfigurierten Netzwerkeinstellungen berücksichtigt werden.

Dem Anmeldenetz müssen alle die Switchports zugeordnet werden, auf denen keine Nutzerrechner angemeldet sind. Die Portsecurity muß für diese Ports ausgeschaltet werden, ansonsten fällt der Switchport zu, weil eine unbekannte MAC an ihm anliegt.
Hierbei ergibt sich das organisatorische Problem, daß die meisten Nutzer beim Auszug aus dem Wohnheim ihre Rechner nicht korrekt abmelden. Der Nutzerrechner ist weiterhin angemeldet, obwohl die Dose faktisch nichtmehr belegt ist. Die Freigabe des Switchports erfolgt erst nachdem die Rückmeldefrist für den Rechner abgelaufen ist.
Während dieser Zeitspanne kann wieder ein neuer Mieter in das Zimmer ziehen, aber er wird keinen Zugriff auf das Anmeldenetz erhalten, da der Switchport noch der Netzklasse zugeordnet ist, in welcher sich der Rechner des Vormieters befand. Hier greift die Portsecurity und der Port wird geschlossen.
Das Anmeldenetz sollte mit diesem Problem umgehen können, damit jeder neue Nutzer seinen Rechner von zuhause aus anmelden kann und dies nicht vom Verhalten des Vormieters abhängig ist.
Eine erste Idee wäre die Verwendung von SNMP-Traps bei einer PortSecurityViolation. Liegt bei einem Switchport, auf dem ein Rechner angemeldet ist, eine falsche MAC an, wird eine SNMP-Trap an ein Managementprogramm gesendet. Daraufhin könnte der Port vom Manager in das Anmeldenetz gesetzt werden.
Dies erfordert weitere Untersuchungen bezüglich neuer Fragestellungen. Wie verlässlich ist die SNMP-Trap? Was passiert, wenn der Nutzer garkeinen neuen Rechner anmelden wollte, sondern einfach nur ein Paket mit der falschen MAC den Switch erreicht hat? Die Gründe dafür können vielfältig sein, von Fehlern der Netzwerkkarte bis zu falsch konfigurierten lokalen Netzen eines Nutzers.

6.2. Fazit

Das Ziel dieser Studienarbeit war es, die Netznutzungsmöglichkeiten im CSN durch die Einführung von Netzklassen differenzierter gestalten zu können. Es wurde ein weitesgehend allgemeingültiges Konzept für die aktuelle CSN-Netzstruktur geschaffen und ein Prototyp implementiert, welcher die Basis der benötigten Funktionaliät bereitstellt. Besonderes Augenmerk lag darauf, dieses neue Konzept "schonend" in die bisher bestehende Netztopologie und Datenbank einzupassen.

Die programmierte Software liegt der Arbeit bei, ist aber aktuell nicht ausgereift und stellt praktisch kein eigenständiges Paket dar, sondern ist sehr stark mit anderen CSN-Bibliotheken verknüpft.
Allerdings werden sich grosse Teile dieser Arbeit sehr bald im OpenSource Downloadbereich der CSN Webseiten [23] wiederfinden. Die Dokumentation der CSN-Switch-API wird im Moment überarbeitet und in Kürze anderen Studentennetzen bereitstellt, um ihnen die automatische Konfiguration der Netztechnik mittels SNMP zu erleichtern.

Aktuell erfordert das Konfigurieren einer neuen Netzklasse noch hohen personellen Aufwand und tiefgreifendes Wissen aus dem Bereich Rechnernetze. Viele Parameter, deren Abhängigkeiten untereinander bedacht werden müssen, sind per Hand einzustellen. Es sind viele Einzelschritte nacheinander notwendig (Konfigurieren von VLAN auf den Switches aller Wohnheime, Anlegen der Interfaces auf dem Router und der pest.csn, ...), bevor eine Netzklasse anfängt zu existieren.
Desweiteren sind zur Zeit grosse Teile der Transaktionsfunktionalität nochnicht ausreichend implementiert.

Neben den bereits im letzten Abschnitt beschrieben speziellen Problemen einzelner Netzklassen gibt es noch viele weitere Punkte, an denen diese Studienarbeit verbessert werden kann.
Einige davon wären z.B. das Vereinfachen der Konfigurationseinstellungen oder die Erweiterung des Konzepts "pro Netzklasse und Haus ein VLAN" auf eine grössere Anzahl an virtuellen Netzen einer Netzklasse pro Haus. Durch die konsequente Aufrüstung der CSN-Technik auf Geräte der Marke Cisco, wird dieser Schritt in wenigen Jahren machbar und sinnvoll sein. Dann könnte man auch einen DHCP-Zwang einführen und die IP-Subnetze der einzelnen Netzklassen wesentlich freier bestimmen.
Eine weitere sinnvolle Ergänzung wäre eine grafische Visualisierung der aktuellen Netzstrukturen, um das Ganze anschaulicher zu gestalten.

Ich hoffe, daß mit dieser Studienarbeit, der Netzzugang für die Nutzer besser gestaltet und die Arbeit der CSN-Mitarbeiter erleichtert werden kann.

A. Programm Dokumenation

A.1. CSN-Switch-API: Übersicht über die Methoden zur VLAN-Konfiguration

listAllVlanIDs

Erzeugt eine Liste aller vom Gerät verwalteten VLAN_IDs

checkIfVlanIsFree($global_vlan_id)

Prüft ob übergebene $global_vlan_id auf dem aktuellen Gerät frei ist

getFreeVlanCount

Liefert die Anzahl der noch freien VLAN zurück.

listAllVlanInfo

Liefert eine Liste aus Hash-Referenzen zurück. Jede enthält alle wichtigen Informationen für ein vom Gerät verwaltetes VLAN (ID, DESCR, STATUS, zugeordnete tagged Ports, zugeordnete native Ports).

createVlan($global_vlan_id,$global_vlan_descr)

Erzeugt ein neues VLAN mit $global_vlan_id als ID und $global_vlan_decsr als Beschreibung.

deleteVlan($global_vlan_id)

Loescht das VLAN mit der übergebenen $global_vlan_id. Dabei wird auch gleichzeitig diese VLAN_ID aus den ihr zugeordneten Trunk-Ports gelöscht. Befinden sich noch native Ports im VLAN bricht das Löschen ab!

addVlanToTrunkPort($global_vlan_id, $port_mapped)

Fügt dem Trunk-Port $port_mapped die VLAN_ID $global_vlan_id hinzu.

removeVlanFromTrunkPort($global_vlan_id, $port_mapped)

Löscht die VLAN ID $global_vlan_id vom Trunk-Port $port_mapped.

setNativeVlanOfPort($global_vlan_id, $port_mapped)

Setzt das native VLAN des Ports $port_mapped auf die VLAN_ID $global_vlan_id.

saveVlanConfig

Sichert die VLAN Einstellungen. Diese Methode sollte vor Ausführung von Änderungen an der Konfiguration aufgerufen werden. Bei Cisco Geräten wird die running_config auf einen tftp Server kopiert. Bei 3com's wird die Rückgabe der Methode listAllVlanInfo auf Festplatte gespeichert.

restoreVlanConfig

Stellt eine vormals gespeicherte VLAN Konfiguration wieder her. Bei Cisco Geräten wird die running_config von einem tftp Server geladen. Bei 3com's mittels vergleichen mit der aktuellen Konfiguration.

3COM SPEZIFISCHE METHODEN FUER 802.1Q

getVlanVirtIfIndex($global_vlan_id)

Liefert den intern verwendeten virtuellen Interface Index der VLAN_ID $global_vlan_id zurück.

getEncapsVirtIfIndex($global_vlan_id)

Liefert den intern verwendeten virtuellen Interface Index der 802.1q Encapsulation der VLAN_ID $global_vlan_id zurück.

getHighestPortIf

Liefert den höchsten real am Gerät existierenden Port in Unit-Darstellung zurück.

getVlanIfHash

Liefert einen Hash aus Zuordnung internes virtuelles Interface => VLAN_ID für alle VLAN zurück.

CISCO SPEZIFISCHE METHODEN FUER 802.1Q

createTrunkPort($port_mapped)

Setzt einen Port $port_mapped als Trunk-Port.

disableVTP

Deaktiviert das VLAN Trunking Protocol.

getVlanIDsOfTrunkPort($port_mapped)

Liefert eine Liste aller am Trunk-Port $port_mapped angelegten VLAN_IDs. Die dem Trunk-Port zugeordneten VLAN werden intern als Bitmaske behandelt.

setVlanIDsOfTrunkPort($port_mapped, @vlan_id_list)

Ordnet eine Liste von VLAN_IDs @vlan_id_list dem Trunk-Port $port_mapped zu. Die Liste wird in eine Bitmaske umgewandelt und dann geschrieben.

Abkürzungsverzeichnis

API	Application Programming Interface
BNC	Bayonet Navy Connectory
CSN	Chemnitzer Studenten Netz
DHCP	Dynamic Host Configuration Protocol
IP	Internet Protocol
IPX	Internetwork Packet Exchange
LAN	Local Area Network
MAC	Media Access Control
OSI	Open Systems Interconnection Reference Model
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VPN	Virtual Private Network
VTP	VLAN Trunking Protocol
WLAN	Wireless Local Area Network

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Chemnitzer Studenten Netz,30.07.2005
https://www.csn.tu-chemnitz.de

[2] Netzentwicklung im CSN, Markus Schade, 20.06.2004
http://archiv.tu-chemnitz.de/pub/2004/0135/index.html

[3] IEEE 802.1: 802.1Q - Virtual LANs, 15.03.2005
http://www.ieee802.org/1/pages/802.1Q.html

[4] VLAN - Wikipedia, 18.03.2005
http://de.wikipedia.org/wiki/VLAN

[5] VLAN, 30.06.1998
http://www.lrz-muenchen.de/services/schulung/unterlagen/netztechniken/sld021.htm

[6] Linux Magazin "network edition", Seite 12-14, Linux New Media AG, Sonderheft 3/2004

[7] Facharbeit Virtuelle lokale Netze, Christian Pötzsch und Jens Langner, 23.07.2000
http://www.jens-langner.de/ftp/vlan.pdf

[8] IEEE 802.1x - Wikipedia, 15.06.2005
http://de.wikipedia.org/wiki/802.1x

[9] LANCOM Techpaper 802.1x, 05.10.2004
http://www.lancom-systems.de/produkte/feature/techpaper/TP-WLAN-80211x-DE.pdf

[10] Zusammenfassung zum Seminar Wireless-Networks, Tobias Feldmann, WS2004/2004
http://www.uni-koblenz.de/ steigner/seminar-wlan/4-feldmann.pdf

[11] Simple Network Management Protocol - Wikipedia, 25.03.2005
http://de.wikipedia.org/wiki/SNMP

[12] SNMP - Netzwerk-Management mit Hilfe von SNMP, 21.03.2004
http://www.jklein.de/techniker_arbeit/tech_html/snmp_allgemein.htm

[13] SNMP - Simple Network Management Protocol, 16.03.2005
http://www.elektronik-kompendium.de/sites/net/0902011.htm

[14] Iptables - Wikipedia, 27.05.2005
http://de.wikipedia.org/wiki/Iptables

[15] Iptables Tutorial 1.2.0, Oskar Andreasson, 30.05.2005
http://iptables-tutorial.frozentux.net/iptables-tutorial.html

[16] Dynamic Host Configuration Protocol - Wikipedia, 25.05.2005
http://de.wikipedia.org/wiki/DHCP

[17] HowTo DHCP, 20.05.2005
http://www.planet-rcs.de/de/article/dhcp_howto/

[18] Abstraktion von Managementaufgaben aktiver Netzkomponenten, Thomas Kuschel, 07.03.2005
http://www-user.tu-chemnitz.de/ tkus/Projekte/Studienarbeit/Arbeit/

[19] Die Projekte Campusnetz II und IP-Telefonie (VoIP), 04.2005
http://archiv.tu-chemnitz.de/pub/2005/0039/data/netz/

[20] Transaktion (Informatik) - Wikipedia, 25.04.2005
http://de.wikipedia.org/wiki/Transaktion_(Informatik)

[21] 802.1Q VLAN implementation for Linux, 15.09.2004
http://www.candelatech.com/ greear/vlan.html

[22] Studienarbeit "Firewall mit nutzerspezifischen Regeln", Heiko Jehmlich
http://archiv.tu-chemnitz.de/pub/2003/0142/index.html

[23] CSN Downloads, 30.07.2005
https://www.csn.tu-chemnitz.de/OpenSource/twiki/WebHome.html