UNIVERSITÄT KOBLENZ · LANDAU
Fachbereich 4 Informatik

gemeinsames hochschul rechenzentrum uni koblenz

Netzwerkmonitoring in einer dynamischen Umgebung

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Computervisualistik

vorgelegt von

Sebastian Mitnacht

geb. am 10.10.1976 in Würzburg

Betreuer: Dipl.-Inform. Uwe Arndt, Rechenzentrum Universität Koblenz

Erstgutachter: Prof. Dr. Christoph Steigner, Fachbereich 4 Informatik
Zweitgutachter: Dipl.-Inform. Uwe Arndt, Rechenzentrum Universität Koblenz
Koblenz, den 28.09.2006

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Sebastian Mitnacht Koblenz, den 28.09.2006

Inhaltsverzeichnis

1 Ziel der Arbeit

Das Thema der Arbeit entstand aus dem Arbeitsalltag des Rechenzentrums und der Suche einer rechnergestützten Verfügbarkeitskontrolle der gesamten IT-Struktur des Campus Koblenz der Universität Koblenz-Landau. Das Netzwerk, welches über Jahre gewachsen ist, wird immer unübersichtlicher und erlangt eine Größenordnung, die ein Management notwendig macht, um eine zufriedenstellende Verfügbarkeit zu gewährleisten. Immer mehr Netzwerkknoten, wie Hubs, Switches, Router und Computer von Universitätsmitarbeitern, erweitern wöchentlich die digitale Infrastruktur. Fachbereiche, von denen man nicht unbedingt erwartet, dass Computer eine große Rolle spielen, nutzen die Technik immer mehr für ihre Arbeit. Computer aus den informatikfremden Fachbereichen, die Computer bislang lediglich für lokale Arbeiten, wie Textverarbeitung, genutzt haben, verwenden vermehrt E-Mail, Word Wide Web und Serverdienste zur Erledigung der täglichen Arbeit. In den letzten Jahren hat die Bedeutung des Internets, zur Informationsbeschaffung und des Datenaustaustauschs, in allen Bereichen enorm zugenommen. Die stetig zunehmende Anzahl an Computer in allen Fachbereichen führt unweigerlich zu einer stärkeren Auslastung des Netzwerks. Der Grund für die permanente Netzwerknutzung soll durch einige Beispiele verdeutlicht werden.
Jeder Computer wird regelmäßig mit Updates und Service Packs versehen, da diese sicherheitsrelevante Veränderungen mit sich bringen. Es existiert kaum eine Software auf dem Markt, die nicht permanent nachgebessert werden muss. Zu diesem Zweck nutzen die Programme das Internet, um die Updates auf den Rechnern aufzuspielen.
Ein weiterer Aspekt behandelt die verwendeten Anwendungen auf den Rechnern, die nicht lokal installiert sind, sondern über das Netzwerk geladen werden. Durch diesen Vorgang ist es zwar leichter eine Software auf allen Computern gleichzeitig auf dem aktuellen Stand zu halten, Nachteil ist die Hochverfügbarkeit des Netzwerks, ohne die das Programm nicht zur Verfügung steht. Das Resultat ist eine höhere Beanspruchung des Netzwerks durch die verwendete Software der Benutzer.
In größeren Umgebungen, wie einer Universität, existieren wesentlich mehr Benutzer als Rechner. Diese Situation löst man durch wandernde Benutzerprofile. Das bedeutet, das die Benutzereinstellungen, die E-Mails und Daten nicht auf einem Computer gespeichert sind, wie man es aus dem Heimbetrieb kennt, sondern auf Servern, die zentral verwaltet sind. Verwendet ein Benutzer einen Computer, authentifiziert dieser sich an einem Server und bekommt seine Benutzerdaten auf den lokalen Rechner kopiert. Nach Beendigung der Arbeit werden alle Benutzerdaten wieder auf den Server kopiert und von dem gerade genutzten Computer gelöscht. Dieser steht ab diesem Zeitpunkt einem anderen Benutzer zur Verfügung. Die Daten der Benutzer sind auf Dateiservern gespeichert, und für den Anwender über das Netzwerk zugreifbar. Das bedeutet, dass nicht nur das Netzwerk in einem stabilen Zustand sein muss, sondern auch die Server, die ununterbrochen Daten

bereitstellen.
Betrachtet man exemplarisch die Fachbereiche Diplompädagogik oder Psychologie, welcher in den letzten Jahren verstärkt die Statistiksoftware SPSS[1] einsetzen, benötigt jeder PC, der SPSS verwendet, eine stehende Verbindung zum Lizenzserver in Landau, damit das Programm funktioniert. Es muss gewährleistet sein, dass das lokale Netzwerk und das Internet permanent verfügbar ist, um ein reibungsloses Arbeiten zu ermöglichen.
Selbst die Institute für Theologie nutzen in den Vorlesungen digitale Medien. Das berührt im ersten Moment nicht das Netzwerk, aber demonstriert deutlich die zunehmende Akzeptanz der „neuen“Medien.
Seit einigen Jahren gibt es die Technik des Wireless LAN. Computer müssen nicht mehr zwingend über ein Kabel an ein Netzwerk angeschlossen werden, sondern stellen den Kontakt über Funk her. Um Computer an ein Funknetz anzubinden, sind Accesspoints erforderlich. Diese dienen als Brücke zwischen dem lokalen, kabelbasierenden Netzwerk und dem funkenden Computer. Die Accesspoints sind herkömmliche durch ein Kabel an das Netzwerk angeschlossen und erlauben auf diese Art eine Erweiterung des Netzwerks ohne weiteren Kabelaufwand. Was Anfangs lediglich durch den Fachbereich 4 *Informatik* verwendet worden ist, gewinnt immer mehr Anwender aus den restlichen Fachbereichen. Es erfolgte ein weitreichender Ausbau des Funknetzwerks. Heute sind alle Gebäude des Campus und auch der Großteil der Außenflächen per Funk erschlossen. Das Angebot des Funknetzes erlaubt es, das deutlich mehr Studierende und Mitarbeiter zu einem Zeitpunkt das Angebot des Netzwerks zu nutzen. Ohne ein Funknetzes wäre die doppelte Anzahl an frei zugänglichen Computerarbeitsplätzen notwendig. Auf diese Weise spart die Universität die Kosten für die Anschaffung von weiteren öffentlichen Computern.

Es ist ein immer höheres Maß an Verfügbarkeit des Netzwerks und seiner Komponenten gefragt. Zur Lösung dieser Problematik sind statistische Erhebungen unbedingt erforderlich. Der Einsatz von Netzen, wie an den eben aufgeführten Beispielen gezeigt, gewinnt immer mehr an Bedeutung und macht die Anwender im gleichen Zuge von diesem abhängig. Um zu gewährleisten, dass das Netzwerk und die benötigten Server dauerhaft verfügbar sind, muss es eine ununterbrochene Kontrolle der Verfügbarkeit und Effizienz geben. Der Einsatz eines Mitarbeiters, der es zur Aufgabe hat, alle eingesetzten Geräte und Dienste zu prüfen, ist keine befriedigende Lösung. Die Überwachung muss rechnergestützt erfolgen und im Problemfall die Administratoren rechtzeitig informieren, damit es zu keinem Ausfall eines Systems kommt. Um nicht in die Lage zu kommen, dass es zu Ausfällen kommt, ist eine rechtzeitige Anpassung sowohl des Netzwerks als auch der Server notwendig. Dies wird erzielt durch regelmäßige Kontrollen und Erzeugung von Statistiken. Anhand dieser Angaben ist es möglich Entscheidungen für Veränderungen zu treffen. Zur Entwicklung eines geeigneten Systems gilt es Überlegungen anzustellen, die eine möglichst effektive Unterstützung versprechen.

[1] http://www.spss.com/de

Aspekte für die Systementwicklung

Es soll ein System zum Einsatz kommen, das alle Netzwerkinfrastrukturkomponenten wie Switches, Router und WLAN-Accesspoints, im Folgenden *Devices*, sowie Endgeräte wie Arbeitsstationen und Server, im Folgenden *Nodes* genannt, verwalten kann und in einem Monitoringsystem bereitstellt. Die Netzwerkkomponenten und Serverdienste sollen nicht nur, durch ein Softwaresystem, automatisch gefunden und aufgelistet werden, sondern es ist der Standort jedes Gerätes auf dem Campus zu ermitteln. Diese Informationen werden im weiteren Verlauf der Entwicklung zum Einen für eine grafische Darstellung benötigt, und zum Anderen erlauben Abhängigkeitsbeziehungen im Fehlerfall eine bessere Einschätzung bezüglich des Problemausmaßes. Auf Grund der erwarteten großen Menge an Netzwerkgeräten soll die Erfassung vollautomatisch erfolgen. Unter dem Aspekt, dass in großen Umgebungen Hardware unterschiedlichster Hersteller zum Einsatz kommt, darf das System nicht auf die Fähigkeiten spezieller Hardware zurückgreifen, um alle vorhandenen Geräte zu erreichen. Neben der Erfassung der Netzwerktopologie, sind aussagekräftige Informationen, wie der aktuelle Status oder Datendurchsatz eines gewissen Zeitraums, der Geräte für die tägliche Arbeit eines Administrators von Bedeutung. Welche Daten von Interesse sind, ist unter Umständen zum Zeitpunkt der Entwicklung noch nicht vorhersehbar und sollte daher nachträglich in das System integrierbar sein. Es ist ein Weg oder geeignete Mittel zu finden, die Informationen so zu ermitteln und bereitzustellen, dass keine aufwendige Modifikation der Software notwendig ist, um auf Situationen reagieren zu können, die erst im Laufe des Einsatzes der Systems in der Umgebung an Bedeutung gewinnen. Ein Monitoringsystem sollte auf die individuellen Bedürfnisse anpassbar sein und nicht geprägt sein, von einem Reservoir an festen Methoden, die sich ein Team von Entwicklern einmal ausgedacht hat. Individualität und Anpassbarkeit ist notwendig, da jede Einsatzumgebung ihre Eigenarten hat und zudem die Absichten der Administratoren zu berücksichtigen sind. Eine Modifikation oder Funktionserweiterung sollte ohne großen Zeitaufwand und aufwändige Implementierungen möglich sein. Die Verwendung von Plugins, die jederzeit in die Umgebung integriert werden können, ohne das bisherige System zu stören, stellt eine brauchbare Basis dar und erweitern den Funktionsumfang in das Unendliche.
Es existieren diverse Tools auf dem Markt, die Monitoring innerhalb eines Netzwerkmanagementsystems ermöglichen. Hersteller wie *IBM*[2], *HP*[3] oder *Cisco*[4] bieten von Haus Tools an, die allerdings über das reine Monitoring hinausgehen, enormen Zeitaufwand in der Konfiguration verlangen und hohe Summen an Lizenzgebühren kosten. Firmen verwenden in der Regel Software des Hardwareherstellers, da somit die korrekte Funktion zwischen Software und Hardware erwartet werden kann. Kommerzielle Systeme bieten jedoch nicht immer frei definierbare Schnittstellen und Konfigurationsmöglichkeiten an, sondern erlauben nur den Einsatz eigener Entwicklungen. Der Administrator ist auf die Funktionen angewiesen, die sich der Entwickler überlegt hat.
Angestrebt ist eine *Open-Source* Lösung, die nach Belieben veränderbar ist. Ein Vorteil

[2] http://www.ibm.com
[3] http://www.hp.com/openview
[4] http://www.cisco.com

einer Eigenentwicklung oder die Verwendung von *Open Source Software* ist die Anpassbarkeit an die vorhandene Umgebung und gegebenenfalls das Einpflegen von eigenem Sourcecode, der die speziellen Ansprüche besser bedient. Um nicht bereits vorhandene Software neu zu implementieren, ist der Markt auf Software zu prüfen, die den gewünschten Funktionsumfang mitbringen. Es ist der Einsatz mehrere Softwarepakete vorstellbar, mit dem Einsatz von eigenen Erweiterungen, die zum Einen die Software verbinden und zum Anderen Ergänzungen einbringen können.
Ein besonderes Anliegen ist die graphische Repräsentation des Netzwerks, ausschließlich der baulichen Gegebenheiten des Campus Koblenz. Aus softwareergonomischer Sicht soll das System den Campus visuell repräsentieren, um den Anwender besser leiten zu können. Es besteht die Vorstellung, auf oberster Ebene der Software, ein Luftbild des Campus zu verwenden, in dem Informationen für den Administrator übersichtlich bereit gehalten werden. Anhand dieser Darstellung sollen Probleme nicht nur in Zahlen dargestellt werden, sondern auch intuitiv die Auswirkungen auf das Gesamtnetz verdeutlicht werden, welches durch reine Wort- und Zahldarstellungen nicht zu erreichen ist. Eine Ebene unter dem Luftbild soll zwischen den Gebäuden unterschieden werden und durch Unterscheidung der Flure einen differenzierten Einblick ermöglichen. So entsteht eine hierarchische Informationsdarstellung, die in jeder Ebene verfeinerbar ist. Der visuelle Eindruck von Problemzonen erlaubt eine bessere Assoziation mit der tatsächlichen Gegebenheit und soll bessere Rückschlüsse für Problemregionen zulassen. Desweiteren sind textuelle Ortangaben nicht immer intuitiv. Ein Bild erlaubt in vielen Fällen ohne sprachliche Ergänzungen bereits die richtige Interpretation eines Sachverhaltes.
Im Folgenden wird in Kapitel 2 das Thema Netzwerkmanagement und naheliegende Themen diskutiert. In Kapitel 3 werden grundlegende Themen von Netzwerken beschrieben, die in dieser Arbeit verwendet werden, welche Techniken Seitens der Hersteller zur Verfügung stehen, um Nachbarschaften zur ermitteln und zur Verfügung zu stellen. Kapitel 4 beschreibt die Technik und den Aufbau des Campusnetzwerks der Universität Koblenz und welche Gedanken zur logischen Umsetzung Einfluss genommen haben. Kapitel 5, in welchem der praktische Teil beginnt, stellt die Werkzeuge vor, die zur Realisierung des Systems verwendet worden sind. Lösungen werden in Kapitel 6 präsentiert und begründet und bietet einen Ausblick auf Möglichkeiten, die im Rahmen dieser Arbeit nicht mehr verfolgt worden sind. Den Abschluss bildet ein Resümee über die Arbeit und eine Anleitung zur Inbetriebnahme.

2 Netzwerkmanagement

Netzwerkmanagement ist ein dehnbarer Begriff. Oberflächlich betrachtet ist es nach [InTeHa01] ein veralteter Protokoll-Analyzer, der Netzwerkaktivität überwacht. Die bessere Alternative wird als die Fusion verteilter Datenbanken und das selbstständige Pollen[1] von Netzwerkgeräten und Workstations betrachtet, aus denen Echtzeit-Grafiken der Topologie generiert, sowie der ermittelte Datenverkehr dargestellt werden. Objektiv betrachtet ist Netzwerkmanagement ein Service mit einer Sammlung von unterschiedlichen Werkzeugen und Anwendungen, die die Geräte erfassen und bei der Überwachung des Netzwerks unterstützen. Das Netzwerkmanagement unterstützt das Team der Administratoren und erlaubt die Durchführung automatisierter Vorgänge. Die Berücksichtigung der fünf Prinzipien des Netzwerkmanagements dient als Basis bei der Entwicklung eines Managementsystems. Den ersten drei Aspekten sollte, bezogen auf Netzwerkmonitoring, besondere Beachtung geschenkt werden. Bei der Entwicklung von Managementsystemen wird der Topologie ein hoher Stellenwert zugeordnet. Sie beschreibt den Aufbau des Netzwerks und seiner Komponenten und erlaubt dadurch Aussagen über Zusammenhänge der Geräte untereinander. Der Einsatz diverser Geräte wie Hubs, Bridges und Switches und derer unterschiedlicher Fähigkeiten, bieten unterschiedliche Möglichkeiten hinsichtlich des Managements, die es zu berücksichtigen gilt.
Ein Netzwerk in kleinem Ausmaß zu verwalten, ist nicht sehr aufwendig. Komplizierter wird es bei Netzwerken, die über ganze Gebäude oder sogar zwischen Städten verteilt sind. Gebäude haben Gebäudeswitches, die zum Einen die Anbindung nach Außen ermöglichen und zum Anderen den einzelnen Rechner innerhalb Stockwerke einen Netzanschluss bieten. Müssen Gebäude innerhalb einer Stadt oder zwischen Städten verbunden werden, ist man auf die Anbindung über die Telekommunikationstechnik angewiesen, da eine eigene Verkabelung mittels Twisted-Pair Kabeln, wie es in lokalen Umgebungen gängig ist, nicht mehr möglich ist. Die räumliche Ausdehnung verlangt den Einsatz mehrerer Mitarbeiter zur Pflege, um ein effizientes Reagieren im Fehlerfall zu ermöglichen. Um den Einsatz von Mitarbeitern, zur Betreuung der EDV in Außenstellen gering zu halten, ist ein automatisches System bei der Überwachung unerlässlich. Mit der Zunahme der räumlichen Ausdehnung und der damit verbundenen Zunahme an Bürofläche, nimmt unweigerlich die Menge der notwendigen Netzwerkkomponenten zu. Für jede Etage wird ein eigener Switches eingesetzt, der den vorhanden Computern den Netzwerkzugang ermöglicht. Jeder der Switches auf den Etagen, wird mit einem zentralen Switch für das Gebäude verbinden, der mit dem restlichen Netzwerk verbunden ist. Eine Infrastruktur alleine macht noch kein Netzwerk aus. Erst der Einsatz von Arbeitsstationen und Servern rechtfertigt ein Netzwerk. Je mehr Gebäude und damit verbun-

[1] Selbstgesteuerter bzw. selbst eingeleiteter Abruf von Daten. Bekannt geworden durch das Faxpolling, bei dem der Benutzer sich Informationen (über z.B. technische Geräte beim Hersteller) besorgen kann

dene Etagen zur Verfügung stehen, desto mehr Benutzer sind zu bedienen und an das Netzwerk anzuschließen. Der Administrationsaufwand nimmt mit steigender Rechner- und Benutzerzahl stetig zu und es wird eine Arbeitsteilung unter den Administratoren notwendig. Spezialisierungen in unterschiedliche Themen- und Aufgabengebiete ist notwendig, auch um die Zuständigkeiten abzugrenzen. Mit wachsendem Benutzerkreis nehmen die Anwender mit erhöhten Ansprüchen an die EDV zu. Desweiteren verlangt es in großen Arbeitsumgebungen nach Techniken, die die Anwender unterstützen und Arbeiten vereinfachen. Für eine größere Anzahl an Anwendern macht sich die Anschaffung von speziellen Softwaresystemen, wie Servern mit unterschiedlichen Diensten, deutlich schneller bezahlt, als in kleinen Unternehmen. Die Betreuung der Hard- und Software, die die Effektivität der Mitarbeiter erhöht, verlangt nach qualifizierten Betreuern. Ist ein ganzes Team mit der Betreuung der digitalen Infrastruktur beauftragt, nimmt die Komplexität zu. Ein Managementsystem gewinnt an Bedeutung, um die Informationen über das Netzwerk und den Diensten, den Systembetreuern zur Verfügung zu stellen. Teilen sich mehrere Mitarbeiter ein Arbeitsgebiet, ist eine gute Organisation notwendig und damit verbunden eine geeignete Kommunikation zwischen den Administratoren. Erfolgt keine organisierte Arbeitsteilung mit funktionierender Kommunikation, entstehen Fehler durch Informationsverlust. Die wichtigste Frage nach [Zenk99] bei der Auswahl von Netzwerkmanagement-Systemen ist die Frage nach den Anforderungen, die an das gesuchte System gestellt werden, und was damit erreicht werden soll. Weiter liest man, dass die Wahl der geeigneten Tools die vorhandene Hardware betrifft. Besteht ein Netzwerk nur aus Elementen eines Herstellers, ist die Wahl des herstellereigenen Überwachungstools die naheliegende Entscheidung. Die Software ist auf die Geräte angepasst und nutzt alle vorgegebenen Möglichkeiten. Es ist zu erwarten, dass die Software für den Administrator aussagekräftige, intuitive Informationen eines Gerätes preisgibt. Ist es der Software nicht möglich die Informationen bereitzustellen, die von Interesse sind, besteht meist keine Möglichkeit einer Anpassung der Software an die individuellen Bedürfnisse. Man ist den Vorgaben des Herstellers ausgeliefert oder genötigt teure Zusatzmodule zu erstehen.
Oftmals möchte, vor allem die Führungsebene eines Unternehmens, nicht von einer einzigen Firma bei Dienstleistungen und Produkten abhängig sein. Daher liegen teilweise sehr gemischte Netzwerke vor. Es entstehen inhomogene Systeme, in denen die Hardware unterschiedlicher Hersteller das Netzwerk bilden. Die Geräte halten sich an Standards in der Datenübertragung, damit die Funktionalität gewährleistet ist. Es liegt im Interesse einer Firma ihre Produkte auf dem Markt zu verkaufen und daher sollten sie mit anderen Herstellergeräten kompatibel sein. Wenn es an die Informationsextrahierung oder Steuerung der Hardware geht, muss sich der Hersteller nicht an Normen halten, wie etwa die Unterstützung durch SNMP (vgl. Abschnitt 3.5.1). Beschränkt sich ein Hersteller nicht nur auf Standards, können Eigenentwicklungen implementiert werden, die unter Umständen die Hardware besser unterstützen, als dies ein Standard tut, der viele Aspekte unter einen Hut bringen muss. Unter Umständen besteht bei Hersteller kein Interesse, dass die Konkurrenz in der Lage ist, mit ihrer Software die eigenen Geräte zu manipulieren. Jede Firma möchte seine Produkte verkaufen, nicht nur die Hardware, sondern auch die passende Software und dies ohne den Markt mit anderen teilen zu

müssen. Das erklärt die Schwierigkeiten hinsichtlich der Herstellung einer allgemeinen Software, die alle Komponenten vereint managen kann.
Die Managementsoftware soll nicht nur eine bestimmte Klasse von Netzwerkgeräte unterstützen, z.B. die Switches der Sterntopologie, sondern auch Hosts und Server, die von Interesse sind [Zenk99]. Herstellertools beschränken sich auf ihre eigenen Produkte und deren Fähigkeiten und werden nicht die Belange und Eigenheiten der Konkurrenz berücksichtigen. Betrachtet man hingegen die Software eines unabhängigen Softwarehauses, welches sich Mühe gibt, alle Eventualitäten zu implementieren, die von Interesse sein könnten, so stellt sich die Frage, ob das überhaupt möglich ist. Desweiteren hat ein freies Unternehmen sicherlich Probleme an gerätespezifische Daten zu kommen, wenn der Produzent der Hardware dies nicht möchte. In [Zenk99] wird auf die steigende Vielfalt der Protokolle hingewiesen. Einige Hersteller entwickeln eigene Protokolle, um spezielle Ideen zu verfolgen, andere setzen auf Standards. Ein Standard ist SNMP. Es ist leicht zu implementieren und wird von nahezu jedem managebaren Netzwerkgerät unterstützt. Etwas ins Abseits geraten, gewinnt es wieder an Bedeutung bei Administratoren [LinMag0306].
Netzwerkmanagement bezieht sich nicht nur auf passive Komponenten, wie Hubs, Switches oder Netzwerkdosen, sondern auch auf Server und Arbeitsplätze, die es zu managen gilt. Bei einer Unzahl von Computern verliert der Verwalter schnell den Überblick. Ein Managementsystem zur Überwachung, übernimmt einen beachtlichen Teil der Arbeit eines Netzwerkverwalters. Auf Seite 247 in [Zenk99] wird die Bedeutung wachsender Anforderungen und Möglichkeiten in drei Punkten herausgestellt:

Die permanent steigende Zahl der Netzwerkstationen und deren Ansprüche an das Netzwerk hinsichtlich Übertragungsraten. Viele Programme werden nicht mehr als lokale Installation genutzt, sondern auf Grund der Wartungsfreundlichkeit als Remotesystem mit zentralem Server. Der Installationsaufwand der Clients kann verringert werden und nur die zentrale Einheit, die die Rechenarbeit übernimmt, ist zu pflegen. Desweiteren gewinnen *Remote Desktop Arbeitsplätze* wieder an Bedeutung. Der Vorteil liegt wiederum in der zentralisierten Wartungsarbeit. Der lokale Arbeitsplatz ist lediglich für die Darstellung auf dem Bildschirm verantwortlich. Deutlicher Vorteil von Remote Desktops, ist die geringe Geräuschentwicklung des Arbeitsplatzes, da die rechenintensiven Arbeiten, die die Temperatur der Prozessoren in die Höhe treiben, vom Server erledigt werden und die Datenspeicher ebenfalls extern untergebracht sind. Notwendig ist ein Bildschirm mit Eingabegeräten und eine stabile Netzwerkverbindung.

Die Komplexität eines permanent wachsenden Netzwerks. Daran beteiligt sind nicht nur die stetig wachsende Zahl der Arbeitsplatzrechner und Switches. Sondern auch wandelnde Netzwerktechniken, wie Funkübertragung. Wer hätte vor 10 Jahren an Funkübertragungen gedacht, die heute 5 mal schneller als damalige Netzwerkinterfaces sind. Die Topologie muss um Accesspoints erweitert werden und das sind nicht wenige, um eine flächendeckende Erschließung der Gebäude zu gewährleisten. Der Vorteil ist die enorme Reduzierung des Kabelaufwands. Des weiteren ist bei Änderung von Spezifikationen eine Neuverlegung von Kabeln überflüssig.

Ein Netzwerk kann in keinem Zustand ewig verweilen. Die rasante Entwicklung der Technik, die den Anforderungen der Arbeitsplätze gerecht werden muss, verlangt permanent nach Erneuerungen. Während 1995 noch 10MBit BNC Karten verkauft worden sind, und heute die GBit Twisted Pair Karte für 10 EUR im Supermarkt zu bekommen ist, und erste Trunks bereits bei 10GBit angelangt sind, stellt sich die Frage, auf welche Technik man setzt, um der rasanten Entwicklung standzuhalten.

2.1 OSI-Prinzipien des Netzwerkmanagements

Aufgrund dieser Punkte, hat nach [Zenk99] die International Organisation for Standardization (OSI) eine fünf Punkteliste veröffentlicht, die einige Prämissen bzw. Anforderungen vorstellt, die ein Netzwerkmanagementsystem erfüllen sollte. Diese Behandeln das Fehler- und Performanzmanagement sowie den Einsatz von Konfigurationen. Desweiteren werden die Aspekte der Sicherheit und des Zugriff und der Verfügbarkeit behandelt. Die Berücksichtigung dieser Vorgaben erlaubt die Entwicklung eines Systems, das den derzeit gegebenen Anforderungen gerecht wird. Das nach diesen Angaben entstehende System verspricht die derzeit bestmögliche Unterstützung der Administration durch ein Softwaresystem.

2.1.1 Fault-Management

Das Fault-Management deckt den Bereich ab, den die meisten Leser mit Netzwerkmanagement in Verbindung bringen, dem Erkennen, Lokalisieren und Beheben von Fehlern. Ab einer gewissen Netzwerkgröße gestaltet sich die Suche nach der Fehlerquelle deutlich aufwendiger. Vor allem nehmen die Abhängigkeiten von Geräten zu, so dass es zu überprüfen gilt, ob nicht Fehler entdeckt werden, die von einem Anderen abhängig sind. Wenn es nur eine Abhängigkeit betrifft, dass der Ping[2] von der Arbeitsstation des Administrators den Switch nicht erreicht, ist zu prüfen ob und welche anderen Switches dazwischen liegen, die die eigentliche Ursache des Nichterreichens sein können. Es gilt festzustellen, wo genau der Defekt aufgetreten ist. Vermutungen anzustellen sind der falsche Weg, denn der Kern des Fehlers kann weitere Knoten beeinflussen. Ist die Problemstelle gefunden, müssen Abhängigkeiten geprüft werden, die von diesem Fehler in Mitleidenschaft gezogen werden. Je nach Fehler oder Bedeutung der abhängigen Komponenten, ist vor der Reparatur besser eine schnellere Alternativverbindung zu dem abgetrennten Netzwerk zu erstellen, damit die angeschlossenen Benutzer weiter arbeiten können, wenn auch Einschränkungen in Kauf genommen werden müssen aufgrund schwächer ausgelegter Übertragungsleitungen. Je nach Bedeutung der Abhängigkeiten ist über eine fest eingebaute Alternative nachzudenken. Ist das Netzwerk vorerst wieder verwendbar, kann der Fehler analysiert und behoben werden und das Netz wieder in den ursprünglichen Zustand versetzt werden. Es gilt also nicht nur Fehler zu finden und zu reparieren, sondern auch schnelle Problemlösungen in der Hinterhand zu haben, die keinen vollwertigen Ersatz der ausgefallenen Leitungen darstellen müssen, aber das Netz-

[2]Programm welches unter Verwendung des ICMP-Requests die Erreichbarkeit eines Computers testet.

werk in Betrieb halten können. Nicht nur die Komponenten, die die Infrastruktur der Netzwerks bilden sind von Interesse, auch die Server, die Dienste bereitstellen gilt es zu überwachen. Der Ausfall eines Servers oder seiner Dienste sind ebensowenig erwünscht. Eine gutes Systems lässt es gar nicht erst zu einem Ausfall kommen, sondern weist durch geeignete Prüfungen auf eventuelle Probleme oder Engpässe hin. Dafür sind regelmäßige Prüfungen notwendig, die den Netzverkehr auf jeden Fall durch Datenverkehr zusätzlich erhöhen, aber dafür einen Vorteil ergeben, die einen Komplettausfall verhindern können, der in jedem Fall schlimmer ist.

2.1.2 Configuration-Management

Netzwerke bestehen aus vielen einzelnen Einheiten wie Switches, Bridges, Router sowie vielen Hosts und Servern. Die Zusammensetzung ist nicht lange in einem festen Stadium, sondern ist von Dynamik geprägt und verlangt permanent nach Anpassung. Veränderungen sind notwendig, sollten aber nicht negativ bemerkt werden. Dabei muss es möglich sein, ein Netz zu reorganisieren, ohne dass Benutzer davon in Mitleidenschaft gezogen werden. Dazu vergleiche man die Reorganisation im Verfahren des Rapid Spanning Trees (vgl. Abschnitt 3.6 auf Seite 51) mit dem des ursprünglichen Algorithmus. Ohne Beeinträchtigung der Verfügbarkeit des Netzes wird eine neue Konfiguration für alle Geräte erstellt und eingesetzt. Der Vorgang ist für den Anwender unmerklich. So ähnlich verlangt es auch die Administration eines lokalen Netzes, dass im laufenden Betrieb Änderungen möglich sind, die die Verfügbarkeit und die Leistungsfähigkeit des Netzwerks nicht beeinträchtigt. Hinsichtlich des Monitoring haben Veränderungen unmittelbar und unauffällig zu sein. Die Integration in das bestehende System sollte problemlos, idealerweise vollautomatisch erfolgen. Das Management hat sich auf Veränderungen dynamisch anzupassen.
Hinsichtlich der Softwarewartung sollten aufwendige Konfigurationsarbeiten dem Administrator erspart bleiben. Dazu gehört auch die Konfiguration ähnlicher oder gleicher Bereiche. Vorlagen, die in Einzelkonfigurationen automatisch eingebunden werden erleichtern die Arbeit deutlich und verhindern *Copy und Paste*[3] Fehler. Veränderungen an Templates propagieren sich in alle abhängigen Konfigurationen. Es ist zu empfehlen, erfolgreiche Konfigurationen zu archivieren und zu protokollieren, um als Anleitung für zukünftige Aufgaben zu dienen.

2.1.3 Performance-Management

In einem Netzwerk sind viele Computer angeschlossen. Normale Arbeitsplatzrechner unterscheiden sich bei der Nutzung von Servern in benötigter Leitungskapazitätund den Zeitpunkten- und räumen. Viele Clients nutzten einen Server. Dem zu Folge wird die Datenleitung zu dem Server deutlich stärker belastet, als die einzelnen Leitungen der Clients. Server nutzen das Netzwerk je nach Anwendungsbereich mit völlig unterschiedlicher Datenaufkommen. Einzelne Server verwenden das Netzwerk ununterbrochen mit

[3] Durch einmaliges Schreiben, einmaliges Kopieren und mehrfaches wieder Einfügen, verbreitet sich ein einmal gemachter Fehler auf alle Kopien.

kleinen Datenpaketen, andere verarbeiten und senden stoßweise extrem große Datenmengen. Der Durchsatz in einem Netzwerk muss auf die Notwendigkeiten des Einsatzes angepasst werden. Das eine Segment belastet das Netz beispielsweise kaum, da lediglich einfacher Emailverkehr entsteht, der andere Bereich ist ausschließlich von Serverdiensten abhängig und ein drittes Segment bedient eine Unzahl an Arbeitsstationen, wie etwa öffentliche Rechnerräume der Hochschule, in denen Benutzer sich pausenlos an und abmelden, wobei jedesmal eine Unmenge an Daten über die Leitungen kopiert werden. So ist auf die Koppelung der Switches zu achten, dass Datenpakete keine unnötigen Strecken zurücklegen und Bereiche belastet, die bei geeigneter Anordnung umgangen hätten werden können. Monitoring hat zur Aufgabe, den Datenverkehr sichtbar zu machen, um brauchbare Anpassungen zu ermöglichen. In [Zenk99] wird nachstehender Aufzählung eine wesentliche Bedeutung zugeschrieben.

- Auslastung des Netzwerks
- Datenverkehr der einzelnen Stationen
- Hat der Gesamtdurchsatz ein akzeptables Maß?
- Wo liegen Engpässe?
- Wie steht es um die Antwortzeiten?

Bezüglich dieser Punkte verlangt es nach Schwellwerten, die bei Überschreitung, ein Zeichen zum Handeln für die Administratoren sind. Eine immer wiederkehrende Beschwerde von Anwendern ist es, die Trägheit des Netzes zu bemängeln. Eine Aussage in der Form bringt dem unvorbereiteten Administrator in arge Bedrängnis. Damit auf Hiweise vermeintlicher Netzwerkmängel entsprechend reagiert werden kann, sind Meßwerte von aktuellen Zuständen und auch aus vergangenen Zeiten ein wichtiges Entscheidungsmerkmal für eine sachliche Antwort. Beobachtungen über einen längeren Zeitraum ermöglichen eine rechtzeitige, bessere Skalierung des Netzes und dienen als Prävention gegen Fehler und Ausfälle. Performancedaten aus dem Monitoring erlauben Aussagen über die Leistungsstärke des Netzes und ermöglichen rechtzeitige Entscheidungen hinsichtlich der Anpassung.

2.1.4 Accounting-Management

Accounting Management gewinnt an Bedeutung, sobald das Netzwerk von mehreren Parteien genutzt wird, die voneinander unabhängig sind. Wird ein Netz innerhalb eines Gebäudes durch viele Unternehmen genutzt, müssen die Kosten umgelegt werden. Dies kann pauschal geschehen, aber auch in Abhängigkeit der Nutzung. So ist es möglich durch Überwachung der Switches, Ports oder VLANs die Datenlast von bestimmten Gruppen zu messen und dann nach einem Kostenmodel abzurechnen. Der zweite Einsatzart ist innerhalb eines Unternehmen zu finden, in denen Abteilungen untereinander Kostenstellen haben, die anteilig an der Finanzierung des Netzwerkes beteiligt sind. Die dritte Situation entsteht durch Beauftragung einer externen Firma innerhalb des eigenen

Hauses, die sich um das Netz kümmert. Über ein Accounting ist es u.a. auch möglich, eine effektive Nutzung des Netzes der Anwender zu protokollieren. Bei Auffälligkeiten ist es der Administration möglich, die Anwender zu unterstützen und ihre Effizienz, unter Zuhilfenahme des Netzwerks, zu steigern. Desweiteren ist es leichter, den zukünftigen Ausbau des Netzes zu planen, wenn die einzelnen Aktivitäten der Anwender bekannt sind. Die Erfassung der Daten einzelner Anwender birgt ebenso Risiken wie Vorteile, denn es besteht die Möglichkeit über diesen Weg die Mitarbeiter zu kontrollieren und zu überwachen. In diesem Fall sollte die Zustimmung des Betriebsrates, und falls vorhanden, des Datenschutzbeauftragten, eingeholt werden.

2.1.5 Security-Management

Auch die Sicherheit im Netzwerk ist über Netzwerkmanagement auf dem aktuellen Stand zu halten. Die Erzeugung, Aktualisierung und Verteilung der Sicherheitsinformationen muss über die komplette Infrastruktur gewährleistet sein. Ein großes Netzwerk umfasst sehr viele Rechner, die eine Superuserkennung[4] haben. Bei einem Personalwechsel muss gesichert sein, dass das ausscheidende Mitglied keine Möglichkeit mehr hat, sich dem System Zugang zu verschaffen. Bei mehreren hundert Rechnern ist, unter Umständen, sehr viel Zeitaufwand aufzubringen, um alle Rechner auf einen neuen Stand zu bringen. Eine automatische Verwaltung der Sicherheitsrichtlinien sorgt für Abhilfe. Durch ein Security-Management wird die Nutzung sicherer gemacht. Vor allem ab dem Punkt, an dem benutzerspezifische Informationen gesammelt werden ist es unerlässlich. In der heutigen Situation bringen die Betriebssysteme ihre Sicherheitssysteme mit. In dem Fall obliegt es der Administration die Systeme sinnvoll zu verwalten, so dass trotz verschiedener Systeme die kleinst mögliche Basis geschaffen wird, um nicht allen Aufwand mehrfach betreiben zu müssen, sondern stabile Automatismen zur Verfügung stehen. Betrachtet man weniger die Seite der Software, sondern die der Hardware, so ist auch eine Kontrolle der Netzwerkdosen sinnvoll. Es sollte nicht jedem möglich sein, jede frei Netzwerkdose in irgendeinem Büro nutzen zu können. Die Verwalter des Netzes sollten wissen, welcher Nutzer mit welchem Gerät im Netzwerk unterwegs ist. Unter Umständen ist es notwendig die Ports der Switches auf bestimmte MAC-Adressen zu begrenzen, um eine Kontrolle auf Hardwareebene zu bekommen. An der Universität Koblenz nutzt man dieses Verfahren, um nur Computern einen Zugang zu gewären, deren Nutzer eine gültige Kennung besitzen. Natürlich sollte das Sicherheitssystem der Switches und Server so eingerichtet sein, dass keine unautorisierte Person Einstellungen an der Hardware verändern kann.

2.2 Netzwerkmonitoring

Es gilt den Begriff Netzwerkmonitoring genauer zu definieren. Der Monitor ist der englische Begriff für den Bildschirm. Einem elektronischen Anzeigegerät für Daten, die ein Computer bereitstellt. Es geht prinzipiell lediglich um eine Tatsache, der visuellen Dar-

[4] Die Kennung, die alle Rechte hat und vor allem Einstellungen am System macht und die Software installiert und wartet

stellung von Informationen. Ob der Bildschirm in der Form des Computerbildschirms verwendet wird, oder eine riesiges Holzbrett mit einzelnen Lichtern, macht keinen Unterschied. Wichtig ist, dass der Betrachter in der Lage ist, das Bild zu interpretieren. Die Bahn verwendet seit Urzeiten ein Schaltpult, welches mit einer Zeichnung der Bahngleise ausgestattet ist und einer Sammlung von Lichtern. Durch entsprechende De- und Aktivierung von Schaltern, werden die Weichen gestellt. Dies wird an diesem Pult angezeigt, wonach der Mitarbeiter seine Entscheidungen für die Zugführung trifft. Es geht prinzipiell um Überwachung und Kontrolle.
Beim Netzwerkmonitoring ist das Überwachungsmedium das Computernetzwerk und die Dienste der Computer. Es sind die Zustände der Switches, Router und Server zu überwachen. Der Begriff Monitoring darf besonders in diesem Anwendungsszenario nicht mit dem Computerbildschirm in Verbindung gleichgesetzt werden. Es geht lediglich um die Repräsentation von Zuständen und die Bereitstellung der Information. Die Darstellung für den Anwender ist auf unterschiedliche Weisen möglich und nicht zwingend der Computerbildschirm. Es ist durchaus möglich, wenn auch nicht unbedingt sinnvoll, das Holzbrett der Bahn zu verwenden. Die Aufgabe des Netzwerkmonitorings auf die Darstellung zu beschränken, ist sicherlich nicht sinnvoll. Durch Einbeziehung von Computerprogrammen in die Darstellung besteht die Möglichkeit detaillierte Informationen abzurufen und Veränderungen anzustoßen. Bezüglich dieser Möglichkeiten ist der Einsatz eines Bildschirmes die richtige Wahl, wodurch ein beliebiges Bild darstellbar ist. In der Regel wird das Netzwerkmonitoring auf einem regulären Computerbildschirm in Kombination mit einem Computer dargestellt wodurch die Interaktion möglich ist.
Ein gutes Monitoringsystem beschränkt sich nicht auf die rein visuelle Darstellung. Dashat sonst zur Folge, dass ein Mitarbeiter ununterbrochen das System im Blick haben muss, um auf Veränderungen zu registrieren. Daher ist ein Benachrichtigungssystem zu integrieren, welches auf Veränderungen aufmerksam macht und den Anwender dazu animiert die neuen Zustände zu betrachten und dran eine Entscheidung festzumachen. Für einen voll automatisierten Prozess ist das reine Monitoring nicht gedacht. Das Monitoring soll als Unterstützung zur Handlungsweise des Anwenders dienen.

2.3 Namenhafte Managementsysteme

Befasst man sich mit dem Thema Netzwerkmanagement, fallen unweigerlich die Namen der großen Hersteller an Netzwerkgeräten wie Cisco, Hewlett Packard oder IBM. Es gibt eine Reihe weiterer Hersteller, diese drei jedoch, sind jedem Netzwerkbetreiber ein Begriff und sollten bezogen auf die Marktanteile die bedeutendsten sein. Es liegt daher nahe, nicht nur die Hardware der drei Großen in Betracht zu ziehen, sondern auch deren Software, die eine maximale Hardwareunterstützung verspricht. Hinsichtlich der langen Zeit, die der diese Firmen bereits Produkte für Computernetzwerke produzieren, ist die Spanne und Mächtigkeit der angebotenen Software recht groß. Die bekanntesten, kommerziellen Produkte für das Netzwerkmanagement sind:

CISCO Cisco Works

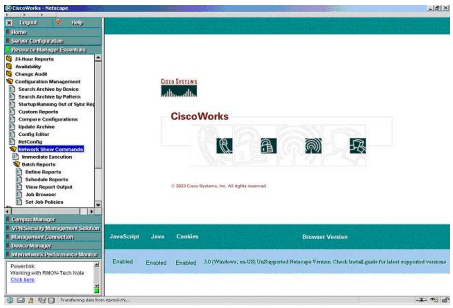

Abbildung 2.1: Bedienoberfläche von CiscoWorks

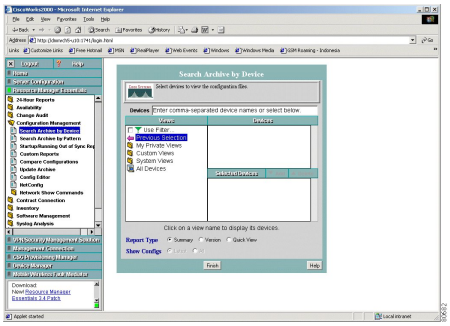

Abbildung 2.2: Bedienoberfläche von CiscoWorks

HP OpenView

IBM Tivoli

Microsoft Microsoft Operations Manager

Alle Produkte werden mit einer weitreichenden Vielfalt an Konfigurations- und Monitoringfunktionen beschrieben. Die Komplexität einiger Produkte ist so imens, dass die Produkte nicht in wenigen Sätzen zu beschreiben sind. Einige verlangen für einen produktiven Einsatz der Software eine mehrtätige Schulung. Es gibt von jedem Produkt Demosoftware, die aber sehr viel Zeit zur Einarbeitung verlangt, sodass der direkte Vergleich aus Zeitgründen nicht vorgenommen wurde. Die Beschreibung der Systeme erfolgt aus Sekundärquellen der Hersteller selbst aus dem Internet. Die Auswahl der Werkzeuge orientiert sich am geplanten Einsatzzweck „Netzwerkmonitorings“, welches auf dem Campus Koblenz in Zukunft erfolgen soll. Die Produkte der Hersteller sind alle deutlich mächtiger und gehen in der Komplettausstattung über die reinen Monitoringfunktionen hinaus.

Cisco Works

Die Antwort auf Netzwerkmonitoring aus dem Hause Cisco ist das System *CiscoWorks - Small Network Management Solution* (SNMS). SNMS besteht nach [CiscoSNMS06] aus mehren Modulen, die zusammen ein System darstellen zur Konfiguration, Darstellung und Fehlersuche für Switches und Systemanwendungen. Die Verwendung erfolgt nicht durch eine Applikation, sondern über ein Webinterface, wodurch eine Betriebssystemunabhängigkeit erreicht wird. Ein weiterer Vorteil eines Webinterfaces ergibt sich aus der Tatsache, dass das Tool jederzeit von einem beliebigen Rechner, in jeder Situation verwendbar ist. Ganz klarer Fokus ist die Unterstützung der hauseigenen Netzwerkgeräte. Die Steuerung und Modifikation der Geräte ist von einem zentralen Punk aus möglich. Selbst Updates der Switchbetriebssysteme ist über dieses Interface möglich. Die Software legt jedoch nicht nur Wert auf die Ciscokomponenten, sondern ist in der Lage Computer,

Abbildung 2.3: Webinterface Catalyst6506

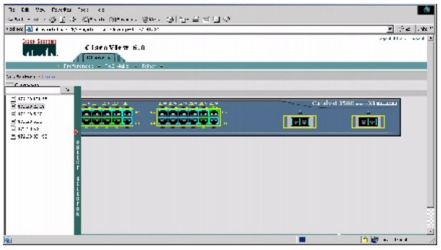

Abbildung 2.4: Webinterface Catalyst3500

Server und deren Applikationen zu verwalten und erlaubt so den Einsatz von nur einer einzigen Software, die ein Management einer inhomogenen Umgebung verspricht. Die folgenden drei Punkte beschreiben die Module von Ciscos SNMS:

Network Discovery SNMS ist in der Lage die Komponenten des Netzwerks zu erkunden und die Topologie grafisch darzustellen. Dazu wird das Programm *WhatsUp Gold* von Ipswitch[5] verwendet. *WhatsUp Gold* zeichnet sich dadurch aus, dass es nicht nur Switches und Router erkennen kann, sondern auch Arbeitsplätze und Server. Die Darstellung der Topologie selbst, kann unterschieden werden in ein physische und eine logische. Die Unterscheidung wird relevant, wenn physische Switchverbindungen durch Trunking oder den Spanning Tree (vgl. Abschnitt 3.6) logisch verändert genutzt werden. Desweiteren nutzt SMNS ein Benachrichtigungssystem mit Filterfunktionen zur Individualisierbarkeit der Nachrichten. Die Topologie- und Performanzdaten der Switches und deren Historie werden festgehalten und an die Partnerkomponenten weitergereicht.

Device Configuration Der CiscoView Manager wird von Cisco als die weitreichendst entwickelte Software angepriesen. Durch den Zugriff über einen Webbrowser bietet er von jeden Rechner aus Zugriff auf die Netzwerkgeräte. Ein interessanter Aspekt ist die photorealistische Darstellung der Switchkomponente im Webinterface, wie in Abbildung 2.3 und 2.4, und einer farblichen Repräsentation der belegten Ports und deren Status. Diese Funktionalität wird nur für Geräte des Marke Cisco angeboten. Anhand dieser Darstellung ist nicht nur der derzeitige Status abzulesen, sondern unmittelbar die komplette Konfiguration des Gerätes möglich. Die Verwendung von Telnet oder der Zugang über die Konsole ist nicht mehr unbedingt notwendig.

Device Management Das Herzstück des Managements ist der *CiscoWorks Resource Manager Essential* (RME). Das Tool bedient mehrere Anliegen. Es dient zur Statusdarstellung, als Datenspeicher für Konfigurationsdetails und als Plattform zur Generierung

[5] http://www.ipswitch.com/

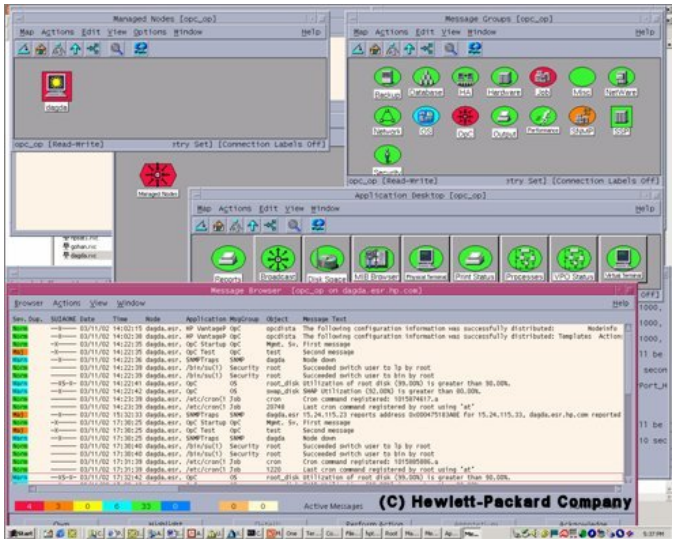

Abbildung 2.5: Übersich in HP OpenView

unterschiedlicher Handlungsregeln. Das Programm verwaltet das komplette Netzwerkinventar und verwaltet jeden einzelnen Datensatz. Es bietet eine Darstellung der Hard- und Software und erlaubt deren Konfiguration unter Berücksichtigung der aktuellen Veränderungen. Problemfälle innerhalb des LAN werden isoliert und zur Darstellung aufbereitet.

OpenView

Hewlett Packard stellt als Hersteller von Netzwerkkomponenten und Servern ebenso ein modulares System zur Verfügung. Die beiden für diese Arbeit interessanten Module sind der *Network Node Manager* und der *Performance Manager und Agent*. Das komplette Paket besteht aus:

- Network Node Manager (NNM)
- OpenView Operations
- Service Desk (SD)
- Business Process Insight
- Data Protector (DP)
- Performance Agent (OVPA)
- Performance Manager (OVPM)

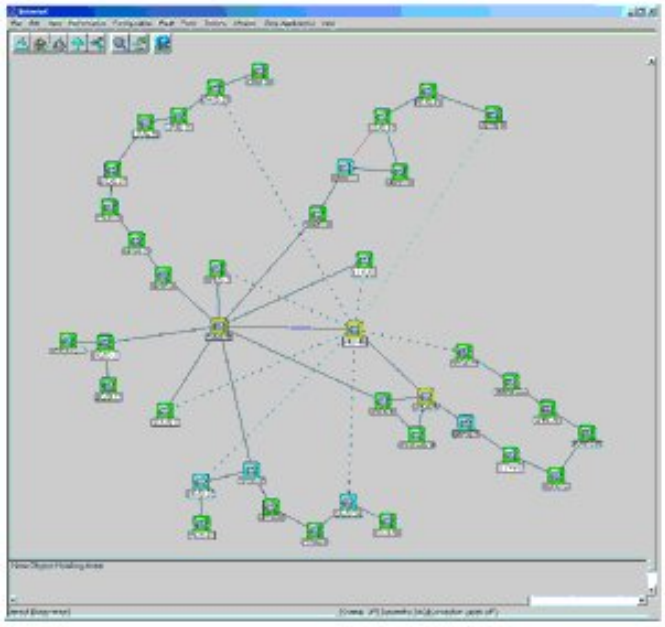

Abbildung 2.6: Topologiedarstellung in HP OpenView

Network Node Manager dient dazu Netzwerke jeder Größe managen und schließt auch komplexe Umgebungen mit heterogener Struktur ein. Er verfügt nach Angaben von HP über eine intuitive grafische Benutzeroberfläche (Abbildung 2.5), die jedem Anwender einen sicheren Umgang bieten soll. Dargestellt wird durch eine Topologiekarte, wie in Abbildung 2.6, eine Zusammenfassung des Netzwerkstatus sowie einem unmittelbaren Zugriff und ausführliche Informationen jeder Komponente. Aus der Beschreibung läßt sich erahnen, das eine Art Zoom auf beliebige Netzwerkbereiche der grafischen Darstellung möglich ist. Diese Ansicht zeigt nicht nur den Status der Netzwerkkomponenten, sondern auch alle Problemfelder auf einen Blick, um dem Anwender eine Entscheidungshilfe zu bieten, bevor daraus echte Störungen werden. Zu den intelligenten Features des *Network Node Managers* gehören eine lückenlose, automatische Fehlerfindung und -aufzeichnung sowie eine Reihe von Diagnose-Features. Die Problemlokalisierung wird über Servicechecks realisiert. Desweiteren liefert das Tool Informationen darüber, welche Komponenten nicht ausgelastet sind und welche bereits am Rande ihrer Kapazität arbeiten.[HPNNM06]

Performance Manager und Agent stellt ein Interface zur Verfügung, zur zentralen Darstellung und Analyse von Performanzdaten, um herstellerunabhängige Vorhersagen bezüglich der Hardwareresourcen anzustellen. Es erfolgt eine Unterteilung in Agent und Manager. Der Agent ist auf jedem System zu installieren. Dieser arbeitet autonom auf dem System, protokolliert und sammelt Daten, die bei Bedarf an den Manager gesendet

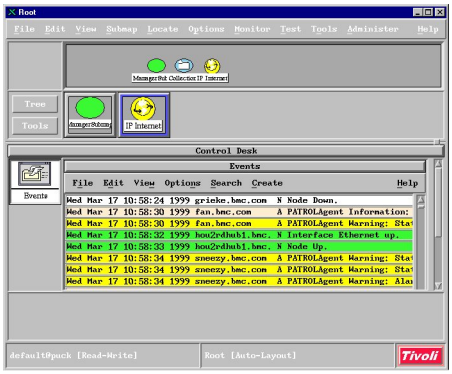

Abbildung 2.7: Statusanzeige in NetView

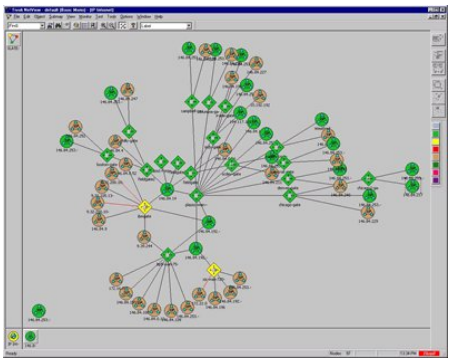

Abbildung 2.8: Topologiedarstellung in Netview

werden. Die Initiative der Datenübertragung startet nicht vom Manager, sondern vom Agenten. Der Manager dient, wie bei allen Produkten, der Darstellung und Speicherung der Daten auf unbestimmte Zeit, zur späteren Verwertung und Weiterverarbeitung. Es sind eine Auswahl an Schnittstellen für andere Tools vorbereitet.[HPPerform06]

Tivoli

Tivoli war einst eine eigenständige Firma, die 1996 von IBM gekauft worden ist. Heute beschreibt es ein eine Produktpalette von IBM, um Informationssysteme zu verwalten. Es ist ein modulares System, welches unter Anderem Rechner überwachen, Software installieren oder Datensicherungen durchführen kann.

Tivoli Monitoring

Das Modul Monitoring bietet die Möglichkeit, wichtige Systemressourcen zu überwachen und Engpässe sowie potenzielle Probleme zu zu erkennen. Zudem verfügt es über Ideen den vorherigen Zustand automatisch wieder herzustellen. Der Systemadministratoren muss die Leistungsdaten nicht selbst analysieren, sondern bekommt fertige Werte anhand derer eine Beurteiliung möglich ist. Tivoli verwaltet wichtige Unternehmens-Hardware und Software, einschließlich Middleware, Anwendungen und Datenbanken. Einen optischen Eindruck der Software vermitteln die Abbildungen 2.7 und 2.8 [IBMMonitoring06]

- Es werden Assistenten angeboten, die eine weitreichende, manuelle Konfiguration hinfällig macht.
- Benutzer können mit Hilfe von Assistenten Ressourcenmodelle entwickeln.
- Der Zugriff ist über einen Webbrowser oder eine Java-basierte graphische Benutzeroberfläche möglich.
- Überwachen aller wesentlichen Softwarekomponenten wie Serverdienste und Datenbanken.

Tivoli NetView NetView ist eine Verwaltungslösung für TCP/IP-Netzwerke. Es ist in der Lage das Netz zu identifizieren und die Netzwerktopologie darzustellen. Im weiteren Verlauf erfolgt eine Überwachung der Ereignisse der identifizierten Geräte. Diese werden verarbeitet und dargestellt. Es wird versprochen, dass die Verwaltungsarbeiten über die reine Geräteverfügbarkeit hinausgehe.[IBMNetview06]

- skalierbare, verteilte Verwaltungslösung
- Schnelle Diagnose der Ursachen von Netzwerkproblemen
- Integration in Anwendungen wie CiscoWorks
- Unterhalt eines Geräteinventars für die Verwaltung von Vermögenswerten
- Funktionen zum Test der Verfügbarkeit sowie Identifikation der Ursachen
- Möglichkeiten für die Problemsteuerung und -verwaltung
- Berichte für Tendenzen und Analysen

Tivoli Switch Analyzer Der Analyzer ist ein Netzwerkverwaltungs-Tool welches eine automatische Ereignisabstimmung und Ursachenanalyse bereit stellt, und über präzise und zeitgenaue Abstimmungsfunktionen sowie Ereignisverwaltungsfunktionen bereithält. Er wird in der Regel in der Kombination mit NetView eingesetzt, um über Netzwerkrouter und -switches die spezifische Ursache von Netzwerkausfällen zu ermitteln. Durch eine Abstimmung aller relevanten Netzwerkinformationen wird daraus die Grundursache des Problems auf den betreffenden Geräten ermittelt.[IBMAnalyser06]

- Automatische Layer-2-Identifizierung
- Identifizierung der Beziehung von Geräten untereinander, einschließlich Layer-2- und Layer-3-Geräten
- Filterung von Ereignissen bei Problemen, um die Ursache hervorzuheben.

Microsoft Operations Manager

Nach einem Artikel von [winkel03] ist der *Microsoft Operations Manager* (MOM) eine Überwachungstool für reine Microsoft©Windows Umgebungen (Abbildung 2.9). Er verlangt ähnlich wie *OpenView* die Installation eines Agenten auf dem zu überwachenden Gerät. Daher die mangelnde Unterstützung für alternative Betriebssysteme. Die Besonderheit sind so genannte Regelsätze, der Kern des Systems sind. Anhand der Regeln reagiert das Überwachungstool. Die installierten Agenten dienen der Sammlung von Daten und sind auf Grund der Herstellerverwandschaft zu den überwachten Systemen in der Lage sehr detaillierte Prozesse zu überwachen. Anhand der Regeln ist eine direkte Lösung des Problems seitens des Managers möglich, ohne eines Eingriffs durch den Administrator. Probleme werden zusätzlich per E-Mail an die Systembetreuer versendet, was eine

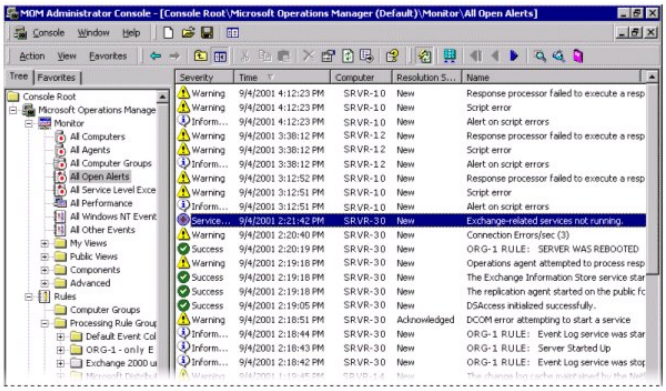

Abbildung 2.9: Microsofts Operations Manager

permanente Überwachung am Bildschirm überflüssig macht. Die Events werden in einer SQL-Datenbank hinterlegt und sind somit für eine Analyse bereitgestellt. Einen großen Vorteil bieten die vordefinierten Regelsätze von Microsoft, die spezielle Einsatzbereiche vorbereitet anbieten und nur auf die Gegebenheiten anzupassen sind. Das verspricht einen schnellen Einsatz des Systems. Dem Einsatzfeld aus dem Bericht von [winkel03] ist zu erahnen, dass der Einsatz von MOM auch für nicht Microsoft©Windowsserver möglich ist. Das verlangt dann aber die Konstruktion entsprechend aufwändiger Regelsätze.

3 Netzwerktechniken

Dieses Kapitel beschäftigt sich mit einem Auszug von Netzwerktechniken, die für das Verständnis dieser Arbeit unumgänglich sind. Es ist nur ein kleiner Auszug eines riesigen Themengebietes.
Topologie beschreibt den Aufbau eines Netzwerks, vergleichbar mit dem Straßennetz von und zwischen Städten. Es gibt eine Vielzahl an Topologien für Rechnernetze, die im Laufe der Zeit entwickelt worden sind. Von den bekannten Arten ist die geeignete zu wählen, die den meisten Nutzeffekt für die Anwender mit sich bringt und im gleichen Zug im finanziellen Rahmen bleibt. Diese Arbeit beschäftigt sich mit lokalen Netzen (LAN). Unterschiedliche LAN-Topologien verlangen unterschiedliche Netzwerkgeräte. Die Auswahl geht von einfachen Kabeln, die die Rechner direkt verbinden über Hubs und Bridges bis zu Switches. Desweiteren existieren verschiedene Übertragungsmedien mit unterschiedlichen Kapazitätsleistungen. In heutigen Netzen kommen in der Regel Switches zum Einsatz, die über Switches und Bridges realisiert werden und mit Hilfe eines Routers an weitere Netze angebunden werden. Switches gibt es in vielen Größen, bezüglich Portanzahl und Funktionalität, was sich deutlich im Preis bemerkbar macht, so dass für den jeweiligen Einsatz das richtige Equipment und die entsprechend angepassten Kabel zur Datenübertragung zu wählen ist.
Nicht nur die physischen Elemente spielen eine Rolle. Logisch müssen sich die Geräte verständigen, damit Datenpakete ihr Ziel erreichen. Computernetze verwendet Ethernet mit dem CSMA/CD Zugriffverfahren [PetDav2003]. Alle Netzwerkkomponenten müssen miteinander kommunizieren können. Daher sind geeignete Verfahren notwendig, die es ermöglichen, Informationen zwischen den Rechnern zu transportieren. Zu diesem Zweck existieren Protokolle, die eine Kommunikation der Geräte untereinander ermöglichen. Angelehnt an das Prinzip eines Briefes muss jedes Datenpaket von einem Rechner über ein unbekanntes, ständig sich veränderndes Wegenetz, seinen Bestimmungsort finden. Nicht nur die reine Möglichkeit der Kommunikation ist von Interesse, auch die Effizienz ist von Bedeutung. Daher wird der Spanning Tree vorgestellt, der eine schleifenfreie Kommunikation unter Switches ermöglicht, denn Schleifen in einem Ethernet-Netzwerk sind ein Grund für einen Totalausfall des Netzwerks.
Zu Beginn wird das OSI-Referenzmodell kurz erläutert, welches in der Rechnerkommunikation ein theoretisches Konzept zur Realisierung von Kommunikation bereitstellt. Auch wenn nicht jedes Protokoll aus dem OSI-Modell entstanden ist, so stellt es einen Leitfaden bereit zur Realisierung von Kommunikation zwischen zwei Rechnern. Es beschreibt auf sehr anschauliche Weise das Schichtenprinzip, anhand dessen die Unterscheidungen der unterschiedlichen Switches vereinfacht wird.

3.1 OSI Referenzmodell

Das OSI[1]-Referenzmodell ist kein Modell, welches für diese Arbeit die entscheidende Rolle spielt, doch verdeutlicht es Informationen, die in den folgenden Kapiteln verwendet werden. Das Modell beschreibt nach [InTeHa01] den Ablauf der Kommunikation von Rechnern über ein Netzwerk. Es besteht aus sieben Schichten, von der reinen Anwendung, welches die Interaktion mit dem Benutzer beschreibt, bis hin zur Bitübertragungsschicht, der physischen Verkabelung. Die Entwicklung erfolgte 1984 und beschreibt ein Architekturmodell anhand dessen sich eine Protokollimplementationen orientieren sollte. Das Modell unterscheidet sieben Gruppen mit eigenständigen Aufgaben. Es erfolgt eine Art Arbeitsteilung. Wichtig ist, dass jede Gruppe mit ihrem Aufgabenfeld in sich abgeschlossen ist. Jede Schicht stellt einen Dienst dar, der die höhere oder niedere Schicht bedient. Der Vorteil dieser Abgeschlossenheit liegt in der Implementierung. Jede Gruppe bietet Schnittstellen zu der nachfolgenden Gruppe an. So ist eine Neuentwicklung, Weiterentwicklung oder Optimierung einer einzelnen Schicht möglich, unabhängig von den restlichen Schichten, die in ihrer bisherigen Version weiter existieren können. Die Veränderung einer Gruppe darf die Funktionsfähigkeit aller abhängigen Komponenten nicht beeinträchtigen.

Schicht 7	Anwendung
Schicht 6	Darstellung
Schicht 5	Kommunikation, Sitzung
Schicht 4	Transport
Schicht 3	Vermittlung, Netzwerk
Schicht 2	Sicherung, Vermittlung
Schicht 1	physikalische Schicht

Die sieben Schichten lassen sich in zwei Gruppen zusammenfassen. Die drei oberen Schichten fassen den Anwendungsbereich zusammen, die dem Benutzer in der Regel als Anwendersoftware bekannt sind. Die Restlichen vier Schichten betreffen die Datenübertragung. Zum Beispiel sind die Implementationen von TCP oder IP, prinzipiell der Transportschicht bzw. der Vermittlungsschicht zuzuordnen. Auch wenn die beiden Protokolle vor der Standardisierung existiert haben, so lassen diese sich diese den beiden Schichten zuordnen.
Das OSI-Modell stellt den konzeptionellen Rahmen der Kommunikation, wobei die Realisierung durch Protokollimplementationen erfolgt. Die Protokolle kommunizieren zwischen zwei Rechnern A und B auf der gleichen Ebene des Referenzmodells. Bezogen auf den Menschen spricht man davon, dass zwei Personen auf dem gleichen Level/Ebene sind, wenn sie sich verstehen. Genauso verhält es sich mit den Protokollen in der Kommunikation zweier Rechner. In einem Protokoll werden nicht immer exakt eine Schicht alleine umgesetzt. Unter Umständen werden die Funktionen von ein oder mehreren Schichten des Modells zusammengefasst. LAN-Protokolle, wie Ethernet, arbeiten auf der physikalischen Schicht, die WAN-Protokolle hingegen berücksichtigen die drei untersten Schichten. Die Klasse der WAN-Protokolle umfasst zum Beispiel die Routingprotokolle, die die

[1] Open System Interconnection

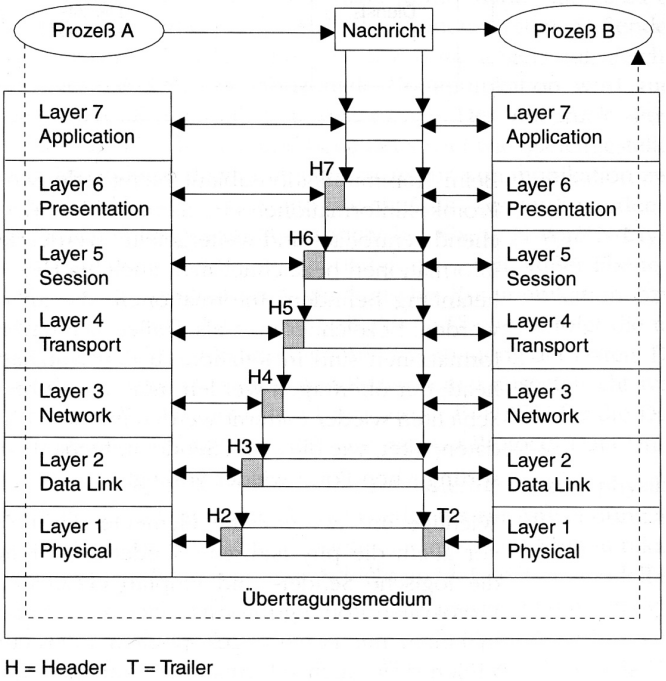

Abbildung 3.1: PeertoPeerKommunikation zwischen Sender und Empfänger; Grafik aus [Zenk99]

Pfade für den Verkehr zwischen lokalen Netzen festlegen. Netzwerkprotokolle, wie etwa HTTP und FTP, setzen auf Protokollen auf. So kann eine umfassende Neuentwicklung vermieden werden.
Die verschiedenen Schichten des Referenzmodells kommunizieren vertikal untereinander in Form von Diensten. Die höchste Schicht zum Beispiel, die Anwendungsschicht, verlangt die Übertragung von Daten, die keinerlei Bezug zu Netzen, Partnercomputern o.ä. haben. Ein Wust an Daten, mit denen ohne weitere Maßnahmen, keine geregelte Kommunikation möglich ist. Es sind Daten, die von A nach B zu kopieren sind und von der jeweiligen Anwendung zu verarbeiten sind. Aus diesem Grund werden die Daten in ein oder mehrere Pakete gepackt und mit einem Header versehen, der Angaben enthält, damit die Daten das Ziel im Netzwerk erreichen. Es findet einen Kapselung statt. Der Datenteil ist für die Versendung nicht von Interesse. Unter der Anwendungsschicht folgende Schichten bereiten die Daten auf, damit diese Netzwerktechnisch besser zu verarbeiten sind und reicht diese eine weitere Schicht weiter runter, bis an die Bitübertragungsschicht, die durch die Hardware die Daten versendet. Wie in Abbildung 3.1 gezeigt, wird in jeder Schicht ein weiterer Header an das bestehende Paket angehängt und das bisherige Paket gekapselt. Bildlich entspricht das Paket einer Zwiebel, die Schicht für Schicht neue Innereien enthält. An der gegenüberliegenden Stelle werden die Header in jeder Schicht wieder entfernt und die darin enthaltenen Angaben verarbeitet. Jede Schicht ist

für ein Teilgebiet zuständig und reicht das Paket nach erfolgreicher Bearbeitung eine Ebene hoch, bis die Daten für die Anwendung verwertbar sind. Durch dieses Prinzip ist der Anwendungsprogrammierer nicht genötigt zu seinem Programm die Kommunikation neu zu erfinden. Für diesen reicht die Anwendung einiger bereitgestellter Methoden, die die Daten erfassen und auf die Reise schicken. Weitere Vorteil dieses Modells ist es, dass eine Betriebssystemunabhängigkeit erreicht wird. Jedes Betriebssystem ist einmal gefragt die jeweiligen Anforderungen einer Schicht zu implementieren und für den Anwender bereitzustellen. Die Realisierung innerhalb der Methoden ist für die Kommunikation unerheblich, solange die Schnittstellen richtig bedient sind.
Die folgenden Beschreibungen erklären nach [Zenk99] die einzelnen Schichten und ihre Aufgaben. Es ist nicht notwendig, dass jede Kommunikation von der Anwendungsschicht des Rechners A bis zur Anwendungsschicht des Rechners B erfolgt. Die Kommunikation kann auch von Sicherungsschicht zu Sicherungsschicht zweier Geräte erfolgen. Das Prinzip bleibt erhalten, dass der Datenteil mit einem Header versehen wird und dies komplett bis zur unteren Schicht geleitet wird. Die Schichten 2 bis 7 kommunizieren logisch mit einander, da keine direkte Datenkommunikation auf dieser Ebene erfolgt. Lediglich die Sicht 1, die Bitübertragungsschicht, ist für den eigentlichen physischen Datenverkehr zuständig. Alle höheren Schichten sind auf die Schicht 1 angewiesen und müssen diese verwenden. Dies ist der Grund für die Verarbeitungsschritte, dargestellt durch den gestrichelten Pfeil, in Abbildung 3.1 für den Weg eines Datenpakets von Sender nach Empfänger. Anhand der Abbildung ist zu sehen, wie in jeder Schicht der Overhead des Datenpakets zunimmt. Zusätzlich ist in der Schicht eins ein Anhang an das Datenpaket zu erkennen. Dies ist eine Zusatzinformation für die Datenübertragung, die das CRC[2]-Feld darstellt.

Physical Layer Die Aufgabe ist die Übertragung des Bitstromes über die Leitung. Auf dieser Ebene gilt es Definitionen bezüglich der Kabelspezifikationen, Stecker und Pinbelegungen oder Widerstandswerte festzulegen. An dieser Stelle erfolgt die Festlegung der Datenflußrichtung, ob mit Voll- oder Halbduplex. Die Unterscheidung bestimmt, ob Daten parallel in beiden Richtungen gesendet werden können, oder immer nur in eine Richtung alternierend.

Link Layer ist für den zuverlässigen Austausch der Datenpakte verantwortlich, die durch die unterste, der Schichten, übertragen werden. Erkennung und Beseitigung von Übertragungsfehlern, sowie Synchronisation der ersten Schicht. Bei lokalen Netzen wird diese Schicht nochmals unterschieden in die MAC-Zugangsverfahren und die Logical-Link-Control (LLC).

Network Layer Diese Schicht ergänzt ein Datenpaket für die Übertragung über mehrere Stationen und gewährleistet einen reibungsloser Ablauf zwischen den logischen Verknüpfungen, vergleichbar mit IP-Adressen, an die Physischen, den MAC-Adressen,

[2]Cyclic Redundancy Check, dient der Überprüfung der Korrektheit des Datenframes, um Übertragungsfehler innerhalb der Leitungen zu erkennen. Eine Korrektur ist anhand der Prüfsumme nicht möglich.

<table>
<tr><th>OSI</th><th colspan="3">TCP/IP</th></tr>
<tr><td>Application</td><td colspan="3">Anwendungen</td></tr>
<tr><td rowspan="3">Presentation

Session</td><td>Standard:

TELNET
FTP
SMTP</td><td>Erweiterte Anwendungen:

NFS
Name Server
Drucker Server
Remote Execution
Terminal Server</td><td>System-meldungen

Fehlerbehandlung</td></tr>
<tr><td>Transport</td><td colspan="3">TRANSMISSION CONTROL PROTOCOL (TCP)
USER DATAGRAM DELIVERY PROTOCOL (UDP)</td></tr>
<tr><td>Network</td><td colspan="3">INTERNET PROTOCOL (IP)
ADDRESS RESOLUTION PROTOCOL (ARP)
INTERNET CONTROL MESSAGE PROTOCOL (ICMP)</td></tr>
<tr><td>Data Link
Physical</td><td colspan="3">Übertragungsmedium
802.3 Ethernet, 802.5 TRN, ProNET 4/10/80
Synchron, X.25/T1</td></tr>
</table>

Abbildung 3.2: Begriffe, die mit den OSI-Schichten korrespondieren; Grafik aus [Zenk99]

zu gewährleisten. Der Network Layer dient dem Auf- und Abbau der logischen Verbindungen, unter anderem der Wegesteuerung und der Flußsteuerung. Die Hauptaufgabe ist es, logische Wege unter den Knoten bis zu hin zum Ziel aufzubauen.

Transport Layer Die Intention der Transportschicht ist es, dem Benutzer eine Schnittstelle zu bieten, Daten über das Netzwerk zu schicken, indem eine einfache Programmierschnittstelle zur Verfügung gestellt wird. Dem Programmierer bleiben so die Details der Übertragung erspart. Zudem bietet die Schicht verschiedene Transportarten, wie in Abbildung 3.2 zu sehen ist. Durch diese Wahl bleiben dem Programmierer eine Auswahl an Transportqualitäten, die er wählen kann. Des Weiteren ist es möglich, von einem Knoten mehrere Transportverbindungen aufzubauen, um mehrere Programme zur gleichen Zeit bedienen zu können. Ein Kontrollsystem bei der TCP-Übertragung zum Beispiel gewährt die Zuverlässigkeit, auch bekannt als Sliding Window[3]. Die Transportschicht stellt die Verbindung der drei Anwendungsschichten zu den vier Datenübertragungsschichten dar.

Session Layer Diese Schicht dient der Aufrechterhaltung der Verbindung bei kurzzeitgen Ausfällen der Kommunikation. Sie stellt die Möglichkeiten zum Auf- und Abbau der Verbindungen den höheren Ebenen bereit.

Presentation Layer An dieser Stelle werden die Darstellungsformate der Nachrichten behandelt, um ein gemeinsam bekanntes Format zu erstellen. Bildlich stellt es eine Art Dolmetscher dar. Es entsteht eine neutrale Form, so dass die Daten von jedem Hersteller interpretierbar sind. Diese Schicht erlaubt es zum Beispiel Typen so

[3]Details zum Algorithmus sind unter [PetDav2003] nachzulesen

abzuwandeln, dass sie im vorliegenden Betriebsystem, ohne angabe des Datentyps, verwertbar werden. Die Implementierung des Zielsystems ist durch diese Schicht in der Lage, die ankommenden Daten zu interpretieren und für sich aufzubereiten.

Application Layer Sie stellt das Bindeglied von Netzwerk zu Anwendung dar. Sie steht für eine Reihe von Protokollen, die entsprechend permanent steigender Anforderungen, der Benutzer, wächst. Zu diesen gehören u.a. Dateitransfer und Management, Virtuelle Terminals, Nachrichtendienste oder Verzeichnisdienste.

3.2 Netzwerkvermittlungsgeräte

Diese Kapitel gibt eine Übersicht über Netzerkgeräte, die die Infrastruktur darstellen. Die Geräte werden an strategischen Punkten innerhalb der Geäude aufgestellt und durch Kabel verbunden. Die Computer in den Büros werde ebenfalls durch Kabel oder alternativ mit Funk an die Vermittler angeschlossen. Durch Weiterentwicklung der Geräte hat die Industrie von Entwicklung zu Entwicklung immer leistungsfähigere Vermittlungsgeräte auf den Markt gebracht. Die ersten Netze in kleineren Betrieben und privaten Haushalten sind durch mit der Bustechnik erfolgt, die noch keine Vermittlungsgeräte verwendet hat. Jeder Rechner ist mit einer Netzwerkkarte ausgestattet und wird durch ein Koppelelement an ein Koaxialkabel angeschlossen. Prinzipiell waren beliebig viele Rechner so zu verbinden. Einzige Beschränkung erfolgte durch die Länge des resultierenden Kabels. War das Kabel zu lang, reichte das Signal nicht aus um jeden Rechner zu erreichen.

Hubs Die erste Stufe der Entwicklung präsentierte den Hub. Die Idee der Telefontechnik mittels Twisted-Pair-Kabel[4] Rechner zu verbinden [Zenk99] hat zum Einsatz von Hubs geführt, einem, nach heutigen Gesichtspunkten relativ primitiven Komponente, deren Funktionsfähigkeit auf die Bitübertragungsschicht beschränkt. Der Hub dient gleichzeitg als Einklinkpunkt (Mehrportrepeater) für Computer. In [Harnisch02] wird der Hub, auf Grund seiner Betriebstechnik auch als Layer1-Switch bezeichnet. Die Arbeitsweise ist vergleichbar mir der, der Bustechnik. Es darf nur ein Computer ein Datenpaket in die Leitung schicken und der Hub sendet dieses Paket an alle verfügbaren Ports weiter. Jeder Computer, der an den folgenden und jeden weiteren Hub angeschlossen ist, wird das Datenpaket empfangen. Prinzipiell ist das Verfahren funktionsfähig, wenn auch mit Seiteneffekten. Da ein einmal abgesendetes Paket jeden Hub erreichen wird, wird auch jeder angeschlossene Rechner eine Kopie dieses Datenpaketes erhalten. Alle Rechner, bis auf den beabsichtigten Empfänger, werden das Paket in Empfang nehmen, prüfen und anschließend verwerfen, da es nicht für sie bestimmt ist. Unter der Annahme, dass ein Netzwerk aus sehr vielen Rechnern besteht, ist zu erwarten, dass die Rechner zu einem Großteil damit beschäftigt sind Pakete zu betrachten, die nicht für sie bestimmt sind. Die Abbildung 3.3 verdeutlicht das Problem in einem sehr kleinen Netzwerk, bei dem

[4]Kabel, in denen die Adern paarweise verdrillte sind.
[4]vgl. Abschnitt 3.1

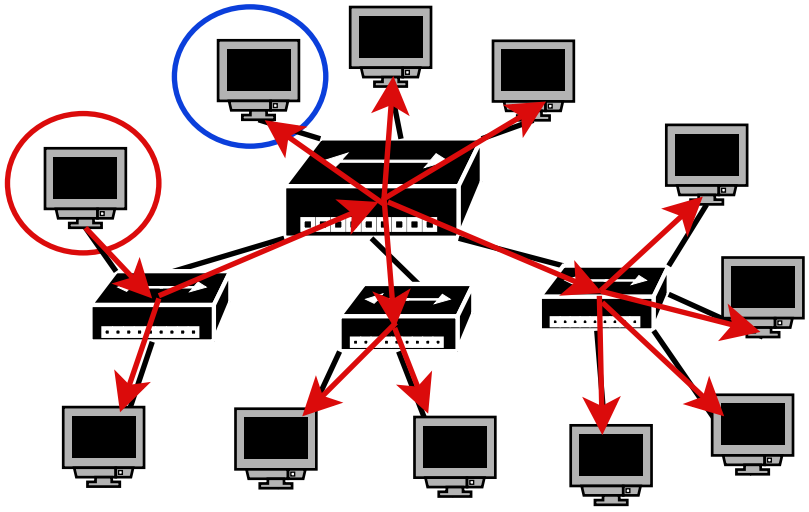

Abbildung 3.3: Paketflodding in einem ungeswitchten Netzwerk

der Weg eines Datenpaketes von dem rot markierten Rechner, zu dem blau markierten Computer gezeigt wird, und welche Computer mit „Datenmüll„belästigt werden. Ein Verfahren, welches nicht gerade durch Effizienz glänzt.
Hubs sind über einen Uplinkport[5] kaskadierbar[6], sodass eine Erweiterung des Netzwerks möglich ist. In jedem Hub, den das Signal durchläuft, wird das Signal erneut verstärkt, wodurch große Netze möglich sind. Folglich steigt die Menge der möglich anzuschließenden Computer. Um so mehr Rechner vorhanden sind, die ihre Pakete auf die Datenleitung schicken, um so mehr müssen warten, da immer nur ein Rechner die Leitung nutzen kann. Rechner, durch Hubs verbunden, befinden sich in einer Kollisionsdomäne.

Bridge Eine erste Verbesserung erreicht man durch Verwendung einer Bridge. Die ersten Bridges sind Minirechner mit zwei Netzwerkinterfaces. Die Bridge ist ein Gerät mit einer gewissen Intelligenz, da sie in der Lage ist, die empfangenen Pakete zwischenzuspeichern und auch selektieren kann. Eine Bridge merkt sich die MAC-Adressen[7] der sendenden Stationen und speichert diese, inklusive einem Zeitstempel, für jeden Port in einer dafür vorgesehenen Tabelle. Erreicht ein Datenpaket eine Bridge, prüft diese, ob die MAC-Adresse des Empfängers bekannt ist. Ist die Adresse nicht bekannt, wird das Paket in das benachbarte Netz weitergeleitet. Zudem prüft die Bridge, ob die Senderadresse bereits in der Tabelle registriert ist. Das ist notwendig, um in Zukunft auf diese

[5] Spezieller Anschluss, bei dem die Verdrahtung überkreuzt ist, so dass eine direkte Kommunikation zweier Partner ermöglicht wird.
[6] mehrfache Hintereinanderschaltung
[7] Eine MAC ist die Hardwareadresse eines Netzwerkgerätes. Sie ist global eindeutig und dient zur Adressierung von Netzwerkpaketen in lokalen Netzen.

MAC reagieren zu können. Erkennt die Bridge, dass die Zieladresse im Netz liegt, aus der das Paket kommt, wird das Paket nicht in das Nachbarnetz weitergeleitet. Dieses Verfahren nennt sich transparentes Bridging. Für den Fall, dass ein Rechner von einem Netzbereich in einen anderen versetzt wird, dient der Zeitstempel der Bridge dafür zu prüfen, welche Information die aktuellere ist.
Die Bridge ist auf diese Weise in der Lage zu entscheiden, in welchem der beiden Netze, an der sie angeschlossen ist, ein Rechner mit einer speziellen MAC zu finden ist. Eine weitere wichtige Funktion ist das Zwischenspeichern der Datenpakete. Die Speicherfunktion erlaubt es, die Leitungsblockierung, durch das CSMA/CD-Verfahrens, zu entschärfen. Wenn das Datenpaket die Bridge erreicht hat, ist dieses für das Netz erfolgreich zugestellt und erlaubt das Senden eines neuen Paketes. Die Bridge hält das Paket bereit und meldet wie ein regulärer Rechner an das Nachbarnetz, das sie ein Paket ins Netz senden möchte. Es entsteht eine logische Trennung, oder auch als Kollisionsdomäne, des LANs[8]. So ist es möglich, dass Computer in unterschiedlichen Kollisionsdomänen gleichzeitig Daten absenden können. Ein weiterer Vorteil der Bridge ist, dass diese in der Lage ist, je nach installierten Modulen, zwei unterschiedliche Netztypen zu verbinden. So ist es möglich ein Ethernet mit einem Token-Ringnetz zu verbinden, da die Bridge in der Lage ist, die Datenpakete für das jeweilige Partnernetzwerk aufzubereiten. Solange nur Computer miteinander kommunizieren, die die Bridge nicht passieren müssen, kommt es zu keinen Kollisionen zwischen den beiden Domänen. Die Kommunikation darf man, in diesem Kntext, nicht mit einem Telefonat vergleichen, bei dem die Information eine Sammlung von Sätzen ist, die die Gegenstelle in einem Zug erhält und dann antwortet. Kommunikation bedeutet hier, die Übermittlung eines Datenpakets. Unter Umständen fließen im Bruchteil einer Sekunde viele kleine Pakete durch die Leitungen, von vielen unterschiedlichen Sendern, die beim jeweiligen angestrebten Empfänger zusammengesetzt werden. So ist es möglich, dass nacheinander jeder Computer Pakete ins Netzwerk sendet und es für den Benutzer eines Computers so aussieht, als würde alles sofort geschehen und er sei der einzige, der sendet. Dabei wechseln sich alle Stationen beim Senden permanent ab, sodass es dem Betrachter nur vorkommt, als würde alles direkt ins Netz gehen. Eine ausführliche Beschreibung ist in [PetDav2003] zu finden.

Switch - Layer2 Da die Bridge mit nur zwei Schnittstellen ausgestattet ist, ist diese in der Praxis nur bedingt netzwerkverkehr entlastend. Je nach Ausmaß des Netzes, ist eine unüberschaubare Menge dieser 2-Portbridges notwendig. Zudem ist es ein zusätzliches Gerät, welches Platz beanspruccht und Geld kostet. Die Vorteile der Bridge führten zur Entwicklung der Multiportbridge, dem Layer2-Switch, einer Kombination aus Hub und Bridge. Der Switch kombiniert die Technik einer Zweiport-Bridge mit dem Einsatzzweck eines Hubs. Hubs werden gegen diese Neuentwicklung ausgetauscht. Das Resultat ist eine deutliche Verbesserung der Performanz des Netzwerks. Die Datenpakete werden durch die in den Switches gespeicherten Adress-Informationen auf exakt einem Weg zum Ziel geleitet. Unbeteiligte Stationen empfangen keine, für sie nicht bestimmte Daten mehr, abgesehen von Broadcast-Paketen. Zudem kann jeder Computer deutlich früher senden,

[8] Local Area Networks

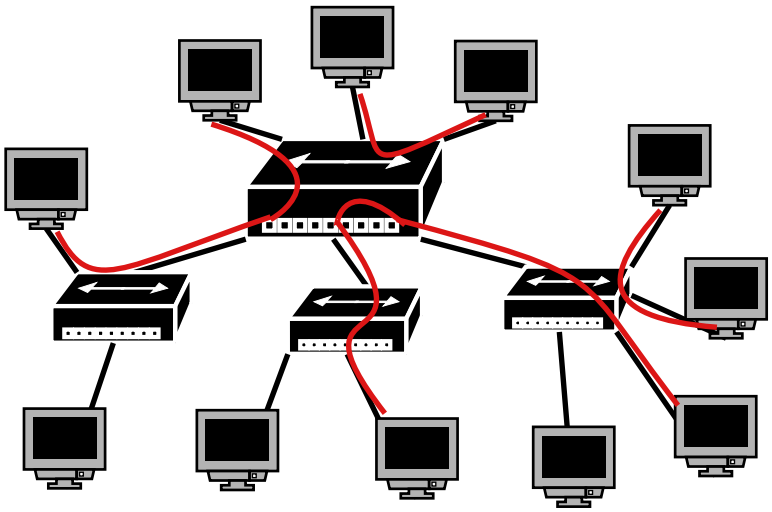

Abbildung 3.4: gleichzeitige Kommunikationsmöglichkeiten in einem geswitchten Netzwerk

da er eine eigene Kollisionsdomäne hat. Die restliche Arbeit erledigen die Switches unter sich. Bei über einen Hub verbundenen Computer, kann immer nur einer zu einem Zeitpunkt senden. Durch den Switch können nun vier Computer an vier Computer im selben Moment senden, da diese logisch in unterschiedlichen Kollisionsdomänen sind. Ein Beispiel, welche Computer gleichzeitig miteinander Daten austauschen können, ist in Abbildung 3.4 zu sehen. Im Gegensatz zu Abbildung 3.3, in der zu einem Zeitpunkt nur ein Rechner mit einem anderen kommunizieren kann, ist in Abbildung 3.4 zu einem speziellen Zeitpunkt die Kommunikation von vier Rechnern möglich. In einem Netzwerk aus Hubs aufgebaut müssen die Netzwerkteilnehmer immer warten, bis ein erkanntes fremdes Paket fertig gesendet wurde, anstatt direkt parallel zu den anderen Teilnehmern zu senden. Die Kommunikation würde also mindestens vier mal so lange dauern.

Switch - Layer3 Switches gibt es heute mit Fähigkeiten Datenpakete nicht nur der Ebene Layer2 zu verarbeiten, sondern bis in die OSI-Schicht sieben (vgl. 3.1) vorzudringen. Diese Switches sind deutlich leistungsfähiger. Günstige Switches, die im Hausgebrauch zu finden sind und in jedem Elektronikmarkt zu kaufen gibt, gehören in die Klasse der Switches, die Pakete auf Layer2 verarbeiten können. In der Literatur wird oftmals nicht genau genug zwischen Switches aus den Stufen Layer2 und Layer3 unterschieden. Layer2-Switches sind Geräte, die lediglich das Switching anhand einer MAC-Tabelle beherrschen. Größere Geräte mit 16 oder 24 Ports sind eventuell in der Lage einige Ports zu trunken. Oftmals wird der Spanning Tree (vgl. Abschnitt 3.6) mit diesen Switches in Verbindung gebracht. Das ist in dieser Form nicht ganz korrekt, denn diese einfacheren Switches nehmen am Spanning Tree nicht teil. Erst ab Geräten der Stufe Layer3, welche

IP-Datenpakete verarbeiten können, ist die Fähigkeit am Spanning Tree teilzunehmen, zu finden. Auch wenn die Verarbeitung lediglich Informationen von Layer2 benötigt, ist kein Layer2-Switch bekannt, der dies unterstützt. Die Spanning-Tree-Fähigkeit würde dem Hersteller höhere Kosten verursachen, welche an den Kunden weiterzugeben sind. In der Regel brauchen die Kunden dieses Fähigkeit nicht, wenn zum Beispiel in einem Privathaushalt nur vier Rechner vernetzt sind, um die Internetanbindung gemeinsam nutzen zu können. Auch der Einsatz in kleineren Unternehmen, die ein Netzwerk aus drei oder vier Switches, nutzen, wird die Verwendung des Spanning Trees keinen nennenswerten Vorteil bringen. In diesen Fällen ist der Datenverkehr zu gering. Erst in Institutionen, die mehrere hundert, gleichzeitige Anwender im Netzwerk haben, würde der Vorteil des Spanning Trees spürbar werden. Unternehmen dieser Größenordnung verwenden wiederum keine einfachen Layer2-Geräte, sondern können sich größere Hardware erlauben, die noch weitere Fähigkeiten bieten. Dies scheint der Grund zu sein, warum erst ab Layer3-Switches Spanning Tree zu finden ist.

3.3 Topologien

Eines der maßgeblichen Themen dieser Arbeit behandelt die Erkundung der Topologie eines Netzwerks. Der Begriff Topologie soll etwas genauer erläutert werden. Im Lexikon [Meyers] sind mehrere Begriff unter dem Stichwort *Topologie* beschrieben:

topo griechisch: topos, der Ort
Wortbildungselement mit der Bedeutung „Ort, Gegend, Gelände“

Topographie Teilgebiet der Geodäsie. Ermittlung der Abweichung der physischen Erdoberfläche welche in geeigneter Form darzustellen ist, sowie Charakterisierung des Geländes.

Topologie Teilgebiet der Mathematik, welches ursprünglich diejenigen Eigenschaften geometrischer Gebilde behandelt, die bei umkehrbar eindeutigen, stetigen Abbildungen erhalten bleiben (vgl. Homomorphismus). Heute bezeichnet als Theorie der topologischen Räume, das bedeutet solche Räume oder Punkmengen, die eine topologische Struktur aufweisen. Es werden Eigenschaften untersucht, die bei den Abbildungen erhalten bleiben. Die Ergebnisse sind besonders für die Funktionsanalyse von Bedeutung.

In einem Computernetzwerk beschreibt die Topologie die Verbindungen der teilnehmenden Geräte untereinander, die den Datenverkehr ermöglichen. Die richtige Wahl der Topologie und deren Pflege kann die Performanz[9] des Netzes beeinflussen. Bei Computernetzen ist zwischen physischer und logischer Topologie zu unterscheiden. *Physisch* behandelt Netzwerkgeräte und ihre Anordnung zueinander entsprechend der Kabelverbindungen. Der Begriff *Logisch* steht in dieser Arbeit für den Aufbau von Netzen in

[9] Performanz kennzeichnet die Leistungsfähigkeit eines Netzwerks und dessen Komponenten.

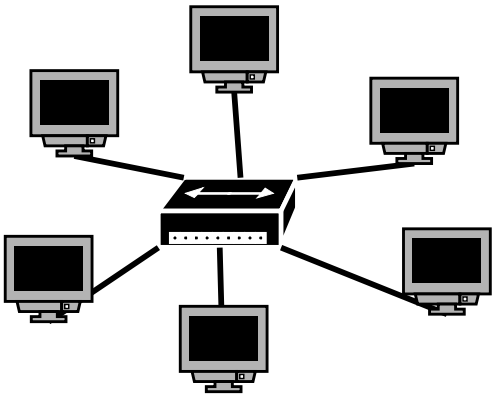

Abbildung 3.5: Sterntopologie

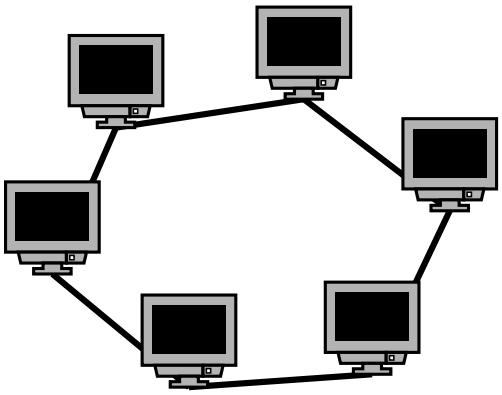

Abbildung 3.6: Ringtopologie

Teilnetze und welche Assoziationen für die Verwaltung dahinter steckten, etwa in Form von Subnetzen, die für jedes Gebäude eingerichtet sind oder spezielle IP-Adressbereiche für Arbeitsgruppen. Eine weiteres Beispiel logischer Netze beschreibt die Verwendung von VLANs, um Rechner logisch in Gruppen zusammenzufassen, damit eine Gruppe Rechner nicht mit einer Anderen in Kontakt geraten kann. VLANs nutzen das physisch gleiche Netzwerk, sehen einander aber nicht und bilden eine eigene Broadcastdomäne. Die Verbindung von VLANs erfolgt über einen Router.
Nachfolgend werden kurz einige physische Topologien genannt, wie sie in vielen Büchern über Netzwerke beschrieben werden:

Bus-Topologie Die war wohl die erste Art der Netze, die in kleineren Firmen und privaten Haushalten zu finden war. Ein Koaxialkabel bildet die Basis, den Bus, welches beliebig, bis zu einem gewissen Grad, verlängert werden kann. Stationen werden über T-Stücke an das Kabel angeschlossen. Fällt ein Rechner aus, wird der Rest des Netzwerks nicht negativ beeinflusst. Am Anfang und Ende des Kabels sind Widerstände angebracht, um Reflexionen zu verhindern. Alle teilnehmenden Rechner müssen sich dieses eine Kabel teilen. Durch Verwendung der Ethernettechnologie mit dem CSMA/CD-Verfahren hatte diese Topologie große Problemen mit Kollisionen. Es kann immer nur eine Station senden. Bei großen Netzen ein Nachteil. Die Weiterentwicklung von Twisted-Pair-Verkabelung und Entwicklung von Hubs brachte in dieser Problematik keine Verbesserungen. Erst die Weiterentwicklung zu Bridges und Multiportbridges (Switches) erlaubte effizientere Netzerkzugriffe. Eine große Problematik stellen desweiteren die Steckanschlüsse dar, die oft zu einem Ausfall des Busses führten. Ein Vorteil ist sicherlich der deutlich geringere Kabel- und Hardwareaufwand im Gegensatz zu der Stern- oder Baumtopologie.

Ring-Topologie In der Ringtopologie (Abbildung 3.6) wird jede Station mit zwei Netzwerkschnittstellen ausgestattet und je mit einer anderen Station verbunden. Es entsteht

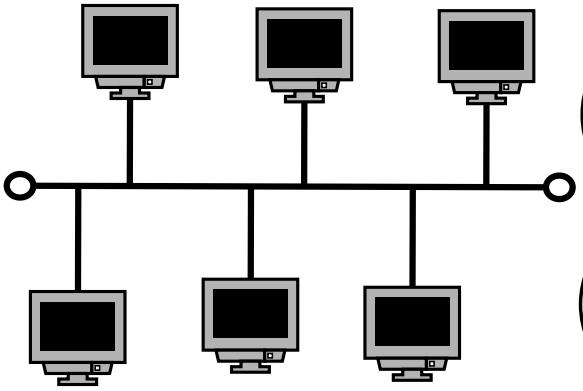

Abbildung 3.7: Bustopologie

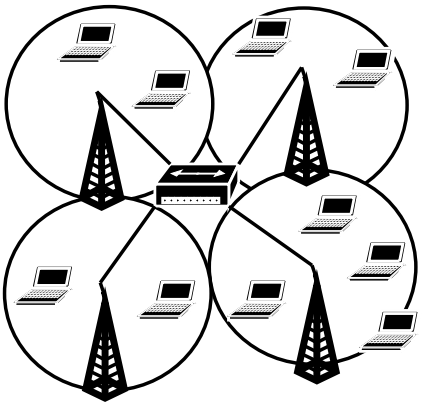

Abbildung 3.8: Zelltopologie

ein Ring über alle angeschlossenen Stationen. Jede Station arbeitet als Repeater[10], wodurch Netze über sehr große Strecken erzeugt werden können. Der Ring benötigt jedoch ein geeignetes Verfahren, um zu entscheiden, welche Station ein Datenpaket schicken darf. Für diesen Einsatzzweck wählte man einst das Tokenverfahren[11]. Kollisionen, wie sie aus Bus und Sterntopologien bekannt sind gibt es nicht. Dafür sind die Wartezeiten unter Umständen extrem lang. Desweiteren kann es bei einigen Arten von Ringen zu Ausfällen kommen, sobald nur eine Station ausfällt und der Ring unterbrochen ist. Neuere Ringe nutzen eine Protektion-Umschaltung, sodass der Computer nicht mehr als Repeater arbeitet und die Daten einfach durchgeschliffen werden. Ist die resultierende höhere Distanz bis zum nächsten Knoten zu lang, kann das Signal unter Umständen zu schwach sein.

Stern-Topologie Charakteristisches Merkmal der Sterntopologie sind kurze Wege, das bedeutet, dass zwischen Sender und Empfänger nur wenige Vermittlungsstationen (Hubs oder Switches) passiert werden müssen. Abbildung 3.5 zeigt eine klassische Sterntopologie, die eine Menge von Rechnern mit nur einem Vermittler verbindet. Es ist deutlich zu erkennen, dass nur exakt eine Station, der Switch, zwischen zwei kommunizierenden Computern ist. Die klassische Sterntopologie besteht aus einem zentralen Element und vielen Stationen, die mit einer Ende-zu-Ende Verbindung angeschlossen sind. Fällt ein Rechner aus, stört es die Kommunikation der übrigen Computer nicht, solange diese nicht mit der Ausgefallenen kommunizieren. Fällt das zentrale Element aus, so können alle Stationen keine Daten mehr austauschen. Dies ist die einzige Schwachstelle dieser Topologie. In kommerziellen Umgebungen verwendet man nicht nur einen Switch, sondern setzt einen weiteren ein, der im Falle einer Fehlfunktion des ersten, dessen Aufgaben

[10] Verstärker für das empfangene Signal

[11] Ein Token ist ein Marker, der den Ring der Netzwerkgeräte durchläuft. Wer den Marker hat, darf senden. Anderenfalls ist zu warten. Diese Idee gewährleistet, dass es keine Kollisionen bei den Sendern gibt

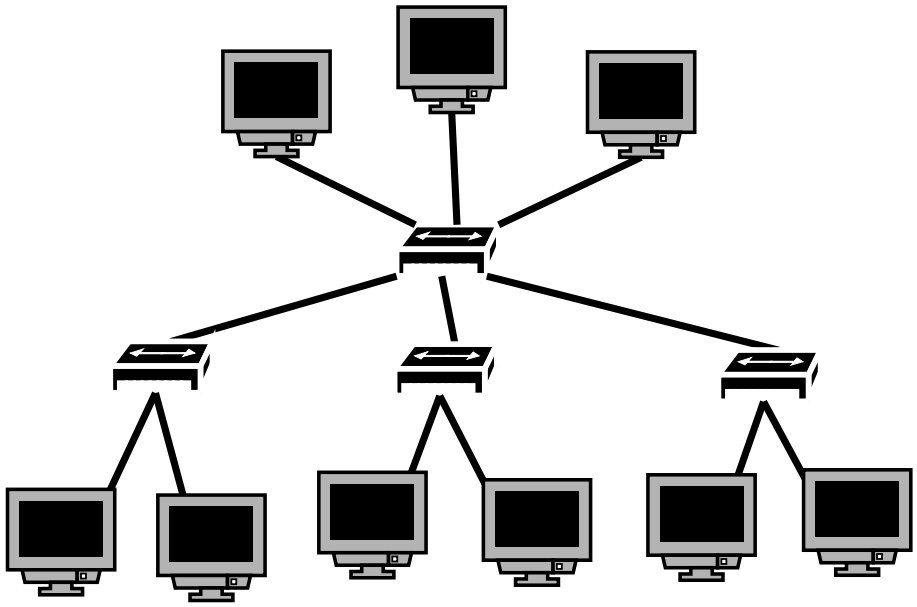

Abbildung 3.9: Baumtopologie

übernimmt.

Baum-Topologie Eine Erweiterung zum Stern stellt der Baum dar (Abbildung 3.9). Durch eine Ersetzung einzelner Rechner durch Switches wandelt sich der Sterne zu einem Baum. Die drei unteren Rechner aus Abbildung 3.5 werden in dem Abbildung 3.9 jeweils durch einen Switch ersetzt, an die wiederum Rechner angeschlossen werden. Bei steigender Rechneranzahl reichen die Ports eines einzigen Switches nicht aus und Switches müssen eine Möglichkeit bieten, das Netzwerk zu erweitern. Der Netzwerkverwalter ist gezwungen Switches zu kaskadieren, um mehr Ports zur Verfügung zu haben. Ersetzt man jeden Rechner aus Abbildung 3.5 gegen einen Switch entsteht ein balancierter Baum. Man könnte dies auch als erweiterten Stern betrachten. Eine serielle Anordnung aller Switches wäre eine Alternative, wenn auch keine gute. Die Datenpakete von den beiden weit entferntesten Geräten müssten alle Switches passieren, um miteinander Daten auszutauschen zu können. Bei der Baumanordnung erreicht man für jeden Rechner den Weg mit den wenigsten Switches. Derzeit ist dies die meist genutzte Netzwerktopologie. Prinzipiell wäre eine Verkabelung von jedem Switch zu jedem anderen Switch möglich. Man sollte meinen, dass so die kürzesten Strecken erreicht werden. Diese Verkabelung nennt sich Netztopologie oder vermaschtes Netz. Die Konstruktion ist aber keine geeignete Variante, wenn nicht die richtige Hardware eingesetzt wird, wie später in Abschnitt 3.6 erläutert wird.

Zell-Topologie Diese Art ist relativ jung. Durch die Entwicklung von Funknetzen entfallen die lästigen Kabel. Rechner können per Funk Daten mit einem Netzwerk austauschen. Einzige Notwendigkeit ist eine Punkt zu Punkt Verbindung zweier funkender Rechner oder der Einsatz eines Access-Points. Dieser fungiert als Vermittler zwischen den Funkgeräten und dem lokalen Netzwerk. Durch die begrenzte Funkleistung muss sich die jeweils funkende Arbeitsstation immer im Bereich des Access-Points aufhalten. Es entsteht eine Zelle um den Access-Point. Durch geeignete Montage der Access-Points erreicht man eine flächendeckende Erschließung mit mehreren Zellen. Ähnlich wie bei Funktelefonen, kann anhand bekannter Positionen der Accesspoints, der grobe Aufenthaltspunkt des Rechners ermittelt werden.

3.4 Adressierung in Netzwerken

Die Ansprechbarkeit aller Rechner im Netzwerk ist ein wichtiges Themengebiet. Datenpakete müssen mit entsprechender Informationen über Sender und Empfänger ausgestattet werden, um das vorgesehene Ziel zu erreichen. Dies geschieht mit Hilfe der MAC-Adresse, die für jedes Netzwerkinterface weltweit eindeutig ist. Idealerweise erfolgt die Datenübermittlung auf eine Weise, die keine festen Wege vorschreibt, sondern von Punkt zu Punkt eine Entscheidung getroffen wird, an welche Station weitergeleitet werden soll. Da die Vermittlung bei den Stationen selbst liegt und kein zentrales Verzeichnis den Pfad der Datenreise vorgibt, ist es möglich, dass unterschiedliche Datenpakete unterschiedliche Wege zum gleichen Empfänger nehmen. Das Netz entscheidet selbst, wie der Pfad zwischen zwei Partnern ist. Durch Auslesen der Sender- und Empfängeradresse weiß ein Switch für welchen Port ein Datenpaket bestimmt ist. Ist noch kein Eintrag vorhanden, wird das Datenpaket an alle Ports weitergeleitet, die für eine Verteilung von Paketen ohne bestimmtes Ziel freigegeben sind. Es entsteht die MAC-Adressentabelle, die für jeden Port geführt wird. MAC-Adressen sind in sechs Tupel unterteilt. Die Zahlen sind in Hexadezimalangabe notiert und erlauben daher $(16 * 16)^6 = 281.474.976.710.656$ verschiedene Möglichkeiten eine Netzwerkkarte zu identifizieren. Eine Beispieladresse des Rechners *GANDALF* wäre *00:C0:DF:11:93:FE*. Die ersten drei Tupel dienen als Herstelleridentifikationsnummer. In diesem Beispiel ist*00:C0:DF* der Code für die Firma Intel[12]. Große Herstellerfirmen besitzen mehrere Herstellernummern, da bei der Produktion von Netzwerkinterfaces diese mit dem Herstellercode und einer fortlaufenden Nummer versehen werden. Dieser Nummernbereich fortlaufender Zahlen ist irgendwann aufgebraucht und ein weiterer Herstellercode, für weitere Interfaces, ist notwendig. Eine MAC-Adresse ist für die Administration hinsichtlich des Computers nicht sehr aussagekräftig, da lediglich der Herstellername und eine fortlaufende Nummer codiert ist.
Bei der Entwicklung des IP-Protokoll entstand eine andere Art der Adressierung, die Adressen in Klassen zusammenfassen kann, die aussagekräftiger sind. Die IP-Adresse identifiziert einen Rechner, und dieser lässt sich damit ansprechen. Intern wird die IP-Adresse auf die MAC-Adresse abgebildet. MAC-Adressen sind für den Menschen nicht intuitiv lesbar. Ein brauchbarer Vergleich beschreibt die MAC-Adresse als die IDENT

[12] http://www.intel.com

Nummer des Personalausweises und die IP-Adresse bezeichnet die aktuelle Hausanschrift. Die Hausanschrift ändert sich bei einem Umzug, die IDENT Nummer bleibt ein Leben lang erhalten. Über die IDENT-Nummer würde niemals ein Postpaket seinen Empfänger erreichen. Das Prinzip der Postleitzahlen und Straßennamen erlaubt eine für die menschliche Denkweise nachvollziehbare Adressierung. Ähnlich arbeitet das IP Adressierungsschema, bei dem Zahlen von Punkten getrennt gewisse Bereiche definieren und dadurch eine Zuordnung möglich ist. Analog zu einem Wohnungswechsel kann sich die Zusammengehörigkeit einer MAC zu einer IP ändern. Die IP für eine MAC kann sich ändern. Um die Verwaltung der MAC-IP Beziehung automatisch zu verwalten werden DHCP-Server eingesetzt. Ein DHCP-Server teilt auf Anfrage eines Rechners seiner MAC eine IP-Adresse zu. Dem Server wird bei der Konfiguration ein Adressbereich zugeteilt, aus diesem die Adressen vergeben werden. Es existiert die Möglichkeit für spezielle MAC-Adressen immer eine bestimmte IP zuzuordnen, anderenfalls erhält der anfragende Rechner eine IP-Adresse aus einem frei zuordbaren Bereich.
Die Angaben, welche IP zu welcher MAC gehört, wird im ARP-Cache eines jeden Rechners gespeichert. Diese Information ist notwendig, da ein Sender mittels IP mit dem Empfänger kommuniziert. Da die Netzwerkgeräte auf Layer2-Ebene keine IP-Adressen verstehen, ist die MAC-Adresse notwendig. Aus diesem Grund wird das IP-Paket mit einem neuen Header versehen, der als Adresse nicht die IP, sondern die MAC enthält. Kommt das Paket über das Netzwerk beim Empfänger an, wird der Header entfernt und die höheren Schichten der Protokollimplementation kann anhand der IP die Daten weiterverarbeiten. Jeder Rechner verwaltet eine Tabelle mit Einträgen von IP- zu MAC- Adressen, um für die Kommunikation nicht jedesmal die benötigte MAC-Adresse nachzufragen. Erhält ein Rechner ein Netzwerkpaket, speichert er die Kombination IP und MAC in seinem Cache, um diese für etwaige späteren Verbindungen zur Verfügung zu haben. Hat ein Rechner mit vielen unterschiedlichen Rechnern Kontakt, nimmt die Menge der IP-MAC Paare stark zu und die Tabelle reicht unter Umständen nicht weitere Paarungen aufzunehmen. In diesem Fall werden länger nicht genutzte Informationen verworfen. In dem Fall, dass ein Rechner keinen MAC-Eintrag zu einer IP hat, prüft er im ersten Schritt, ob die Ziel-IP-Adresse in seinem eigenen Netzwerk liegt. Ist dies nicht der Fall, wird das Paket an den nächsten Router gesendet, da das Ziel offensichtlich in einem anderen Netzwerk liegen muss. Ist das Ziel im eigen Netzwerk, erfolgt ein Broadcast, um den Zielrechner zu erreichen. Der Zielrechner teilt dem Sender seine MAC mit. Eine detaillierte Beschreibung ist in [PetDav2003] nachzulesen.
Die Identifikation anhand der IP-Adresse stellt für den Anwender keinen wirklichen Komfort dar. Namen sind für den Menschen einprägsamer und bieten deutlich mehr Freiheit. Es erfolgt die Entwicklung des Domain Name Service (DNS). Eine Gruppe von Rechnern werden zu einer Domäne zusammengefasst und bei einem zugehörigen Server registriert. Dieser Server ist in der Lage natürlichsprachliche Namen auf IP-Adressen abzubilden. Das erlaubt dem Benutzer den Namen des Zielrechners, ohne Kenntnis über dessen IP-Adresse, anzugeben. Der DNS übernimmt für den Anwender das ersetzen des Namens durch die IP-Adresse. Das Datenpaket kann an die nächste Verarbeitungsschicht gereicht werden, bei der zum Beispiel der Header mit der passenden MAC aufgesetzt wird. Ein Name kann in einer Domäne nur einmal existieren. In disjunkten Domänen sind daher

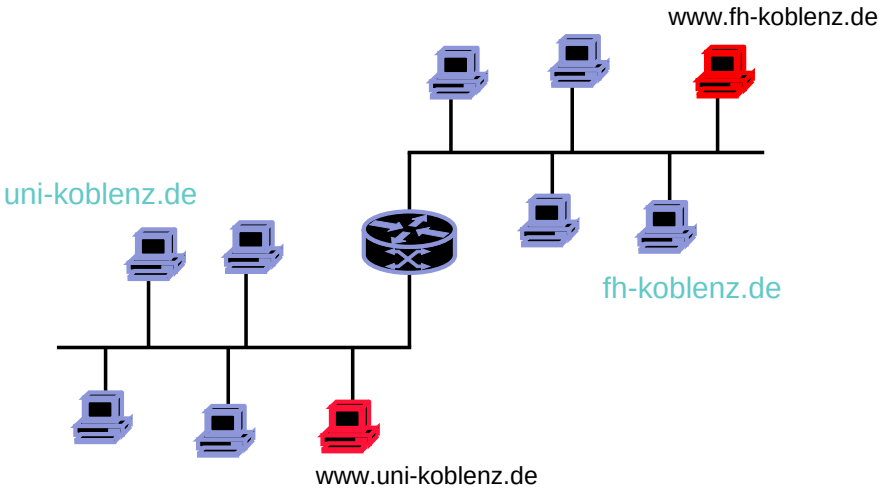

Abbildung 3.10: Gleiche Rechnernamen in unterschiedlichen Domänen

die gleichen Präfix-Namen erlaubt, da eine weitere Unterscheidung anhand der Domäne erfolgt. Die Abbildung 3.10 zeigt die symbolisch die Netze der Universität Koblenz und das der Fachhochschule Koblenz. Beide Hochschulen haben einen Webserver, mit dem Namen *www*, der aus dem Internet ansprechenbar sein soll. Die Server nutzen unterschiedliche IP-Adressen zu Adressierung. Verwendet man den Servernamen im Internet, so sind die Server durch das Namensuffix global unterscheidbar. Lokal ist die Verwendung des Namensuffix nicht unbedingt notwendig.

3.5 Protokolle

Das folgende Kapitel verdeutlicht die Funktionsweisen und Möglichkeiten dreier, für diese Arbeit relevanter Protokolle. Ein Protokoll bezeichnet eine Sammlung von Regeln über Format, Inhalt und Reihenfolge beim Austausch von Nachrichten. Das IP-Protokoll sowie das aufgesetzte TCP-Protokoll wird nicht näher behandelt, da es in allen Büchern über Netzwerke, zum Beispiel [PetDav2003], ausreichend dokumentiert ist. TCP/IP wird für die Implementation genutzt, stellt aber keine Informationen bereit, die für die Lösung der Topologieproblematik von Bedeutung sind. CDP und LLDP, vorgestellt in Abschnitt 3.5.2, sind Protokolle, die auf der Ebene 2 des OSI-Modells arbeiten und lediglich Informationen über Netzwerkgeräte gegenseitig austauschen. CDP und LLDP arbeiten autonom zwischen den Geräten ohne Verarbeitung der Informationen durch Algorithmen imnerhalb der Geräte. Sie arbeiten völlig autonom zwischen den Netzwerkgeräten, die dafür konfiguriert sind. SNMP übertrögt Daten, auf Anfarge, für das jeweilige Gerätan eine zentrakle Managementeinheit. Eine Interpretation innerhalb des Netzwerkgerätes selbst erfolgt nicht. SNMP ist ein Netzwerkmanagementprotokoll und dient zur Informationsvermittlung über ein Netzwerkgerät, welche von einem Manager explizit abge-

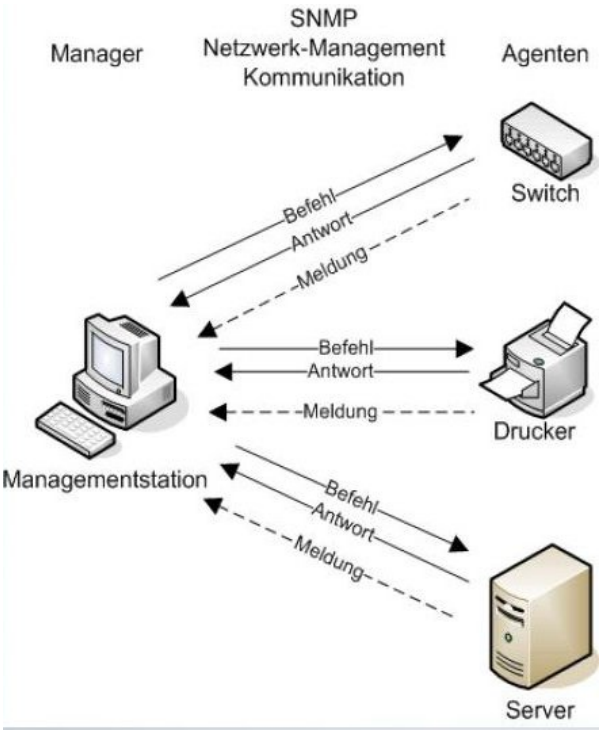

Abbildung 3.11: Kommunikationsmodell für SNMP. Grafik aus [Bohdan06]

fragt werden kann. SNMP stellt die Schnittstelle bereit, die zwischen Netzwerkgerät und Managementtool, Daten austauscht, zum Beispiel um die Informationen, die durch das CDP-Protokoll verteilt werden.

3.5.1 SNMP

Knapp erklärt ist Simple Network Management Protokoll (SNMP) nach [Harnisch02] ein Protokoll, welches zur Verwaltung und Konfiguration unterschiedlicher Netzwerkkomponenten dient. Router, Switches und Hosts können beliebig im Netz verstreut sein und von theoretisch jedem Arbeitsplatz mit einem Managementtool verwaltet werden. SNMP verwendet das UDP-Protokoll und gehört daher in eine Klasse von Protokollen, deren Priorität die Performanz ist. SNMP dient zur Sammlung von Informationen der einzelnen Netzwerkkomponenten aus dem Netzwerk selbst. Es ist nicht nur ein Protokoll, sodnern auch ein Zusammenspiel mehrerer Komponenten bestehend aus Agenten und zentralen Steuerelementen. In der Arbeit von [Bohdan06] wird SNMP in drei Kernkomponenten unterteilt.

- Überwachung von Netzwerk-Komponenten
- Fernsteuerung und Fernkonfiguration

- Fehlererkennung und Fehlerbenachrichtigung

Einzige Voraussetzung ist, dass das zu überwachende Gerät SNMP-fähig und konfiguriert ist. Bei Hardwarekomponenten wie Switches oder Routern ist SNMP installiert und lediglich zu aktivieren und mit Sicherheitsrichtlinien auszustatten. Für reguläre Arbeitsstationen ist eine detaillierte Konfiguration notwendig und unter Umständen eine SNMP-Unterstützung in Form von Software zu installieren. Bei falscher Anwendung kann es durch SNMP zeitweise zu sehr vielen Anfragen kommen. SNMP bietet die Möglichkeit, dass nicht nur eine Information ausgelesen werden kann, sondern es gibt MEthoden, die eine Reihe von Informationen in einem erlauben. Ein Problemfall wird später im Abschnitt 5.3 erläutert. Da diese Daten von SNMP zusätzlich zum regulären Verkehr über das Netzwerk übertragen werden müssen, liegt es nahe, dass sie das Netz nicht zusätzlich belasten sollten, in Form von Flußkontrolle und Sicherheit, wie es über das TCP Protokoll erfolgen würde. Daher ist es eine sinnvolle Entscheidung, SNMP auf UDP laufen zu lassen, das zwar nicht die Sicherheit von TCP bietet, aber dafür mehr Performanz.
Das SNMP-Modell wird in [Bohdan06] in vier Bausteine unterteilt.

Verwaltete Knoten - Agenten Jeder zu verwaltende Netzwerknoten ist mit einem autonomen Programm ausgestattet, dem Agenten. Dieser arbeitet auf dem Knoten und sammelt die notwendigen Daten, um diese in einer lokalen Datenbank zu abzulegen. Der Agent überwacht die Dienste des Knotens und horcht auf Aktionen, die für ihn definiert sind. Der Agent arbeitet autonom, wird aber von einem Manager konfiguriert. Es besteht keine permanente Verbindung zwischen Manager und Agent. Die Kommunikation erfolgt durch Anfrage des Managers oder über definierte Traps[13] seitens des Agenten.

Managementstation - Manager Der Manager ist ein Programm aus einer Vielzahl von möglichen Werkzeugen und ist von jedem Rechner im Netzwerk ausführbar. Der Manager ist die Schnittstelle für den Anwender, um Anfragen an einen Agenten senden zu können. Der Manager kann auf unterschiedlichen Abstraktionsebenen existieren. Der simple Kommandozeilenaufruf *SNMP-GET* kann als rudimentärer Manager betrachtet werden. Für den Anwender ist eine grafische optimierte Applikation die angenehmere Variante. Diese kommuniziert mittels des SNMP-Protokolls mit dem Agenten, verwertet die erhaltenen Daten und stellt diese aufbereitet dem Anwender dar.

ManagementInformationenBase (MIB) Diese ist der Datenspeicher für den Agenten. Der Aufbau entspricht einem Baum, in dessen Blätter die Informationen abgelegt sind. Die Knoten entsprechen den Schlüsseln. Die Definition einer MIB erfolgt durch die Beschreibungssprache SMI[14]. Die Darstellung erfolgt durch Objekt-Identifizierungen(OID) in einer Punktnotation. Eine textuelle Beschreibung der OIDs für genormte MIBs existiert in zugehörigen RFCs[15] oder sind beim Herstel-

[13] Selbstständige Übermittlung der Daten an eine Managementstation, ohne expliziete Anfrage.
[14] SMI ist eine auf ASN.1 basierenden Regelsammlung zur Beschreibung von Objekten.
[15] Request for Comments. RFCs beschreiben Standards für das Internet, welche keine offiziellen Standards sind, aber von vielen akzeotiert sind. Die dienen Diskussionen und Forschungskonzepten

ler zu beziehen. Diese sind für MIB-Browser (vgl. Abbildung 3.13) notwendig, um den unleserlichen Dot-Strings[16] eine verständlichere Form zu verleihen.

Netzwerkmanagement-Protokoll - SNMP ist das Protokoll zur Kommunikation zwischen Agent/Datenbank und Manager. Durch verschiedene Kommados wird ein Anfrage-Antwort Protokoll realisiert. Die verschiedenen Versionen von SNMP erlauben unterschiedliche Varianten die Daten zu empfangen. SNMPv1 erlaubt nur simple GET, SET und WALK Befehle. Hingegen existieren in SNMP Version 2c Ergänzungen wie BULK und TABLE, die mittels einer einzigen Anfrage eine Menge an Variablen auslesen können.

Der Manager, welcher seine SNMP-Anfragen über UDP/IP sendet, benötigt ein geeignetes Transportmedium. Idealerweise werden neben dem heute weit verbreiteten Ethernet zusätzlich noch die langsam auslaufenden Medien Token-Ring, FDDI oder ATM unterstützt, was eine zuverlässige Nutzung bei unterschiedlichen Übertragungsmedien ermöglicht. Als Gegenstelle zur Managementeinheit ist ein Agent auf jedem Netzwerkgerät notwendig und benötigt zwingend eine SNMP und UDP/IP Implementierung. Nur eine Protokollimplementierung reicht nicht aus, um das Gerät zufriedenstellend managen zu können. Jede Komponente benötigt den Agenten-Prozess, der zum Einen die SNMP Anfragen verarbeiten und zum Anderen die Daten aus dem Gerät sammeln kann [Zenk99]. Diese Notwendigkeit ist der Grund dafür, warum SNMP nur auf hochwertigeren Geräten zur Verfügung steht, die über ein Betriebssystem verfügen. In der Regel sind dies Geräte ab Layer3 des OSI-Models.
Ein zentrales Konzept, welches bei SNMP zum Tragen kommt, ist das der MIB. Die MIB ist nach Beschreibung von [Zenk99] eine Datenbank, welche die Grundlage für das SNMP-Management ist. In ihr sind alle Elemente definiert, die überwachbar sind. Für einen detaillierteren Einblick in Definition und Aufbau von MIBs sei auf [Klug99] verwiesen. Die MIB Variablen enthalten Angaben über Systemnamen, Verbindungen einzelner Protokolle oder Routingtabellen. Zusätzlich obliegt es jedem Hersteller eigene MIBs zu definieren, die herstellerspezifische Angaben enthalten, oder auch Angaben, die der Hersteller für den Anwender zur Verfügung stellen möchte. Jedes Netzwerkgerät unterstützt die Standard-MIBs (MIB I und MIB II). Die Standard-MIB enthält lediglich herstellerunabhängige Definitionen, welche von jedem Managementsystem auslesbar sind. Eine brauchbare Hilfe sind MIB-Browser. Sie haben den Vorteil, dass der Anwender graphisch geführt die Variablen und ihre Inhalte betrachten kann (Abbildung 3.13). Unter UNIX/Linux steht das NET-SNMP[17] Paket zur Verfügung, mit dem auch auf der Kommandozeile Informationen abgefragt werden können. Mittels des *snmpwalk* - Kommandos ist es möglich einen kompletten Teilbaum der MIB auszulesen. Setzt man den Befehl an der Wurzel an, erhält man alle Informationen, die in der MIB verfügbar sind. Eine gezielte Informationsbeschaffung erfolgt durch einen *Get-Request*. Für diese Aktion muss der Benutzer wissen welche Variable er auslesen möchte. Das bedeutet er muss den namentlichen Bezeichner oder den String in Dot-Notation kennen. Bei her-

[16] Aneinanderreihung von Ziffern, die durcj Punkte separiert sind.
[17] http://net-snmp.sourceforge.net/

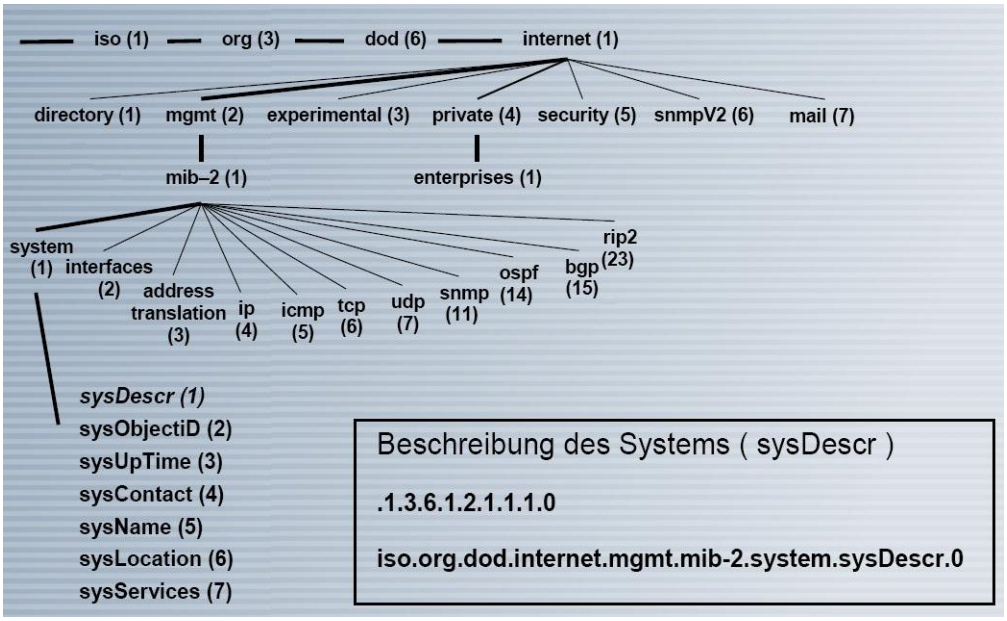

Abbildung 3.12: MIB-Baum für die Standardvariablen *SYSTEM*. Grafik aus [Bohdan06]

stellerspezifischen Angaben wird das schwer, ausgenommen der Hersteller veröffentlicht eine MIB-Definition. Diese kann zum Beispiel in den MIB-Browser eingelesen werden und so hat der Anwender die gleichen Möglichkeiten in der Navigation innerhalb der MIB wie unter den Standardangaben. Dieser Punkt macht es für Managementsysteme schwer herstellerunabhängig zu bleiben. Herstellerunabhängige Systeme sind darauf angewiesen, dass die MIB anderer Hersteller verfügbar sind. Sei es frei oder kommerziell. Bei der Auswahl der Managementsoftware ist darauf zu achten welche Netzkomponenten derzeit und in naher Zukunft zu verwalten sind und ob Software und Hardware zusammenpassen.
Das Protokoll *SNMP* gibt es derzeit in drei Versionen. Version SNMPv2c ist die derzeit am besten unterstütze, welche den vollen Funktionsumfang von SNMPv1 enthält, mit einigen Ergänzungen bezüglich der Datenabfrage, wie eine komplette Tabelle, für die bislang jedes Element einzeln abgefragt werden musste. IN der Frage um die Sicherheit konnte keine Einigung bei den Entwicklern gefunden werden, wodurch die Sicherheitsfunktionen bei denen von SNMPv1 geblieben sind. Der einzige Sicherheitseffekt wird durch Vergabe von zwei Passwörter, eines zum Lesen und eines zum Lesen und Schreiben, gegeben. Die Passwörter werden zudem im Klartext übertragen und sind daher nach heutigen Sicherheitsaspekten, nicht empfehlenswert. Version 3 nutzt verschlüsselte Passwörter; jedoch wird diese Version von vielen Geräten noch nicht unterstützt der Administrator wird im Regelfall mit der Version zwei konfrontiert. Solange das Management mit einem rein lesenden Zugriff zurecht kommt, ist die Anwendung noch vertretbar. Switches und Router haben selten Informationen, die sensible Daten beinhalten, die es

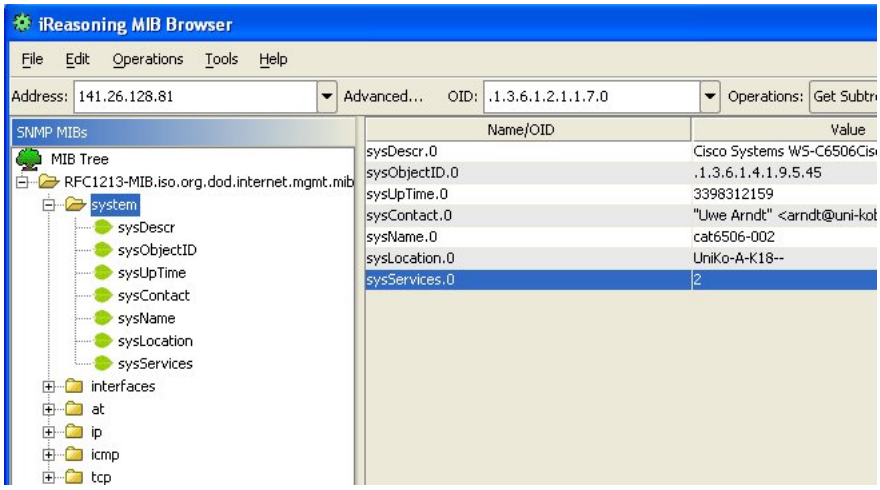

Abbildung 3.13: MIB-Browser

geheim zu halten gilt. Anders sieht es aus, wenn durch Schreibrechte die Konfigurationen geändert werden können, um zum Beispiel Zugang zu andere VLANs zu bekommen. Ab diesem Punkt hat das Netzwerk ein Sicherheitsproblem. In diesem Fall empfiehlt es sich, das Management nur in einem VLAN zu ermöglichen oder den Zugriff auf bestimmte Netzwerkadressen zu beschränken. Das setzt voraus, das Netzwerkkomponenten und administrative Netzwerkanschlüsse physisch nicht zugänglich sind.

3.5.2 Discover Protokolle

Switches ab Stufe Layer2 sind in der Lage, Informationen aus dem Header der Datenpakete zu entnehmen, die sie weiterleiten. Die Sender- und Empfänger-MAC-Adresse wird ausgelesen und die Daten für den jeweiligen Port in einer Tabelle gespeichert. Der Switch ist so in der Lage, Datenpakete gezielt weiterzuleiten. Bei einfachen Switches hat der Benutzer keine Möglichkeit die Daten des Gerätes auszulesen oder zu betrachten. Vereinzelt existieren Geräte, die Trunking erlauben, welches durch eine Konsolenschnittstelle[18] ermöglicht wird. Es ist jedoch kein höheres Betriebssystem integriert, welches sich über IP ansprechen lassen würde. Diese Fähigkeit existiert bei Geräten ab Layer3, die eine Schnittstelle für eine Konsole anbieten, über die der Anwender das Gerät detailliert administrieren kann. Die Bezeichnung Layer3 wird durch das IP-fähige Betriebsystem erlangt, welches eine Kommunikation über das Netzwerk unter Verwendung des IP Protokolls ermöglicht. Der Switch sammelt nicht nur die Angaben über die MAC-Adressen, sondern stellt zusätzlich Angaben über Auslastung und Fehler bereit. Desweiteren ist diese Art Switch in der Lage Multicastpakete zu verarbeiten, in denen spezielle Infor-

[18] Manche Hardware hat keine Anschlüsse für Ein- und Ausgabegeräte. Für Konfigurationen wird eine Schnittstelle angeboten, über die man einen regulären Rechner anschließen kann und die Einstellungen über den Computer vornimmt.

mationen enthalten sind. Zum Beispiel wird es von CDP und LLDP genutzt, um ihre Informationen über Nachbargeräte bekannt zu machen. Neben TCP ist auch das SNMP-Protokoll implementiert. SNMP wird von den bekannten Herstellern unterstützt, sodass es mittels SNMP-Clients sehr einfach ist detaillierte Informationen aus den Geräten auszulesen.
Durch die Kombination der Informationen aus den Switches, welcher Port welche MAC-Adressen kennt, existiert eine Möglichkeit zur Darstellung der Gerätebeziehungen. Die Information der MAC-Adresse wird ab dem Punkt verwertbar, ab dem eine Adresse auf eine IP oder einen Rechnernamen abbildbar ist. In den seltensten Fällen sind die MAC-Adressen der Geräte dem Administrator direkt geläufig. Es ist wichtig, dass die Informationen so aufbereitet sind, dass der Verwalter eine direkte Assoziation zu der Netzwerkkomponente hat, wie etwa Name oder IP eines Gerätes. Zu den Standardangaben der Switches bieten viele Hersteller zusätzliche Informationen an. Für diesen Fall wird die herstellerspezifische MIB benötigt. Eine der herstellerabhängigen Informationen ist die des *Cisco Discovery Protokolls*, welches im Anschluss erklärt wird. Ein Protokoll, dass durch seinen Nutzen, einen so hohen Wert erlangt hat, dass es von vielen anderen Herstellern mit in die Software integriert worden ist. Vor einiger Zeit erfolgte die Verabschiedung des LLDP-Standards, einem Protokoll, welches die gleiche Arbeit verrichtet, wie es das CDP-Protokoll tut. Beide Protokolle erlauben es, auf sehr einfache Art und Weise, direkten Nachbarn zu ermitteln. Seit der Einführung von LLDP wird die Unterstützung für CDP von diversen Herstellern abgelehnt. Neuere Produkte setzen auf den offiziellen Standard LLDP, der die gleiche Arbeit verrichtet wie CDP. Die beiden Protokolle sind nicht zueinander kompatibel. Seitens der Nutzer wäre es wünschenswert, dass Netzwerkgeräte beide Protokolle beherrschen und aus beiden Informationsquellen eine Tabelle aufbauen. Bislang werden, wenn Geräte beide Protokolle beherrschen, zwei getrennte Datenspeicher geführt.

CDP

Das Protokoll, von Cisco entwickelt, dient zum Informationsaustausch zwischen herstellereigenen Geräten, um Nachbarschaftsinformationen zu verteilen. Das Protokoll arbeitet auf der Sicherungsschicht[19] und wird von Bridges, Switches und Router verwendet. Auch reguläre Computer können mit dem Protokoll ausgestattet werden. Die wesentlichen Informationen, die durch das Protokoll verschickt werden, sind der Name, die IP und der angeschlossene Port des sendenden Rechners. Das Datenpaket wird durch alle Ports des Gerätes verschickt. Eine Multicastadresse gewährleistet, dass Layer3-Switches diese Datenpakete nicht weiterleiten, sondern nach der Verarbeitung verwerfen. Die Informationen sind nur für den direkten Nachbarn bestimmt und nicht für weitere Geräte im Netzwerk. Hubs arbeiten auf der Bitübertragungsschicht[20] und leiten die Multicastpakete an alle Ports weiter, da sie für eine Unterscheidung von Paketen nicht konzipiert sind. Switches auf Layer-2 Ebene sind zwar in der Lage, Pakete zu selektieren und an ausgewählte Ports zu senden, sind aber nicht für Multicastpakete sensibel. Layer2-Switches

[19] Layer 2-Ebene des OSI-Modells
[20] OSI-Ebene 1, die rein physikalische Verbindung

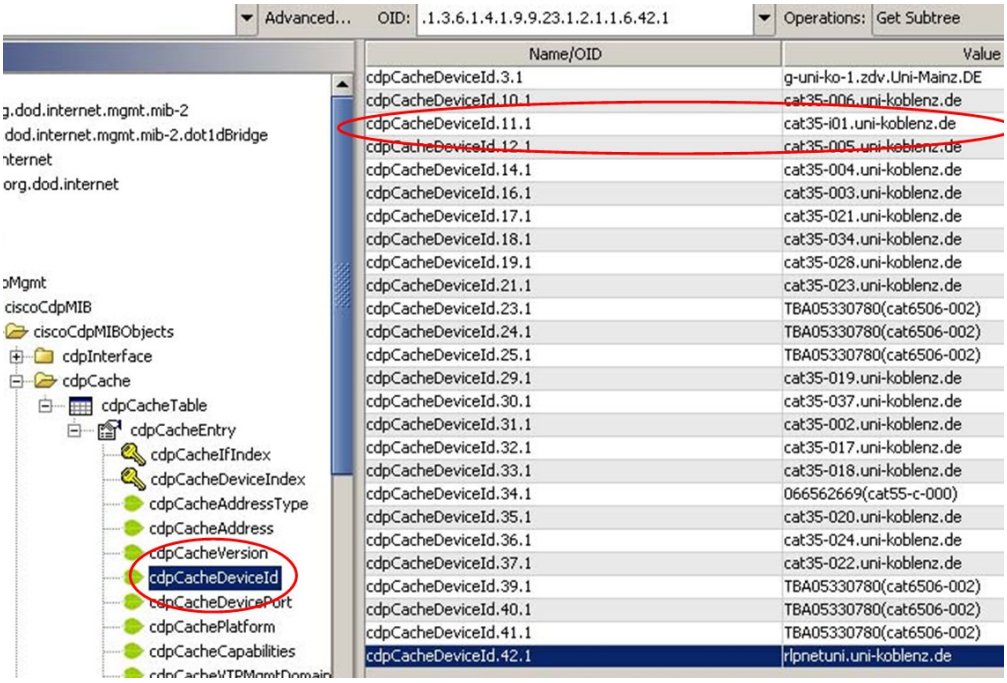

Abbildung 3.14: Informationen des CDP-Cache in einem MIBBrowser

bieten keine Verarbeitungsmethoden, die mit den Informationen etwas anfangen könnten, desweiteren könnten sie keine entsprechende CDP-Daten senden. Die CDP-Daten dienen dem Administrator als Unterstützung. Eine Abfrage der Informationen wäre aber nur an der Konsole des Switches möglich und nicht über ein Netzwerk mit zum Beispiel *Telnet*. Dies dürfte der Grund sein, warum bei günstigeren Switches darauf auf dieses Protokoll verzichtet wird.

Wichtig ist, dass das CDP von den darunter liegenden Protokollen unabhängig ist, wodurch es ermöglicht wird, Geräte zu ermitteln, die nicht nur über Ethernet, sondern auch über Token-Ring oder ATM angeschlossen sind. Die Komponenten lernen so Ihren Nachbarn kennen und geben im Gegenzug ihre eigenen Informationen weiter. Würde der gegenseitige Informationsaustausch ausfallen, wird der Nachbar aus der Tabelle entfernt. Die in den Switches gespeicherten Informationen, welcher Switch welchen Switch als Nachbarn hat, ist vom Netzwerkverwalter zu extrahieren und mit Hilfe geeigneter Maßnahmen als vollständige Topologie darzustellen. Sinnvoller Weise wird dazu SNMP verwendet, damit die Sammlung der Informationen durch eine Software realisierbar ist. Hervorzuheben ist, dass CDP nicht die Topologie erzeugt, sondern ein Protokoll ist, dass gerätespezifische Daten an seines Gerätes übermittelt und diese die Informationen in einer Tabelle bereitgestellen. Das Protokoll hat sich als sehr brauchbar erwiesen, so dass Konkurrenzhersteller von Cisco das Protokoll in ihre Systeme mit übernommen haben, um ihren Kunden die gleichen Möglichkeiten zu bieten. Große Netze, die ein Management benötigen, werden auf dieses Feature nicht verzichten wollen.

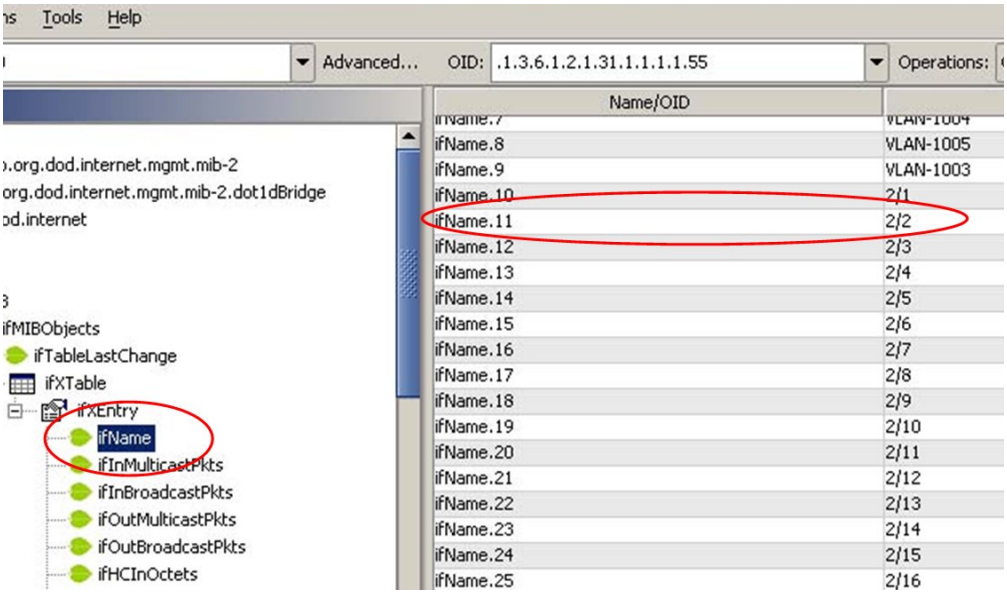

Abbildung 3.15: Informationen der MIB *Interface* in einem MIBBrowser

LLDP

Das Link Layer Discovery Protokoll ist nach [Felser06] ein Protokoll auf Ebene 2 des OSI-Modells und dient zur Mitteilung von Identität und Eigenschaften eines Gerätes. Die Informationen dienen nicht der Funktionalität des Gerätes, sondern sind Informationen für den Verwalter des Netzwerkes. Mit diesen Informationen ist es sehr einfach die Topologie des Netzwerkes, sowie der angebundenen Geräte darzustellen, wenn das jeweilige Gerät LLDP unterstützt. LLDP, welches unter dem Standard IEEE 802.1AB im Mai 2005 verabschiedet worden ist, ist eine Weiterentwicklung des CDP-Protokolls von CISCO, mit dem Vorteil, dass es herstellerunabhängig ist. Die Daten werden in einer MIB gespeichert und sind über SNMP zugänglich. Die MIB speichert sowohl die Informationen des eigenen Agenten, sowie Informationen über seine direkten Nachbarn. Wichtig ist das Vorhandensein einer eindeutigen Geräte-ID, die im Netzwerk einmalig sein muss.
LLDP erlaubt es, ein Gerät im Netzwerk bekannt zu machen und Daten von anderen zu sammeln. Es stellt die Daten für Netzwerkmanagementsysteme bereit. Die eigentliche Intelligenz hinsichtlich der Datenverarbeitung obliegt der Managementsoftware, nicht dem Agent, der auf der zu überwachenden Hardware installiert ist. Je nach Implementierung oder Interpretation kann es bei unterschiedlichen Managementsystemen zu unterschiedlichen Ergebnissen kommen. Konzipiert ist das Protokoll sehr einfach. In periodischen Intervallen werden die LLDP-Pakete über alle Ports versendet. Eine Antwort oder Empfangsbestätigung gibt es nicht. Dem Sender ist es egal, ob ein anderes Gerät seine Informationen verarbeitet oder nicht. Die Datenpakete werden an die Multicastadresse *01:80:c2:00:00:0e* geschickt.

Ebenso wie beim CDP-Protokoll werden die Datenpakete von Hubs und Layer2-Switches nicht erkannt und als Broadcast weitergereicht. Wäre dies der Fall, würden Switches nach dessen Knoten ebenfalls die Nachbarschaftsinformation bekommen, obwohl es kein Nachbar ist, da noch eine Station dazwischen liegt. Das bedeutet, dass ein Datenpaket von einem Port lediglich einen Port eines Netzwerkgerätes erreichen sollte. Wird ein Layer1/Layer2-Switch zwischen drei oder mehr Layer3-Geräte gesteckt, führt das zu der Situation, dass an einem Port des Layer3-Switches mehrere LLDP-Nachbarn registriert sind. Eine weiteres Feature ist, dass die LLDP Pakete auch über Ports verschickt werden, die über Softwaresteuerung abgeschaltet sind, aber physisch eine Verbindung haben. Ein Fall wäre die Abschaltung durch den Spanning-Tree Algorithmus (siehe Abschnitt 3.6), da die Verbindung eine Schleife produzieren würde. Die Information, dass das Gerät einen Nachbarn hat, macht Sinn, um den Verwalter die Information über eine Alternativleitung zu geben, die aber im aktiven Netzwerkverkehr nicht genutzt wird. Bei der Topologieerkennung, die nur auf Spanning-Tree basiert, würde diese Verbindung nicht sichtbar werden (vgl. Abschnitt 6.2 ab Seite 96). Eine brauchbare Erweiterung bietet LLDP in der Definition von organisatorisch oder herstellerspezifischen Informationsblöcken, wodurch benutzerorientierte Erweiterungen möglich sind. Die minimale Angabe beinhaltet:

- Chassis ID, oftmals die MAC-Adresse
- Port ID
- Time to Live, wie lange das Paket seine Gültigkeit haben soll

Generell werden die Informationen der Nachbarn nur eine gewisse Zeit gespeichert. Werden die Infos nicht regelmäßig erneuert, wird der Eintrag des Nachbarn gelöscht.
LLDP bietet eine sehr übersichtliche MIB, auf die mittels SNMP sehr einfach zugegriffen werden kann. Die Informationen sind durch Übergabe der eigenen IP, Port und Name, an die angeschlossenen Nachbarn, und der Verarbeitung der Informationen, die empfangen worden sind, sehr gut zu verarbeiten. Die Angaben beschränken sich nur auf direkte Nachbarn. Erst die Übertragung aller Informationen mit Hilfe von SNMP an eine zentrale Verarbeitungseinheit ermöglicht den Aufbau der gesamten Topologie. Da die Angaben unter den Geräten nur verteilt werden und keine Berechnungen hinsichtlich der Gesamttopologie angestellt werden, ist das Verfahren sehr schnell und unauffällig im Netzwerk.

3.6 Spanning Tree

Ist die Topologie kein einfacher Stern , sondern zum Beispiel ein Baum, mit einer tiefen Verschachtelung, sind die Switches untereinander über Uplinkports verbunden. Es ist zu erwarten, dass der Uplinkport deutlich stärker belastet ist, als die Ports, an die Computer angeschlossen sind. Aus diesem Grund werden bei hochwertigeren Geräten die Uplinkports leistungsstärker dimensioniert, da die Mehrheit der Datenempfänger in großen Netzen nicht am eigenen Switch zu finden sein werden, sondern weit im Netzwerk

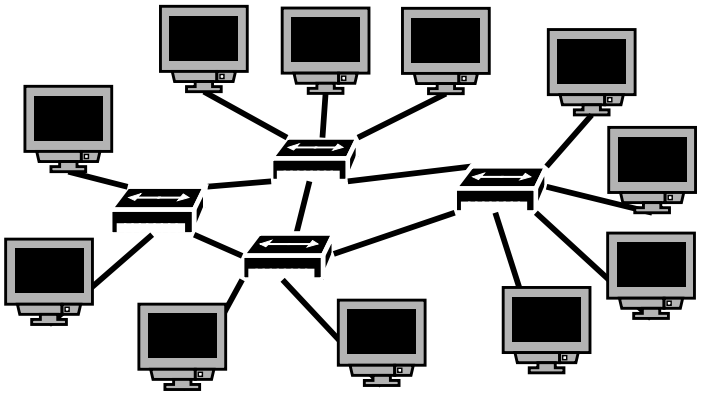

Abbildung 3.16: Vermaschtes Netzwerk

verstreut sind. Um die Leitungskapazität eines Switches zu erhöhen liegt es Nahe, mehrere Leitungen für den Uplink zu verwenden. Zu diesem Zweck erlauben einige Switches das Trunken. Bei dieser Idee definiert man bei den beiden Switches, die einen höheren Datendurchsatz benötigen, jeweils die gleiche Anzahl physischer Ports logisch als einen. Zwischen diesen beiden Geräten werden Datenpakete ab diesem Moment gleichmäßig auf die definierten Leitungen verteilt. Eine weitere Variante, das Netzwerk effektiver zu machen, ist Switches nicht nur mit einen anderen Switch zu koppeln, sondern einen Switch an mehrere verschiedene Switches anzuschließen. In der der Erwartung, dass sich die Datenpakete gleichmäßig auf die verfügbaren Leitungen verteilen. Durch den Anschluss eines Switches an mehrere Eltern-Switches[21] besteht die theoretische Möglichkeit, dass sich Datenpaketen intelligent verteilen, um weniger Switches passieren zu müssen. Desweiteren würde die mehrfache Verkabelung eine Sicherheit für den Switch bedeuten, wenn eine Verbindung ausfallen würde. Hat ein Switch nur einen Uplink zu einem anderen Elternknoten und dieser fällt aus, ist der Kind-Switch, und alle abhängigen Geräte, vom restlichen Netzwerk abgeschnitten und nicht mehr erreichbar. In diesem Fall besteht nur noch eine Kommunikation unter den Geräten in den beiden entstandenen Teilnetzen. Bei der Verkabelung mit mehreren Eltern-Switches besteht für die Datenpakete eine alternative Route.
Abbbildung 3.16 zeigt ein vermaschtes Netzwerk, bei dem jeder Switch eine Verbindung an mehrere andere Switches hat. Die Datenpakete werden, wie in Abschnitt 3.2 beschrieben, an den angeschlossenen Hubs oder Layer2-Switches dupliziert. Auf diese Weise entstehen Schleifen im Netzwerk, was zu einem Totalausfall wegen Überlast führt. Das folgende Beispel erklärt die Abbildung 3.16.

[21] Switch, der in der Hierarchie der Baumstruktur weiter oben steht.

Problemszenario für ein Netzwerk mit Schleifen: Wird ein Netzwerk aus Hubs oder Layer2-Switches aufgebaut und es wird eine vermaschte Verkabelung, wie in Abbildung 3.16 verwendet, führt dies zu Problemen. Bei Verwendung von Hubs wird ein einmal gesendetes Paket einen Hub erreichen und dort an jeden verfügbaren Port, bis auf den, von dem das Paket kommt, weitergereicht. Ist an dem Port ein Rechner angeschlossen, wird dieser das Paket verwerfen, wenn es nicht für ihn bestimmt ist. Ist ein weiterer Hub angeschlossen, so wird dort wiederum das Paket vervielfältigt und an jedem Port weitergeleitet. Bei der vermaschten Anordnung wird das Paket, welches von Hub eins an Hub zwei gesendet wurde als Kopie wieder Hub eins erreichen und der Prozess beginnt von vorn. Wie in Abschnitt 3.2 zu lesen, wird durch das CSMA/CD-Verfahren keine Rechner mehr in Lage sein, neue Pakete in das Netzwerk zu senden.
Versucht man das Netz, durch Einsatz von Layer2-Switches, effizienter zu machen, erwartet man, dass keine Probleme in dieser Form auftreten. Rechner senden des öfteren Broadcastpakete durch das Netz. Zum Beispiel, wenn ein Rechner neu in ein Netzwerk integriert wird. Der erste Switch lernt die MAC-Adresse des Senders und leitet die ersten Datenpakete, wie bei einem Hub an alle Ports weiter. Auf diese Weise lernen Switches die gleiche MAC-Adresse des Senders über unterschiedliche Ports. Der Empfänger wird das Paket empfangen und eine Antwort ins Netzwerk senden. Der angeschlossene Switch nimmt das Paket entgegen und wird es im Beispiel an beiden Ports weiterreichen, die Tabelle für zwei Ports den Eintrag der Ziel-MAC-Adresse registriert haben. Ebenso wird sich jeder folgende Switch verhalten. Es wird eine Unzahl an gleichen Paketen entstehen. Diese Situation gilt es zu verhindern.

Eine Lösung, Schleifen in Netzen zu erhalten, aber die Probleme zu umgehen, stellt der Spanning-Tree-Algorithmus (STA) dar. Nach [InTeHa01] definiert der STA eine schleifenfreie Untermenge der Netzwerktopologie. Die Idee ist, dass die Switches selbstständig entscheiden, ob es alternative Routen gibt. Ist dies der Fall, wählt der Switch den Pfad mit den niedrigsten Pfad-Kosten und schaltet die übrigen Uplinks über die Switchsoftware ab. Diesen Zustand nennet sich *Blocking*. Auf diesen Ports werden keine Daten weitergeleitet, stehen aber bei einem Ausfall des bislang aktiven Ports als Reserve zur Verfügung und werden durch den Switch selbst aktiviert, während der bisherige Port in den Zustand *Down* wechselt. Die Abbildung 3.17 zeigt das vermaschte Netz nach der Modifikation durch den Spanning-Tree-Algorithmus. Die rot gepunkteten Linien stellen physische Links dar, die durch den STA des Switch abgeschaltet sind und logisch nicht verfügbar ist. Das Vorgehen des Spanning-Tree-Algorithmus wird in [InTeHa01] wie folgt beschrieben:
Jeder Switch wird mit einer Id versehen. In den häufigsten Fällen ist dies die MAC-Adresse des Switches. Jeder Port besitzt eine Kostenstelle in Abhängigkeit seiner Leistungsfähigkeit. Dadurch soll gewährleistet werden, dass möglichst der Port als Uplink dient, der den besten Datendurchsatz bis zur Wurzel des Netzwerks verspricht. Die Summierung von mehreren mittelmäßigen Strecken kann unter Umständen günstiger sein als die Summierung einer extrem schnellen und mehreren extrem langsamen Strecken.

[21] Pakete, die alle Stationen des Netzwerks ereichen sollen.

An dieser Stelle ist darauf hinzuweisen, dass es für die Pfadkosten offensichtlich keine Norm verwendet wird. Bei der Betrachtung der Kosten unterschiedlicher Hersteller, im Netzwerk des Campus Koblenz, sind starke Unterschiede verschiedenen Herstellern hinsichtlich der Pfadkosten für die gleiche Geschwindigkeit eines Ports festgestellt worden. Die Pfadkosten sind durch den Administrator anpassbar, um diesen Fehler auszugleichen. Ebenso besteht die Möglichkeit die Priorität eines speziellen Gerätes zu manipulieren

Der erste Schritt besteht darin, den Switch mit der niedrigsten Id zu finden. Im Anschluss ermitteln alle übrigen Switches den Root-Anschluss, der Port mit den geringsten Gesamtpfadkosten zum Root-Switch. Im Anschluss wird die designierde Bridge[22] ermittelt. Der Switch, dessen Leitung die geringsten Pfadkosten verspricht wird erhält den Zuschlag. Sind zwei oder mehr Bridges gleichermaßen günstig, entscheidet die Id des Gerätes. Die Berechnungen erfolgen unmittelbar nach dem Einschalten eines Switches. Damit STA funktioniert, ist eine Kommunikation unter den Geräten in Form von Datenpaketen, den BPDUs[23], notwendig. Sie enthalten Angaben über den Switch, welche von den Nachbarn verarbeitet werden. Die Daten werden regelmäßig gesendet und erlauben so eine Kontrolle. Bleiben die BPDUs bei einem Switch aus, gilt der Nachbar als nicht erreichbar. Es wird eine neue Berechnung des Spannbaums eingeleitet. Die Umorganisation benötigt einige Sekunden Arbeitszeit. In der Zeit ist das Netz unbrauchbar. Das ist in der Form nicht akzeptabel. Die Umgehung dieses Problems führte zur Entwicklung des Rapid-Spanning-Trees. Dieser organisiert den Baum, ohne Beeinflussung der aktuelen Pfade, neu und aktiviert diesen erst dann, wenn der neue Spanbaum in einem schleifenfreien Zustand ist. Nach erfolgreicher Überprpüfung der neuen Pfade, werden die Ports der Switches umgeschaltet und die Reorganisierung dauert augenscheinlich weniger als eine Sekunde.
Der Spanning Tree versucht immer die Strecke zu finden, auf der die schnellste Datenübertragung möglich ist. Für die Teilnahme am Spanning Tree ist ein Betriebssystem im Switch notwendig, welches die Berechnungen anstellt. Einfache Layer2-Switches, die zwar in der Lage sind, MAC-Adressen ihrer Ports zuzuweisen, haben in der Regel kein geeignetes Betriebssystem, welches diese Berechnung ermöglicht. Sie nehmen nicht am Spanning Tree teil. Eine Mischung von Layer1/Layer2-Geräten zerstören den Nutzen der Layer3-Switches, durch die man sich einen Vorteil erhofft hat. Die Intention, alternative Routen bereitzustellen, geht bei dieser Art Switches am Ziel vorbei.
Nicht zu verwechseln mit der Erzeugung von Schleifen, ist der Einsatz von Porttrunking. Ist der Switch für Trunking ausgelegt, regelt die Software die Verteilung der Pakete auf die zusammengeführten Leitungen. Die Leitungen verteilen sich aber nicht auf unterschiedliche Geräte, sondern jedes Kabel mündet in den gleichen Switch und fasst mehrere Leitungen zu einer zusammen, um die Gesamtbandbreite zu erhöhen.
Ist der Administrator in der Situation hochwertige Switches sein Eigen zu nennen, die den Spanning Tree beherrschen, kann er ohne Bedenken eine vermaschte Verkabelung durchführen. Unter der Beachtung, dass kein Layer1/Layer2-Geräte zwischen zwei Layer3-Switches eingebaut ist. Der Einsatz von Hubs oder Layer2-Switches, als An-

[22] Die Bridge, die als direkter Nachbar ermittelt wird.
[23] Bridge Protokoll Data Units

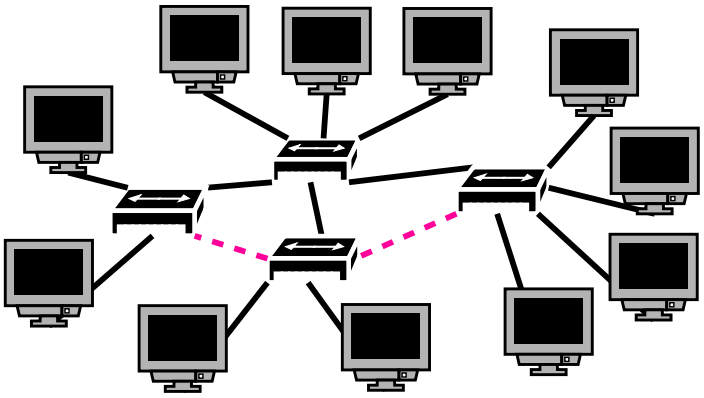

Abbildung 3.17: Vermaschtes Netzwerk mit deaktivierten Leitungen durch den Spanning Tree

schlußpunkt einer kleinen Gruppe von Rechnern, stellt keine größeren Probleme dar. Für weitere, tiefergehende Informationen bzgl. des Spanning Trees ist [Harnisch02], [HeinKoehler02] und [InTeHa01] empfehlenswert.

4 Netzwerk des Campus Koblenz

Jedes Netzwerk ist einzigartig. Beeinflusst von den gegebenen Räumlichkeiten und deren Ausdehnung, ist eine Entscheidung bezüglich der Art der Verkabelung und der zu verwendenden Netzwerkkomponenten zu treffen. Einfache Layer2-Switches, wie sie im Hausgebrauch üblich sind, erledigen ihre Aufgaben zufriedenstellend, indem sie ein nutzbares Netzwerk bereitstellen. Diese Geräte sind deutlich günstiger als ihre managebare Konkurrenz bieten aber keine oder nur geringe Möglichkeiten des Managements. Managebare Switches, die in der Regel als Layer3-Geräte bekannt sind, bieten eine Vielfalt an Optionen, die eine Netzoptimierung ermöglichen. Für kleine Netzwerke ist dieser Kosten- und der erhöhte Konfigurationaufwand, nicht notwendig. In großen Netzen bieten hochwertigere Switches eine Vielzahl an Optimierungen wie QoS[1], VLANs[2] oder sie stellen Messdaten und Statistiken zur Verfügung. Dadurch ist es möglich die Planung des Netzwerks auf die Bedürfnisse der Benutzer anzupassen, wodurch die Möglichkeiten einer effektiven Nutzung, im vollen Umfang, ausgenutzt werden können. Der Aufbau der Topologie, mit Hilfe der Performanzdaten der Switches, soll eine effiziente Nutzung des Netzwerks ermöglichen und für den Administrator aussagekräftige Informationen über das Netzwerks bereitstellen. Nur so ist eine benutzerorientierte Wartung eines Netzwerks möglich und erlaubt im Fehlerfall eine schnelle Problemlokalisation und effektive Fehlerbeseitigung.
Allein die physische Vernetzung erlaubt noch keine verwertbare Informationsgewinnung aus einem Netzwerk. Erst der Einsatz geeigneter Software unter Verwendung betriebssystemgestützter Netzwerkknoten, Switches und Router, und vor allem einer wohl durchdachten logischen Umsetzung, erlaubt eine brauchbare Interpretation der gewonnenen Informationen. Bei der Konstruktion eines Netzwerks berücksichtigen die Administratoren den erwarteten Netzwerkverkehr und dimensionieren anhand dieser Schätzungen die zu verwendende Hardware. Statistische Daten aus den Geräten, die zentral gesammelt werden, unterstützen sinnvolle Veränderungen im laufenden Betrieb. Vorausschauende Baumaßnahmen, wie der Einsatz von Patchpanels für ganze Bürobereiche, erlauben eine Skalierung des Netzwerks im laufenden Betrieb, ohne merkliche Baumaßnamen oder Störungen des bestehenden Netzwerks.
Probleme, in großen Netzwerken, ergeben sich durch Erweiterungen selbst ernannter Administratoren, die nicht mit der Betreuung des Netzwerks beauftragt sind, jedoch durch den selbstständigen Einsatz von billigen Netzwerkkomponenten die Topologie beeinflussen. Ethernet-Netzwerke lassen sich problemlos, und im ersten Moment unmerklich, durch Anschluss eines einfachen Hubs an einen freien Netzwerkport erweitern. Anwender, ohne entsprechende Kenntnis der Topologie und des technischen Hintergrundwissens,

[1] Quality of Service beschreibt die Möglichkeit Leitungen oder Daten den Vorrang zu gewähren
[2] Virtual Local Area Networks

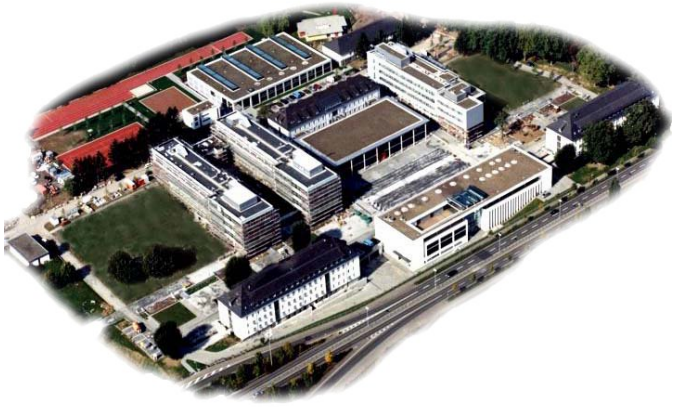

Abbildung 4.1: Luftaufnahme der Universität Koblenz-Landau, Campus Koblenz

können durch selbstständige Modifikation negativen Einfluss auf das Netzwerk nehmen, indem zum Beispiel Schleifen (vgl. Abschnitt 3.6) erzeugt werden. Diesen Situationen der Veränderungen, vor allem in großen Umgebungen, wie Firmen und Universitäten, teilweise ausgestattet mit autonomen Bereichen, gilt es vorzubeugen. Die Verwaltung eines Netzwerks muss in der Lage sein, für jede Situation zu einem beliebigen Zeitpunkt, den Status des Netzwerks zu erklären. Veränderungen sind sichbar zu machen. Die Netzwerkverwaltung bietet die Schnittstelle der Anwender ins Netzwerk, sei es die einzelne Person oder ein autonomer Betrieb, und ist bei einem Problem die erste Instanz, bei der die Problemursache gesucht wird. Daher ist es notwendig nicht nur die Komponenten des Netzwerks zu kennen, sondern es sollte möglich sein, Zusammenhänge sichtbar zu machen. Switches müssen, in Bezug auf Gebäude und deren Etagen, lokalisierbar sein, was eine Beurteilung vereinfacht. Rechnerarbeitsplätze und Server gilt es zu identifizieren und die Position, innerhalb der Gebäude und Flure, ist festzustellen. Durch die Kenntnis der räumlichen Situation sind Berichte in Problemfällen leichter zu erstellen und begründbar. Die folgenden beiden Abschnitte beschreiben zum Einen die Topologie des Campus Koblenz und zum Anderen Ideen zur Konfiguration und Entwicklung einer logischen Abbildung der physischen Umgebung zur geeigneten Verarbeitung mit einer Software zur visuellen Darstellung.

4.1 Netzwerkaufbau Campus Koblenz

Der Campus Koblenz der Universität Koblenz-Landau ist seit April 2002 in Koblenz Metternich auf dem ehemaligen Gelände der Pionierkaserne (Abbildung 4.1). Ein Teil der alten Gebäude sind übernommen worden und ein größerer Teil durch neue Gebäude ersetzt. Im Zuge dieser Neugestaltung ist von Anfang an auf die digitale Infrastruktur Wert gelegt worden. In der ersten Umzugsphase ist lediglich der Fachbereich 4 in die modernisierten Kasernengebäude eingezogen. Dadurch bedingt, sind die heutigen

Gebäude A und *Gebäude B*, ehemalige Kasernengebäude, komplett vom Fachbereich der Informatik sowie dem Rechenzentrum belegt. Das Rechenzentrum befindet sich im Erdgeschoß des A-Gebäudes, wodurch dieses Gebäude die Basis der Infrastruktur bildet. Diese beiden Gebäude, sowie die Gebäude C und K, die ebenfalls in der ersten Bauphase erschlossen worden sind, sind mit Lichtwellenleiterkabeln verbunden. Jede Etage eines jeden Gebäudes, ist mit mindestens einem Switch ausgestattet, welche direkt am Backbone[3], im A-Gebäude, angeschlossen sind. Das entspricht einer balancierten Baumtopologie (vgl. Abschnitt 3.3). Durch die, zum Einbauzeitpunkt fortschrittliche Technik der LWL-Verkabelung[4], ist zum gegenwärtigen Zeitpunkt immernoch eine zukunftssichere Basis zur Datenübertragung gegeben. Durch die wenigen Stationen, die passiert werden müssen und die hohe Kapazität der Leitungen, werden Flaschenhälse weitestgehend vermieden. Zur Fertigstellung der restlichen fünf Gebäude wird ein zweiter Switch angeschafft, in der gleichen Größenordnung des bisherigen Backbones, der die neuen Gebäude in gleicher Art und Weise verbindet. Um zwischen alter und neuer Infrastruktur keinen Flaschenhals zu bekommen, werden die beiden Hauptswitches mit sechs Trunkports verbunden. Diese beiden Switches bilden heute logisch betrachtet ein Gerät, den Backbone des Campus Koblenz. Die Anbindung des Campus nach Außen erfolgt direkt an das Landeshochschulnetz. Die beiden Hauptswitches sind das Zentrum des Netzwerks der heutigen, balancierten Baumtopologie. Eine visuelle Verdeutlichung erlauben die drei Grafiken 4.2 bis 4.4. Die grünen Rechtecke repräsentieren Switches innerhalb der Gebäude, die beiden rosafarbenen die beiden Backboneelemente. Abbildung 4.3 zeigt die Switches in der Anordnung bezüglich der Gebäude, während in 4.4 eine Umsortierung der Rechtecke erfolgte, zur besseren Verdeutlichung der Baumtopologie. Der Großteil der Gebäudeswitches auf den Etagen sind direkt am Hauptswitch angeschlossen und bilden somit einen echten Stern. Eine Ausnahme bilden einige Switches im Serverraum zur Anbindung der Server. Theoretisch betrachtet hat jedes Datenpaket maximal drei Switches zu passieren, um an sein Ziel innerhalb des Campus zu kommen. Jeder Switch auf den Etagen, ist in einem eigenen Raum untergebracht, in dem ein Patchpanel[5] verbaut ist. Eine symbolische Darstellung ist in Abbildung 4.5 zu erkennen. Das Panel ist fest mit den Netzwerkdosen der jeweiligen Büroraumverkabelung verbunden. Die Verkabelung aller Räume ist vollständig installiert, wird aber zum gegebenen Zeitpunkt nicht im vollen Ausmaß genutzt, da nicht jede der vorbereiteten Leitungen benötigt wird. Bislang sind in den Büros je eine Person mit einem netzwerkfähigen Computer untergebracht. In den letzten Jahren hat die Personalzahl zugenommen. Damit verbunden werden die Räume knapper, so dass die Büros von mehreren Personen geteilt werden und die brachliegenden Netzwerkkabel immer mehr zum Einsatz kommen. Im Verteilerraum werden ein oder mehrere Switches aufgestellt, die die Verbindung zum restlichen Netzwerk ermöglichen, so dass das Minimum an benötigter Anschlussports abgedeckt ist. Die Netzwerkdosen der Büroräume sind bislang ohne Funktiona-

[3] Zentraler Netzwerkswitch im LAN, der die Basis eines Netzwerkes bildet. In der Regel ist es der leistungsstärkste Switch mit einer sehr großen Backplane.

[4] Lichtwellenleiter, der auch auf große Distanzen hohe Übertragungsraten erlaubt

[5] Ein Patchpanel ist eine Leiste von Anschlüssen, die in Ihrer Front frei veränderbar sind. Auf der Rückseite sind die Stecker fest mit Kabeln verbunden, die in der Regel nicht mehr verändert werden.

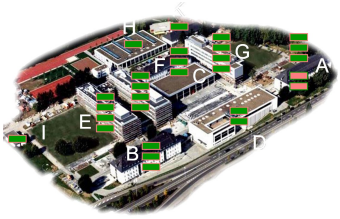

Abbildung 4.2:

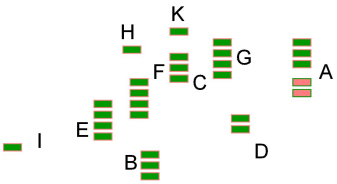

Abbildung 4.3:

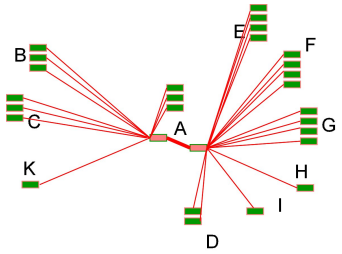

Abbildung 4.4:

lität. Erst die Verbindung eines Steckplatzes des Panels mit einem Port des Switches erzeugt die gewünschte Verbindung. Wird eine weiterer Computer in einem Raum aufgestellt, ist ein freier Port des Switches, unter Verwendung eines kurzen Patch-Kabels, mit dem zu dem Büro gehörigen Steckplatz des Panels, zu verbinden. Auf diese Weise wird die Zimmer-Netzwerkdose, in dem Raum der neuen Arbeitsstation, dem Netzwerk zugänglich gemacht. Dieser Vorgang erlaubt, die Netzwerkanschlussmöglichkeiten der Flure zu erhöhen, ohne neue Kabel in den Gebäuden legen zu müssen und die Verfügbarkeit des Anschlusses den jeweiligen Notwendigkeiten anzupassen.

Unter Umständen wird es in vereinzelten Räumen verlangt, dass mehrere Computer benötigt werden. Man betrachte den Einsatz eines Computerlabors. Reicht in diesem Fall die Menge an vorinstallierten Netzwerkdosen nicht aus, besteht die Möglichkeit einen weiteren Switch in diesem Raum aufzustellen und diesen mit einer der Netzwerkdosen zu verbinden. Die Computer, die bislang ohne Netzwerkanschluss sind, können an den hinzugefügten Switch angeschlossen werden. Es ist auf diesem Weg theoretisch möglich, beliebig viele Computer in einem Raum unterzubringen. Der Nachteil dieser Netzwerkerweiterung ist es, dass ein Flaschenhals entsteht. Ein Switch, der im Verteilerraum steht, hat einen Uplink[6], der mit eintausend MBit/s ausgestattet ist und ca. 50 Arbeitsstationen mit dem Netzwerk verbindet. Billigere Switches oder Hubs haben lediglich einen Uplink mit einhundert MBit/s, die 16 bis 24 Anschlüsse bereitstellen. Der zusätzliche hub oder Layer2-Switch bietet den Computern den Zugang zum Netzwerk, ist aber mit seinem 100 MBit/s Uplink an den einen Port des Etagenswitches angeschlossen. Im Endeffekt muss der Etagenswitch dann nicht nur einen Computer an diesem einenPort bedienen, wie vorgesehen, sondern unter Umständen bis zu 24 Stück. Eine Lösung, die die Performanz, für die angeschlossenen Computer, deutlich beeinträchtigt. Bei einem lokalen Netzwerk ist es jedem Benutzer möglich, das Netz beliebig zu erweitern, ohne das die Netzwerkadministratoren diese Veränderung mitbekommen. Die Planungen und Bemühungen ein performantes Netzwerk zu unterhalten, können von jedem Benutzer, oftmals durch Unwissenheit, negativ beeinflußt werden, indem Hubs und Switches eingefügt werden. Das Resultat ist ein Stern, der sich zu einem unbalancierten Baum wandelt.

Desweiteren unterhält die Universität ein Funknetz. Der Campus ist flächendeckend mit Accesspoints ausgestattet, so dass es jedem registrierten Benutzer der Uni möglich ist,

[6]Spezieller Anschluss, der Switch mit Switch verbindet

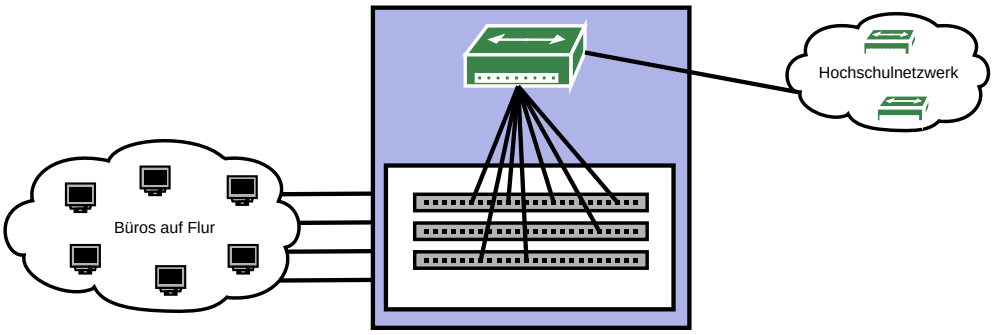

Abbildung 4.5: Symbolische Darstellung eines Patchschranks

per Funk das Netzwerk zu nutzen. Zu diesem Zweck sind die Gebäude mit Accesspoint ausgestattet, die als Brücke zum lokalen Netzwerk fungieren. Zudem sind vereinzelt Außenantennen angebracht, die es ermöglichen, auch im Freien eine Netzwerkverbindung aufzubauen. Funkwellen werden durch Beton und Stahl behindert. Als Resultat ist in jeder Etage eines Gebäudes an der Nord- und Südseite oder, falls geeignet, im Zentrum des Flurs ein Accesspoint angebracht, die mittels Kabel an den Etagenswitch angeschlossen ist. Ein VLAN[7] sorgt für eine Isolierung der Daten. Erst durch den Router erfolgt der Austausch mit dem lokalen Netzwerk.
Ein Netzwerk besteht nicht nur aus den Switches, Routern und Accesspoints. Server spielen eine entscheidende Rolle. Das Rechenzentrum ist, im Idealfall, mit mehreren Switches unmittelbar am Backbone der Topologie angeschlossen. An diesen Geräten werden die Server betrieben. Dieser Aufbau soll den Datenpaketen eine möglichst kurze Verbindung zwischen Server und Client ermöglichen. Da das Datenaufkommen für die Server deutlich höher erwartet wird sollten die Uplinks zum Backbone entsprechend Leistungsfähig dimensioniert sein. Auch wenn der durchschnittliche Datenverkehr die Grenze der möglichen Übertragung nicht erreicht, so sollte in Spitzenzeiten keine Beeinträchtigung erfolgen.Die meisten Server und einige Switches sind in einem klimatisierten Raum untergebracht. Somit sind alle relevanten, zu betreuenden Server, an einem Ort, ohne lange Wege. Durch den Einsatz mehrerer Switches ist eine Lastverteilung möglich. Einige Institute, for allem der Informatik, nutzen eigene Server, die selbst gewartet werden. Die Verfügbarkeit liegt nicht in der Hand des Rechenzentrums. Dennoch ist bei der Überwachung die Erreichbarkeit durch die Infrastruktur zu gewährleisten. Daher ist es sinnvoll, dass die Administratoren von der Existenz eines fremden Servers wissen.

[7] logisch abgetrenntes Netzwerk

4.2 Logische Umsetzung der Campus-Topologie

Als Basis eines Netzwerkmanagementsystems ist es notwendig, die vorhandene, physische Topologie sowie Ihre Veränderungen, logisch darzustellen und die Daten zur Weiterverarbeitung aufzubereiten. Es ist eine geeignete Beschreibung der Örtlichkeiten zu finden, die eine Lokalisierung von Switches und Hosts ermöglicht. Eine Beschreibung kann fest im Gerät codiert sein, oder unter Zuhilfenahme von örtlich gegebener Identifikatoren. Eine Variante beschreibt die Nutzung der SNMP-Variablen *Location*, einsetzbar für Geräte, deren Position in der Regel einmal fest vergeben wird und zu erwarten ist, dass das Gerät zum Einen in geraumer Zeit nicht verändert wird und zum Anderen nur von Personal verwaltet wird, welches zu den eingeweihten Administratoren des Netzwerks gehört. Die Alternative verwendet eine dynamische Lokalisierung der Netzgerätes, in der durch Sekundärinformationen aus der Nachbarschaft, die Position ermittelt wird. In diesem Fall werden keine Daten in dem Gerät gespeichert und die Position durch Kombination verschiedener Informationsquellen zusammengeführt. Informationen können aus Datenbanken, die durch andere Arbeitsvorgänge existieren, gewonnen werden, die daraus die Beziehung zu einer bekannten Position herstellen. Der folgende Abschnitt erklärt eine Möglichkeit zur Lösung dieser Problematik.

4.2.1 Bezeichung der Netzwerkkomponenten

Das Rechenzentrum unterhält managebare Layer3-Switches der Firmen *Cisco* und *HP*. Um Daten aus den Layer3-Switches zu sammeln, wird das SNMP-Protokoll[8] genutzt. Der erste interessante Aspekt ist die Örtlichkeit, an der das Gerät aufgestellt ist. Bislang werden Switches ohne GPS[9] ausgestattet, mit dessen Hilfe man das gesuchte Gerät finden könnte, ähnlich wie es in der Automobilindustrie genutzt wird. Die Ortsangabe obliegt der Sorgfalt der Administration. Bei der Einrichtung und Konfiguration des Switches besteht die Möglichkeit die Variable *Location* aus der *System-MIB* zu setzen. Es stellt sich die Frage, wie diese Variable zu füllen ist, um mit einer Angabe möglichst viel Informationen unterzubringen. Die Informationen sollten möglichst feingranular sein, um eine eindeutige Bezeichnung zu finden. Eine Angabe wie *Universität Koblenz* ist in diesem Fall zu wenig. Um zum Beispiel die Einarbeitung eines neuen Mitarbeiters zu vereinfachen, sollte eine eindeutige Bezeichnung, unter Verwendung der exakten Raumnummer und Gebäudebezeichnung, erfolgen. Desweiteren ist es so für alle Mitarbeiter eine Erleichterung ein für sie bislang unbekanntes Gerät eindeutig lokalisieren zu können. Die Variable wird mit einem *SNMP-Get*-Befehl ausgelesen. Der erhaltene String kann von einem Programm untersucht werden, um die benötigte Information zu extrahieren und bereitzustellen. Um die Verarbeitung zu erleichtern, wird die *Location*-Angabe genormt. Das bedeutet, jedes eingesetzte Gerät muss sich dieser Regel unterwerfen. Geschieht dies nicht, wird es unter Umständen nicht vom System erkannt. Alle gewünschten Informationen werden in einem String untergebracht und mit einem Minuszeichen von einander getrennt. Die Informationen, die als interessant erachtet werden, sind:

[8] Kapitel 3.5.1 Seite 43
[9] Global Position System

Ort in diesem Fall der Campus Koblenz.

Gebäude für eine erste grobe Trennung.

Etage innerhalb eines jeden Gebäudes, als erste Verfeinerung.

Raumnummer die auf jeder Etage eindeutig ist.

Sonstige Informationen eine Bemerkung, wie zum Beispiel die Positionierung ein Accesspoint, der außerhalb des Gebäudes angebracht ist. Für diesen Fall gibt es derzeit keine eindeutige Bezeichnung, die eine jedermann verständliche Ortsangabe zulässt.

Die Informationen entsprechen dem Aufbau der Raumbezeichnungen der Universität Koblenz (vgl. Kapitel 4.1). Der Locationeintrag für den Switch *Cat35-032.uni-koblenz.de* im Serverraum des Rechenzentrums erhält zum Beispiel die Bezeichnung: *UniKo-A-021–*
.
Der erste Teil entspricht dem Ort, dem Campus Koblenz, gefolgt von der Nummer des Gebäudes *A*. Der dritte Teil enthält die eindeutige Raumnummer, die unmittelbar die Etageninformation enthält. Die erste Ziffer repräsentiert die Etage, wobei in Kellern ein *K* und im Souterrain des C-Gebäudes ein *S* verwendet wird. Somit ist es für jeden Mitarbeiter problemlos möglich, den Standort jedes vom Rechenzentrum verwalteten Gerätes, zu ermitteln. Desweiteren erlaubt diese Darstellung eine einfache elektronische Verarbeitung. Anhand des Minuszeichens ist eine Trennung der Informationen möglich und jede Angabe in Variablen zu speichern. Die Einbettung dieser Angaben erlaubt es, einen Zusammenhang der Geräte auf die räumliche Anordnung der Gebäude und Flure zu erstellen.

4.2.2 Dynamische Raumbezeichnung

Die Positionsermittlung eines regulären Computers ist über die Information des *Location*-Eintrag eines Switches möglich. Der Host ist unmittelbar an den Etagenswitch angeschlossen, welcher das Netzwerk für jeden Flur bereitsstellt, auf dem auch der Computer aufgestellt ist. Aufgrund der Positionsinformation des Switches ist eine grobe Zuordnung eines angeschlossenen rechners möglich. Über den Switch ist die Information über Gebäude und Flur gegeben. Eine Alternative ist es, bei jeder Arbeitsstation SNMP zu aktivieren und die notwendigen Einträge, wie Location und Passwörter, zu setzten. Arbeitsstationen unterscheiden sich von Netzwerkgeräten darin, dass Arbeitsplatzrechner wesentlich öfter den Standort wechseln, als die Geräte, die die Infrastruktur bilden. Aus Erfahrung zeigt sich, dass der Computer vom Chef zum Mitarbeiter heruntergereicht wird, oder Mitarbeiter das Büro wechseln. Der administrative Aufwand wäre zu groß, jedesmal die SNMP-Einträge zu ändern. Aus Abschnitt 4.1 ist das Prinzip der Patchpanels bereits bekannt.
Das Patchpanels erlaubt es, anhand der Netzwerkdosen in einem Büro, die jeweilige Raumnummer dem Computer zuzuweisen, ohne diese am Rechner selbst einzutragen. Für die Umsetzung wird eine Tabelle in einer Datenbank angelegt, die es zu pflegen gilt.

Eine einmalige Arbeit mit geringem administrativen Aufwand. Die Datenbanktabelle enthält die Angaben Raum- und Netzwerkdosenbezeichnung sowie Panelbeschriftung und den zugehörige Port des Switches. Da Switch und Port bekannt sind, an dem der Rechner angeschlossen ist, ist die Beziehung zu dem Raum möglich. Lediglich wenn eine neue Dose eines Raumes gepatcht[10] wird, muss die Tabelle mit den Portnummern des Switches und der Bezeichnung des Panels aktualisiert werden. Die zusätzliche Pflege der Datenabnktabelle mag im ersten Moment als Mehrarbeit erscheinen, wird aber auf lange Sicht hin die administrative Arbeit erheblich erleichtern. Zudem ermöglicht die Inventarisierung der verwendeten Ports eine weitere Kalkulation der Netzwerkresourcen.
In den vorangegangenen Kapiteln wird beschrieben, dass Layer2-Switches und Hubs keine Informationen über die Belegung ihrer Ports bereitstellen können. Aus diesem Grund ist es notwendig, dass das komplette Netzwerk mit Switches der Stufe Layer3 ausgestattet ist. Bei der Verwendung niedrigerer Switchtechnik[11] entstehen Abstriche hinsichtlich der korrekten Raumangabe, da mehrere Rechner in unterschiedlichen Räumen, die Raumbezeichnung über den Port des Etagenswitches zugewiesen bekommen. Für den Fall, dass für eine Etage die Ports des Etagenswitches zu Neige gehen, ist es eine schlechte Variante ein Layer1/Layer2-Gerät zu verwenden, um die benötigten Ports zu schaffen. Zum Einen entsteht ein Flaschenhals am Uplinkport des neu integrierten Switches zu dem gemanagten Switch, und zum Anderen erscheinen die Rechner, die an dem neu angeschlossenen Switch hängen, exakt in dem einen Raum, für den der Port registriert ist. Eine genaue Lokalisierung ist so nicht mehr möglich. Ist der Switch innerhalb eines Seminarraumes aufgebaut und verbindet lediglich diese Rechner aus dem einen Raum, funktioniert zumindest die Bezeichnung für diese Gruppe Rechner. Wird die Lösung dafür verwendet, um mehrere unterschiedliche Büros anzuschließen, liefert die Angabe, dass ein gesuchter Rechner im Raum des Etagenswitches angeschlossen ist, keine brauchbare Information. In diesem Fall versagt die dynamische Lokalisation. Um den Administrationsaufwand gering zu halten, mit bestmöglichen Informationen, ist es zu empfehlen nur gemanagete Switches zu verwenden. *

[10] Der Anschluss des Panels zu der Dose in dem gewünschten Raum wird mit einem Netzwerkkabel an einen freien Port des Switches angeschlossen
[11] Geräte der OSI-Schichten Layer1(Hub) oder Layer2(Switch)

5 Verwendete Tools

Ziel der Arbeit ist es, ein Netzwerkmanagementsystem für das Rechenzentrum der Universität Koblenz zu entwickeln. Der Einkauf von Software entfällt zum Einen wegen der hohen Kosten und zum Anderen aufgrund der zu erwartenden, unzureichenden Anpassung hinsichtlich der individuellen Anforderungen. Um das Rad nicht neu zu erfinden, werden Werkzeuge gesucht, die dem Einsatzzweck nahe kommen und zugleich die gewünschte Individualität mit sich bringen.
Bei der Konzeption des Themas kristallisieren sich zwei wesentliche Vorstellungen heraus. Ein Ziel dieser Arbeit verlangt es, dass die Netzwerktopologie, beschrieben in Kapitel 4 ab Seite 57), des Campus voll automatisch erkundet und auf dem aktuellen Stand gehalten wird. Änderungen durch Austausch oder Erweiterungen von Switches müssen unmittelbar erkannt werden. Die Informationen müssen zur weiteren Verarbeitung anderer Tools zur Verfügung stehen. Bei Recherchen zu dem Thema Topologieerkundung wird eine Diplomarbeit [Barthel05] gefunden, die sich bereits mit dem Thema Topologie beschäftigt hat. In dieser Arbeit wird das Tool Netdisco erläutert, welches auf Basis von CDP (siehe Abschnitt 3.5.2), die Nachbarschaftsverhältnisse der Topologie, ermittelt und die Daten in einer Datenbank speichert. Nach der Erfassung der Switches und der Zuordnung zueinander wird weiterhin auf die Möglichkeiten von Netdisco gesetzt. Dieses bietet die Möglichkeit die an die Switches angeschlossenen Hosts zu finden, indem die MAC-Adresstabelle eines jeden Switch ausgelesen wird, und die Ergebnisse ebenfalls in der Datenbanktabelle speichert werden. Auf diese Weise wird die Beziehung von Host zu Switch festgehalten. Netdisco vollzieht keine besonders komplizierten Berechungen und wird verwendet, um den Entwicklungsaufwand zu minimieren.
Das zweite Ziel verfolgt die graphische Darstellung des Netzwerks in einer Form, die einfache Rückschlüsse auf die Position der Geräte erlaubt, um Zusammenhänge besser zu erkennen. Die erste Idee bestand darin, ein Graphenwerkzeug zu verwenden, um die ermittelten Daten visuell zu präsentieren. Eine statische Grafik ist nur bedingt brauchbar, daher wird ein weiteres Ziel verfolgt, der Einsatz eines Monitoringwerkzeug. Wegen der sehr guten Kritik der Fachzeitschrift *c't* [ct0306] über das Netzwerkmonitoringtool *Nagios*, wird dies als Basissystem verwendet. Es ist ein OpenSource-Programm[1] und erlaubt eine sehr feingranulare Konfiguration auf Basis von Textdateien. Hier verspricht man sich, die Konfigurationen über externe Programme automatisch erzeugen zu können. Zudem arbeitet das Programm mit dem Ampel-Prinzip, beschrieben in Abschnitt 5.2 auf Seite 71, zur Darstellung der unterschiedlichen Zustände überwachter Geräte.
Die von Netdisco gesammelten Daten, über Switches und Hosts, sollen mit Hilfe eines selbst entwickelten Programms verwendet werden, um die Nagioskonfigurationsdateien

[1] OpenSource oder Quelloffenheit bedeutet, dass der Quelltext des Programms frei erhältlich ist, und es erlaubt ist, diesen beliebig weiterzugeben und zu verändern.

automatisch zu erzeugen. Netdisco arbeitet von Nagios unabhängig. Es dient dazu, die Daten der Topologie zu sammeln und in einer Datenbank bereitzustellen. Nagios ist das Monitoringwerkzeug, welches zu konfigurieren ist. Nagios verlangt die Definition von Hosts und zugehörigen Service-Checks, die die eigentliche Monitoringarbeit erledigen. Die Hosts werden automatisch in Nagios integriert und mit einem Basischeck versehen, um eine funktionsfähige Konfiguration zu erzeugen. Im Anschluss definiert der Administrator weitere Services für ausgesuchte Hosts wie etwa DNS-, HTTP- oder Datenbankserver.
Bezogen auf die OSI-Prinzipien, beschrieben in Abschnitt 2.1, sind die Aspekte des Fault-Managements durch Nagios und das Configuration-Management durch Netdisco erfüllt. Das Performance-Management wird in gewisser Weise ebenfalls von Nagios angeboten, aber nicht entsprechende der Vorstellung der Arbeit. Ein letztes Kriterium betrifft die uneingeschränkte Erweiterung der Monitoringumgebung. Um nicht ausschließlich auf die Fähigkeiten von Nagios angewiesen zu sein, soll eine Erweiterung entwickelt werden, die von Nagios komplett unabhängig , aber in der Menüführung des Systems integriert ist. Zu diesem Zweck entsteht eine Auslastungsüberwachung der Switches mit Hilfe von MRTG[2]. Durch Verwendung von MRTG wird das dritte Prinzip, das Configuration-Management erfüllt. Das Programm ermittelt mit SNMP den aufkommenden Verkehr jedes einzelnen Ports eine Switches und erzeugt eine Webseite mit Graphiken, die über die Detailansicht eines Hosts, im Webinterface von Nagios, erreichbar sind.
Im Folgenden werden die einzelnen Tools vorgestellt, welche Aufgaben ihnen anvertraut sind und welche Ergänzungen notwendig sind.

5.1 Netdisco - Discovering

Netdisco steht zum Zeitpunkt der Entwicklung dieser Arbeit in der Version 0.64 zur Verfügung. Entwickelt wurde die Software von Max Baker an der Universität von Kalifornien, Santa Cruz (UCSC). Netdisco ist ursprünglich geteilt in einen Dienst, der die im Hintergrund Daten von den Netzwerkgeräten sammelt und ein Webinterface, welches die Daten aufbereitet und darstellt. Der Dienst ist in Perl implementiert. Bei dieser Scriptsprache ist der Sourcecode einsehbar und für individuelle Bedürfnisse veränderbar. Wird am Code etwas verändert, ist dies so zu tun, dass bei einem Update der Software, die Veränderungen nicht verloren gehen, oder mit wenigen Handgriffen einzupflegen sind. Das Webinterface basiert auf der Apache Version 1.3.x, welche in heutigen Distributionen nicht mehr vertrieben wird. Es ist daher notwendig die Sourcen des Apache und der verwendeten Perlmodule für die jeweilige Distribution selbst zu kompilieren, was bei Apache mit Komplikationen verbunden ist. Der Apache2.x ist zum aktuellen Zeitpunkt nicht in der Lage das Webinterface darzustellen, da eine große Anzahl benötigter Perl-Pakete nicht verfügbar, oder nicht mit Netdisco kompatibel sind. Der Netdisco-Dienst funktioniert ohne das Webinterface ohne Probleme und ist mit der aktuellen PostgreSQL[3] Datenbank kompatibel, so dass die Daten gesammelt und bereitgestellt werden

[2] Multi Router Traffic Grapper, http://oss.oetiker.ch/mrtg/
[3] http://www.postgres.org

können. An der Darstellung der Daten durch das Netdisco-Webinterface besteht kein direktes Interesse, da dies durch Nagios erfolgen soll.
Netdisco ist dafür konzipiert mittlere und große Netze, unter Verwendung des Nachbarschaftsprotokolls *CDP* (vgl. Abschnitt 3.5.2), nach den verwendeten Netzwerkkomponenten zu scannen und die Beziehungen der Geräte untereinander in Datenbanktabellen festzuhalten. Da Hosts in der Regel nicht mit dem CDP-Protokoll installiert sind, werden mit Hilfe der *Forwarding Data Base* die an die Switches angeschlossenen Hosts ermittelt. Da die *Forwarding Data Base* nur die MAC-Adresse eines Rechners speichert, sind weitere Optionen für Netdisco notwendig, die anhand des ARP-Caches der Switches, die IP-Adressen zu den gefundenen Host-MAC-Adressen ermitteln. Um einen weiteren Komfort zu bieten, existiert zu den Tabellen für MAC- und IP-Adressen eine weitere Tabelle, die mit Hilfe von Netbios den Rechnernamen zu einer IP bereitstellt. Die Sammlung der Informationen durch Netdisco ist, wie die folgenden Abschnitte zeigen, nicht sonderlich kompliziert.

Nachbarschaftserkundung durch CDP

Der Erkundungsprozess benötigt einen initialen Switch, von dem aus die Erkundung beginnt. Alle Informationen werden über SNMP aus den Geräten ausgelesen. Netdisco wird mit der Option *-r* und einer IP als Parameter aufgerufen. Im Idealfall wird die IP des Switches verwendet, welches als die Wurzel des Netzwerks gesehen wird. In der ersten Phase ermittelt Netdisco Angaben über das aktuelle Gerät. Es werden Hostname, DNS-Name, Standort usw. ermittelt und in der Tabelle *device* gespeichert. Sie dient als Referenz für die Switches des Netzwerks und beinhaltet Angaben, die das Gerät selbst betreffen. Die zweite Phase durchsucht den CDP-Cache des Switches nach Nachbarschaftsinformationen. Die CDP-MIB speichert für jeden Port eines Gerätes, das angeschlossene Nachbargerät mit Namen, IP und Port an, solange dieses CDP-fähig ist, und sich regelmäßig zu erkennen gibt. Die gefundenen Nachbar-IP-Adressen kommen in eine Warteschlange von Netdisco. Um die Nachbarschaftdaten für weitere Verarbeitungen bereitzustellen, werden die Nachbarschaftangaben in der Tabelle *device_port* hinterlegt. Die Tabelle enthält einen Eintrag für jeden Port jedes Switches. Neben den Nachbarschaftsangaben wird zum Beispiel vermerkt welche Geschwindigkeit ein Port verwendet, oder welchem VLAN dieser zugeordnet ist. Sobald die Datenextraktion des aktuellen Gerätes abgeschlossen ist, wird seine IP-Adresse aus der Warteschlange entfernt. Eine weitere Liste vermerkt die bereits untersuchten Geräte. Da der CDP-Cache die IP-Adressen der Nachbarn bereithält, ist es für ein Programm unkompliziert, jedes weitere Gerät direkt anzusprechen. Es folgt eine, „First in First out“, Abarbeitung der Liste von IP-Adressen, die von jedem weiteren, untersuchten Gerät immerfort gefüllt wird, bis es keine unbekannten Switches mehr gibt. Durch diese Vorgehensweise „hangelt“sich Netdisco von Switch zu Switch. Jedes weitere managebare Gerät kann durch die übergebenen IPs unmittelbar direkt angesprochen werden, ohne aufwendige Vergleiche und Zusammenführungen einzelner Daten aus unterschiedlichen Informationsquellen. Durch Abarbeitung der Warteschlange werden alle Geräte des Netzwerks gefunden, die CDP-Daten senden und speichern können.

Durch diese Vorgehensweise entsteht ein großes Problem, da Nicht-CDP-Switches nicht gefunden werden. Selbst CDP-fähige Geräte, die hinter einem Nicht-CDP-Gerät positioniert sind, werden nicht gefunden, da die Erkundung nach dem „Durchhangel-Prinzip“verläuft. Verhindert wird es durch die Multicastadresse, die von Layer3-Switches gefiltert werden und so die Information nicht über das Nicht-CDP-Gerät weitergereicht werden (vgl. Abschnitt 3.5.2). Für diesen Fall hat der Entwickler von Netdisco bereits einen weiteren Parameter *-R* definiert, der es erlaubt ein Gerät einzeln zu erkunden und mit Hilfe einer Textdatei in die Topologie aufzunehmen. Das Gerät wird auch korrekt in die Topologie eingebaut, wird aber in einem erneuten Durchlauf des Scans nicht miteinbezogen, so dass ein dahinterliegendes CDP-Gerät automatisch integriert werden könnte. Für diese Problematik ist eine Lösung zu finden.

Informationen der Forwarding-Database-Table

Jeder Switch, auch wenn nicht managebar, speichert für jeden seiner Ports die MAC-Adressen, die über ihn erreichbar sind. Dieser Adressspeicher ist flüchtig und verwirft nach einem gewissen Zeitraum die Informationen. Dies ist notwendig, damit zum Beispiel ein Umzug eines Rechners an einen anderen Port möglich ist. Aus diesem Grund ist das Auslesen der Tabelle in regelmäßigen, relativ kurzen, Zeitabschnitten notwendig. Zudem ist der Speicher eines Switches begrenzt. Nehmen die Einträge überhand, gehen die älteren Einträge verloren und nur die aktuellen MAC-Adressen werden im Speicher behalten. Netdisco bietet die Option *-m* an, die zu verwenden ist, wenn Netdisco mindestens einen Scandurchlauf absolviert hat und eine Reihe von Switches in der *device*-Tabelle sind. Bei Angabe dieser Option werden alle, in der Datenbank eingetragenen Switches, nach Einträgen ihrer *Forwarding Database Table* durchsucht. Jede MAC-Adresse wird in einer separaten Tabelle *node* mit einem Verweis auf die zugehörige Switch/Port Kombination eingetragen. Zusätzlich wird ein Zeitstempel vermerkt. Diese Zeitangabe ist relevant um zu entscheiden an welchem Switch eine MAC-Adresse zuletzt angeschlossen war, falls das zugehörige Gerät seinen Standort gewechselt hat oder der Switch getauscht wurde. Auf diese Weise entsteht eine Liste aller MAC-Adressen, die in dem Netzwerk Datenpakete senden oder empfangen. Der Zeitstempel erlaubt zudem eine Aussage über ehemalige Verbindungen und Erzeugung einer Historie für jede MAC-Adresse. Um eine Überflutung von Angaben zu unterbinden, prüft Netdisco im Vorherein, ob der betrachtete Port als Uplink fungiert. Die MAC-Adressen sind nicht von direkter Bedeutung, da die zugehörigen Rechner nicht an diesem Port unmittelbar angeschlossen sind, sondern nur der nächsten Switch, der Datenpakete zu den, mit den MAC-Adressen versehenen, Hosts weiterleitet.

Informationen der ARP-Tabelle

Sind MAC-Adressen in der *Forwarding Database Table* eines Switches enthalten, bewerkstelligt die Option *-a* die Ermittlung der zugehörigen IP-Adressen. Bei Aufruf von Netdisco mit dieser Option wird die Datenbanktabelle *devices*, die die Switches beinhaltet, durchlaufen und die zu jedem Switch zugehörige ARP-Tabelle nach Einträgen

durchsucht. Die ARP-Tabelle entsteht in Rechnern, durch auslesen der Paket-Header von Datenpakete, die den Switch passieren. Das OSI-Modell von Seite 28 beschreibt die Adressierung von Datenpaketen. Da ein Switch bei der regulären Ausübung seiner Aufgabe von jedem Datenpaketen nur den Header betrachtet, bleibt ihm alles Weitere, wie die IP-Adresse, verborgen. Für die Aufgabe eines Switches, ist nur die MAC-Adresse von Bedeutung. Managebare Switches haben einen ARP-Cache, da sie ein einfacher Computer sind. Der Cache wird aber nicht von den Angaben jedes vorbeilaufenden Pakets beeinflusst, da dieser lediglich die Angaben des Headers untersucht, der die MAC-Adresse von Sender und Empfänger enthält. Sind Pakete an den Switch adressiert, wie etwa eine SNMP-Anfrage der Managementstation, wird der ARP-Cache mit den Angaben der Managementstation ergänzt. Gleichfalls, wie bei Arbeitsplätzen und Servern, füllt sich der ARP-Cache mit den Adressen der Rechner mit denen kommuniziert wird. Aus diesem Grund, werden nur wenige Hosts in diesem registriert, da nur vereinzele Rechner mit den Switches selbst kommunizieren. Netdisco wird zu dem Großteil der ermittelten MAC-Adressen niemals die passenden IP-Adressen finden, wenn nur die Tabelle der Switches für die Auslesung des ARP-Caches herangezogen werden. Eine Lösung des Problems sind Router. Im Idealfall ist ein Router durch die Erkundung durch das CDP-Protokoll bekannt geworden. Ein Router entfernt den Header mit der MAC-Adressierung und verwendet die IP für die Weiterleitung des Datenpaketes. Ein Router ist auf diese Weise ein Rechner, der mit sehr vielen Computern des lokalen Netzwerks kommuniziert. Bei der Datenextrahierung des ARP-Caches des Routers, wird in der Regel eine sehr große Tabelle mit IP-Adressen ausgelesen, die in der Regel den meisten MAC-Adressen zugeordnet werden können.

Ermittlung von Rechnernamen

Netdisco ermöglicht durch Angabe der *-w* Option die Suche nach Rechnernamen. Über die ermittelten IP-Adressen werden mit Hilfe von *Netbios* die Rechnernamen ermittelt. Das hat den Nachteil, dass nur Windowsrechner und UNIX-Workstations mit aktivem Samba-/Netbiosdeamon gefunden werden können. In einem Netz, mit Knoten ausgestattet mit unterschiedlichen Betriebssystemen, reicht das nicht aus. Da, zum Beispiel das Netzwerk der Universität Koblenz, sehr viel Server und Arbeitsplatzrechner mit dem Betriebssystem Linux betreibt, ist diese Variante der Namensgewinnung nicht weiter interessant, da bei diesen der Name durch Netbios nicht ermittelt werden kann, wenn kein spezieller Dienst installiert worden ist.

Probleme mit Netdisco

Grob betrachtet ist Netdisco auf Windowsnetzwerke mit Ciscoswitches ausgerichtet. Das große Problem an Netdisco ist die alleinige Verwendung der Daten aus CDP-Cache der Switches zur Erkundung der Topologie. Um die Software ohne Anpassung nutzen zu können muss ein homogenes Cisconetzwerk, oder Switches die CDP voll unterstützen, mit Windowshosts vorliegen. Für diesen Spezialfall wird Netdisco ohne Probleme ein zufriedenstellendes Ergebnis liefern. Verwendet das Netzwerk Switches anderer Hersteller ist

darauf zu achten, dass diese vollständig CDP kompatibel sind. Neuere *Hewlett-Packard* Switches verwenden CDP lediglich noch passiv und unterstützen dafür vollständig LLDP. Passiv bedeutet, dass eingehende CDP-Multicasts angenommen und verarbeitet, CDP-Multicasts aber vom Gerät selbst nicht versendet werden. So entsteht die Situation, dass HP-Geräte ihre Cisco-Nachbarn kennen, aber die HP-Geräte für Cisco-Geräte unsichtbar sind, da sie keine CDP-Nachrichten von den HP-Nachbarn bekommen. Ein zweites Problem ist die singuläre Verwendung von Netbios zur Namenerkennung der IP-Adressen. Besonders Netzwerke in Universitäten verwenden vermehrt *Linux/Unix*, *Sun* oder *Macintosh*. Systeme mit diesen Betriebssystemen bleiben bei der Namensabfrage in Netdisco unberücksichtigt.
Ein weiterer wichtiger Aspekt im Umgang mit großen Netzen ist die Verwendung von Trunking einzelner Switches. Für den Administrator ist es von großer Bedeutung zu wissen, wo er welche Trunks eingesetzt sind. Aus diesem Grund wäre eine bessere Darstellung in den Netdiscotabellen erforderlich, die eine Übersicht über verwendete Trunkports erlaubt.

Lösungstrategien CDP

CDP liefert viele brauchbare Informationen, die über die MIB ohne Schwierigkeiten auslesbar sind. Die Informationen nicht zu verwenden, resultiere in einem programmiertechnischen Aufwand, der unverhältnismäßig groß ist, wenn die Daten mit einigen wenigen SNMP-Anfragen aus den Switches zu lesen sind. Ebenso stellt LLDP in gleicher Art und Weise wie CDP seine Informationen über eine spezielle MIB bereit. Beide Protokolle vertragen sich leider nicht und können Ihre Daten nicht gegenseitig ergänzen. Die MIBs sind unterschiedlich aufgebaut, was die Vorgehensweise beim Auslesen zudem erschwert. Durch die Verwendung unterschiedlicher Multicastadressen, die von den Switches gefiltert werden, werden die Nachbarschaftsinformationen nicht weitergereicht, sobald ein LLDP-Gerät als Nachbar eines CDP-Gerätes verwendet wird.
Eine Lösungsvariante wäre es darauf zu achten, dass CDP- und LLDP-Geräte separat voneinander gruppiert werden und nur an einer Stelle der Übergang durch den Administrator erfolgt. Dieser Ansatz ist allerdings nicht besonders elegant, da die Hardware nicht der Maßstab sein sollte, an welcher Stelle sie eingesetzt wird. Diese Entscheidung sollte weiterhin bei denjenigen liegen, die ein Netzwerk verwalten.
Ein Verfahren, das jeder Layer3-Switch unterstützt, ist der Spanning Tree. Mit seiner Hilfe ist es möglich geräteunabhängig Beziehungen auf MAC-Ebene herzustellen. Eine Idee zur Umsetzung von Nachbarschaften unter Verwendung der Daten des Spanning Trees wird ab Seite 96 beschrieben.

Lösungstrategien Computernamen

Die eindeutige Adressierung eines Rechners erfolgt über MAC- und IP- Adressen, wie in Abschnitt 3.4 bereits beschrieben. Zahlen sind allerdings nicht intuitiv, wohingegen Namen erlauben einfachere Assoziationen zulassen und daher besser zu merken sind. Die durch Netdisco ermittelten IP-Adressen, werden in der Tabelle *node_ip* festgehalten.

Netdisco benutzt durch Angabe der *-w Option* Netbios, um den Namen zu einer IP zu ermitteln. Die Namensauflösung unter Windows setzt einen laufenden Netbiosdeamon auf jeder Maschine voraus. Da das bei UNIX oftmals nicht zutrifft, wird eine Alternative gesucht.
Jeder Rechner wird bei der Installation mit einem Hostnamen versehen. Desweiteren befindet sich jeder Rechner in einer Domäne von Rechnern, woran deren Zusammengehörigkeit zu erkennen ist. In den meisten großen Rechnerumgebungen werden die IP-Adressen eines Computers nicht fest eingetragen, da der Verwaltungsaufwand zu groß ist. Um jedoch jeden Rechner mit einer IP-Adresse auszustatten, wird ein DHCP[4]-Server verwendet. Ein Computer kann so konfiguriert werden, dass dieser beim Einschalten über das Netzwerk nach einer IP-Adresse verlangt. Der Server nimmt diese Anfrage entgegen und teilt dem Computer eine IP-Adresse zu. Desweiteren besteht die Möglichkeit durch den DHCP-Server auch den Domänennamen eines Rechners mit zu teilen. Auf diese Weise lernt der Rechnern nicht nur seine IP-Adresse, sondern auch den für ihn zuständigen *DNS-Server*[5], der den Rechnernamen bei sich registriert. Die Aufgabe eines DNS ist in Abschnitt 3.4 erklärt. Durch diesen Dienst ist es mit Hilfe des Befehls *nslookup* möglich den Namen zu einer IP zu ermitteln, egal welches Betriebssystem.

5.2 Nagios - Monitoring

Monitoring macht sich besonders in Zeiten der Personalknappheit bezahlt. Automatisierte Prozesse übernehmen die manuelle Überprüfung der Systeme, die bislang von Hand durchzuführen waren. Die Menge der Überwachungsprozesse ist frei wählbar und unter Umständen auf mehrere Maschinen verteilbar. Ein besonderer Vorteil entsteht durch die mögliche, regelmäßige Abarbeitung, die ein Mensch in dieser Regelmäßigkeit nicht einzuhalten vermag. Das Open-Source-Werkzeug *Nagios* von Ethan Galstad übernimmt diese Aufgabe.
Das besondere Augenmerk liegt im Aufbau der Konfiguration des Systems und seinem Benachrichtigungssystem. Um eine Entscheidungskraft, ähnlich die eines Menschen zu simulieren, verwendet Nagios das *Ampelprinzip*. Die Idee basiert darauf, die Farb-Assoziationen aus dem täglichen Leben von Menschen zu benutzen. Es kann davon ausgegangen werden, dass jeder Mensch in industrialisierten Ländern die Farbcodierung einer Ampel aus dem Straßenverkehr versteht und erhofft so eine intuitive Interpretation bei der Verwendung der Software. Im Verkehr bedeutet die Farbe *Grün*, das für den Betreffenden alles OK ist, und dieser zum Beispiel die Kreuzung überqueren darf. Im Gegensatz dazu bedeutet *Rot*, das das gewünschte Vorhaben behindert, und im Beispiel die Überquerung der Kreuzung derzeit nicht möglich ist. Gelb wird im Straßenverkehr hingegen für Warnungen verwendet, wie die Vorbereitung auf einen Farbwechsel der Ampel, oder zur Markierung bei Verkehrsbehinderungen von Baustellen oder Fahrzeugen. Diese Codes macht sich Nagios zu Nutze, um die verschiedenen Status optisch hervorzuheben.

Grün signalisiert, daß alles in Ordnung ist. Der jeweilige Service auf dem Rechner funk-

[4]Dynamic Host Control Protokoll
[5]Domain Name Service

Abbildung 5.1: Nagios Startseite mit Warnsystem für den Campus Koblenz

tioniert erwartungsgemäß und der zugehörige Rechner ist erreichbar, da der Service auf dem Rechner arbeitet.

Gelb steht für eine Warnung. Es sind definierte Schwellwerte, in der Konfiguration der Serviceprüfungen, notwendig, ab deren Überschreitung die Warnung erfolgt. Warnungen sind noch kein Grund in Panik zu geraten. Sie sind wichtig für Statistiken und Beobachtungen seitens der Administratoren. Anhand ihrer Angaben obliegt es den Administratoren zu reagieren. Diese können aufgrund der Kenntnis des Netzes und der angeschlossenen Geräte entscheiden, ob es sich um einen temporären Fehler handelt. Durch Kontrollen am Netzwerk im laufenden Betrieb, welche der Benutzer nicht wahrnimmt, ist es möglich den Ursachen von Problemherden auf den Grund gehen, ohne das es zu merklichen Behinderungen für die Benutzer kommt. Je nach Ergebnis der Überprüfungen kann eine Entscheidung erfolgen, die eine Verbesserung oder Entlastung des Netzwerk mit sich zieht.

Rot beschreibt einen Zustand, der eine Behinderung für den Benutzer des Netzwerks darstellt. Wird ein Service Rot gekennzeichnet, bedeutet es, das dieser aus irgendwelchen Gründen nicht verfügbar ist und daher abhängige Anwender und Hardwarekomponenten zu ermitteln sind. Ein Arbeiten ist für einen oder mehrere Benutzer nicht möglich. Fällt ein Switch aus, so betrifft es eine Menge an Computer und unter Umständen sogar weitere Switches und abhängige Rechner. Fällt ein Dienst eines Servers aus, ist zu prüfen, ob der Rechner erreichbar ist. Bei Ausfall eines Dienstes sind meist viele Rechner betroffen, die anhand des Dienstes regional nicht

Host Group	Host Status Totals	Service Status Totals
Workstations_A (A-Gebaeude)	305 UP	156 OK 152 CRITICAL
Workstations_B (B-Gebaeude)	255 UP	70 OK 185 CRITICAL
Workstations_BIB (BIB-Gebaeude)	6 UP 6 PENDING	No matching services
Workstations_C (C-Gebaeude)	117 UP	21 OK 96 CRITICAL
Workstations_D (D-Gebaeude)	7 UP 5 PENDING	No matching services
Workstations_E (E-Gebaeude)	34 UP 42 PENDING	No matching services
Workstations_F (F-Gebaeude)	76 UP 57 PENDING	No matching services
Workstations_G (G-Gebaeude)	29 UP 54 PENDING	No matching services
Workstations_H (H-Gebaeude)	10 UP 6 PENDING	No matching services
Workstations_I (I-Gebaeude)	4 UP	No matching services
Workstations_K (K-Gebaeude)	3 UP 5 PENDING	No matching services
Workstations (hosts)	12 UP	11 OK 1 CRITICAL
Server des Campus (server)	5 UP	8 OK
Switches Campus Koblenz (switches)	49 UP 2 DOWN	49 OK 2 CRITICAL
Accesspoints WLAN (wlan-ap)	64 UP	64 OK

Abbildung 5.2: Zusammenfassung aller Gruppen mit aktuellen Zuständen

zuordbar sind und das Ausmaß der Störung nicht abzuschätzen ist. Ein Ausfall sollte niemals eintreten und kann durch Interpretation entsprechender Warnzustände verhindern werden. Das ist der Grund für den Einsatz eines Managementsystems

Durch die Verwendung von Schwellwerten, die für die Servicechecks in Nagios verwendet werden, ist es der Maschine möglich, Entscheidungen für Zustände zu treffen. Die Projektion der numerischen Angaben in ein visuelles System, unterstützt den Anwender auf einfache Art und Weise. Assoziationen über Farben ermöglichen auch ungeübten Anwendern ein intuitives Interpretieren der Informationen. In diesem Punkt unterscheidet sich Nagios von anderen Tools, da der Anwender nicht mit allen Daten, ob kritisch oder unkritisch, in Berührung kommt. Viele Tools zeigen in Diagrammen alle gesammelten Information. Selbst wenn die Informationen derzeit keine Relevanz haben, muß sich der Systembetreuer mit diesen auseinander setzen und selektieren was für ihn wichtig und unwichtig ist.
Das Hauptaugenmerk von Nagios sind Serviceprüfungen. Es muß für jeden Host mindestens ein Service definiert werden. Nagios setzt voraus, dass ein Gerät, welches zu überwachen ist, einen wichtigen Dienst anbietet, dem das eigentliche Interesse gilt. Erst

in zweiter Instanz, wenn der Service nicht verfügbar ist, wird eine Überprüfung der Erreichbarkeit des Rechners selbst durchgeführt. Durch die Verwendung von Plugins ermittelt Nagios den Zustand für den angegebenen Service. Die Plugins sind eigenständige Programme auf Kommandozeilenebene, ohne grafische Oberfläche, in einer beliebigen Programmiersprache. Jedes Plugin kann für sich verwendet werden, einzeln ohne die Einbindung in Nagios. Die Plugins müssen sich bei der Ausgabe Ihrer Daten an eine Norm halten, sodass diese von Nagios verarbeitbar sind. Jede Überwachung wird durch ein externes Plugin realisiert, welches jederzeit ausgewechselt werden kann.
Ein Webinterface bereitet die Daten graphisch auf. In mehrere Rubriken und Darstellungsformen unterteilt, werden die Informationen präsentiert. Unabhängig von der Darstellung verwendet Nagios ein Benachrichtigungssystem. Eine permanente Überwachung des Bildschirms auf Änderungen durch einen Administrator ist nicht effizient. Nagios entscheidet, ob und wann für einen überwachten Service eine Benachrichtigung erfolgen soll. Ein weiterer Schritt prüft die Zuständigkeit, welcher Systembetreuer in Kenntnis gesetzt werden soll. Wenn die Konfiguration für den zuständigen Kontakt eine Benachrichtigung erlaubt, wird die Benachrichtigung zugestellt. Dies ist auf unterschiedliche Arten wie E-Mail oder Pager realisierbar.

5.2.1 Objekte in Nagios

Nagios definiert die Konfigurationen in Objekten. Jede Komponente, unabhängig ob Arbeitsstation oder Switch, wird als Objekt definiert und mit den vorgesehenen, spezifischen Angaben erzeugt. Ähnlich wie in der objektorientierten Programmierung besteht die Möglichkeit der Vererbung. Angaben, die für alle, oder eine Klasse von Geräten gleich ist, wird separat als Template[6] definiert und in das jeweilige Objekt eingebunden. Die Schreibarbeit verringert sich und Änderungen sind an nur einer Stelle notwendig. Ein zusätzlicher Vorteil ist es, dass sich Fehler nur an einer Stelle einschleichen können. Die Möglichkeit der Vererbung ist aber nicht nur auf die Netzwerkkomponenten beschränkt, sondern gilt für jedes Objekt, gleich ob Kontaktpersonen oder Benachrichtigungsdefinitionen. Jedes Objekt kann als separate Datei im Filesystem definiert werden. Das erleichtert dem Programmierer die Erstellung automatischer Konfigurationsdateien, da keine Datei nach vorhandenen Einträgen durchsucht werden muss. Für eine nachvollziehbare Ordnung erlaubt Nagios die Angabe des Pfades einer jeden Konfigurationsdatei. Alternativ kann ein Ordner angegeben werden, der rekursiv nach Konfigurationsdateien durchsucht wird. Diese Möglichkeit erlaubt eine saubere Sortierung der Objektdefinitionen in Unterordner nach Netzwerkswitches, Server oder Hosts. Bei automatischer Erzeugung von Dateien kann es unter Umständen notwendig sein, diverse Konfigurationen, die automatisch erzeugt sind, auszusortieren. Das Selektieren der gesuchten Einträge gestaltet sich in einer Datei etwas aufwendiger. Liegen die Konfigurationen pro Objekt einzeln im Verzeichnisbaum, ist es leichter diese zu selektieren und ggf. an einen anderen Ort im Dateisystem zu verschieben.

[6] Englischer Begriff für Vorlage.

```
01: define host{
02:   use                     generic-device
03:   host_name               cat35-015.uni-koblenz.de
04:   hostgroups              switches
05:   address                 141.26.128.115
06:   alias                   cat35-015.uni-koblenz.de (UniKo-A-021--)
07:   parents                 cat6506-002,
08:   check_command           check-host-alive
09:   max_check_attempts      5
10:   notification_interval   30
11:   notification_period     24x7
12:   notification_options    d,u,r,f
13:   contact_groups          switchadmins,imapwriter
14: }
```

Abbildung 5.3: Konfigurationsdatei - Host

Hostdefinition

Die Definition eines Hosts beschriebt den Computer oder Switch. Abbildung 5.3 zeigt eine Beispielkonfiguration für einen Host. Das Schlüsselwort *use* aus Zeile 2 bindet eine Konfigurationdatei ein und demonstriert die Objektvererbung in Nagios. In der Definition *generic-device* werden eine Parameter definiert, die für alle Objekte der Gruppe *switches* gleich ist. Auf diese Weise ist eine Änderung für eine ganze Gruppe von Hosts nur an einer Stelle notwendig. Die folgenden Parameter beschreiben das Objekt mit Namen, IP und zu welcher Gruppe es gehört. Die Angaben von Zeile 9 bis 12 betreffen Informationen zur Benachrichtigung des Hosts, was im Abschnitt 5.2.1 genauer beschrieben wird. Um in Nagios Zuordnungen und Zuständigkeiten zu erreichen, existiert der Parameter *hostgroups* aus Zeile 13. Mit Hilfe dieser Angabe ist es möglich den Host einer oder mehrerer Zuständigkeitsgruppen zuzuweisen, wie etwa *Server*, *Linuxmaschine* oder *Switch*. Der Host kann deswegen mehreren Gruppen zugeordnet werden, da es als Hilfe bei der Orientierung innerhalb des Webinterfaces dient, und um Kontaktgruppen feiner abstimmen zu können. Die *check_command* Angabe in Zeile 8 verlangt die Angabe des Kommandos, welches ausgeführt werden soll, wenn Nagios einen Hostcheck durchführen möchte. Im Regelfall ist ein simpler Erreichbarkeitstest mit Hilfe des *ping-Befehls* definiert. Besonderen Stellenwert wird der Option *parents* in Zeile 7 zugeordnet. An dieser Stelle wird der Name des Switches angegeben, mit dem eine direkte physische Netzwerkverbindung besteht. Über diese Angabe wird die komplette Topologie in Nagios abgebildet. Die Grafiken 5.4 und 5.5 zeigen die Baumdarstellung in Nagios bezüglich der Abhängigkeiten der Geräte untereinander, die durch die Angebe der *parents-Option* erfolgte.

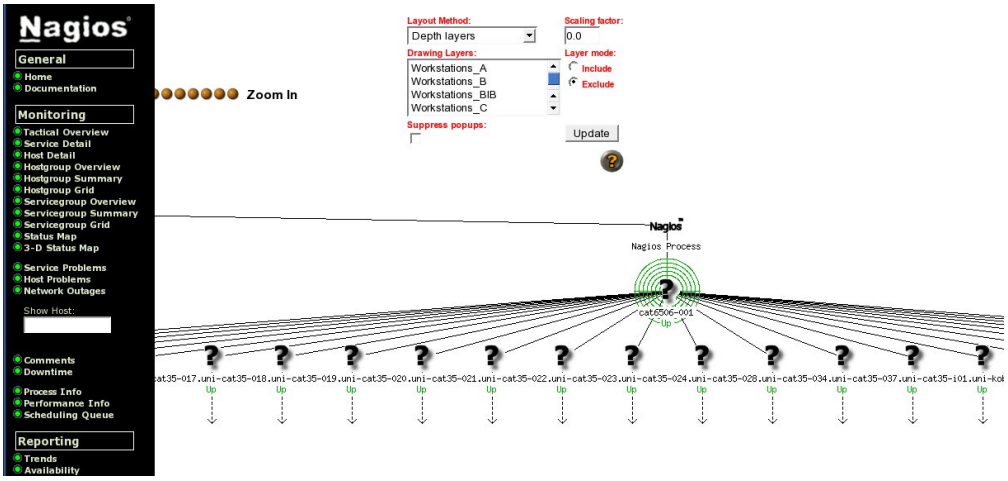

Abbildung 5.4: Baumdarstellung des Backbones mit dessen Kindknoten

Host- und Servicechecks

Host- und Servicechecks sind der erste von zwei bedeutenden Kernen in Nagios. Entgegen der Angabe in [Barth05], welches sich noch auf eine Beta2-Version bezieht, muss jedem Host, ab Nagios Version 2.0, zwingend ein Servicecheck zugeordnet sein. Ein Servicecheck prüft die Funktionsfähigkeit eines Dienstes. Auf Servern laufen Dienste wie zum Beispiel Mail- oder Webserver, Datenbanken, Fileserver oder Dienste wie DHCP[7] oder DNS[8]. Jeder Rechner im Netzwerk, der einen Dienst zur Verfügung stellt, hat seine Berechtigung, weil unterschiedliche Anwender ihn benötigen. Der Verwalter eines Netzwerks weiß nicht wann die Gruppe der Anwender einen Server nutzt. Deswegen ist zu gewährleisten, dass Serverdienste permanent zur Verfügung stehen, um Beeinträchtigungen der Anwender aufgrund von Serverausfällen zu vermeiden. Nagios nutzt zur Diensteprüfung Plugins, die durch frei definierbare Nagioskommandos (vgl. Abschnitt 5.2.1) angesprochen werden. Das Kommando führt das Plugin mit einigen Parametern aus und liefert den Rückgabewert an Nagios. Nagios erwartet in der Regel die Definition von zwei Schwellwerten. Der erste Wert definiert den *Warnzustand*, und ein weiterer, ab wann der Zustand als *kritisch* einzustufen ist.

OK Der Dienst funktioniert problemlos und liefert das Ergebnis in annehmbarer Zeit.

WARNING Der Dienst liefert Daten, aber nicht in dem Maße, wie es der Schwellwert des Plugins erwartet.

CRITICAL Der Dienst liefert die Daten in nicht akzeptabler Zeit, oder im schlimmsten Fall gar nicht.

[7] Dynamic Host Configuration Protokoll
[8] Dynamic Name Service

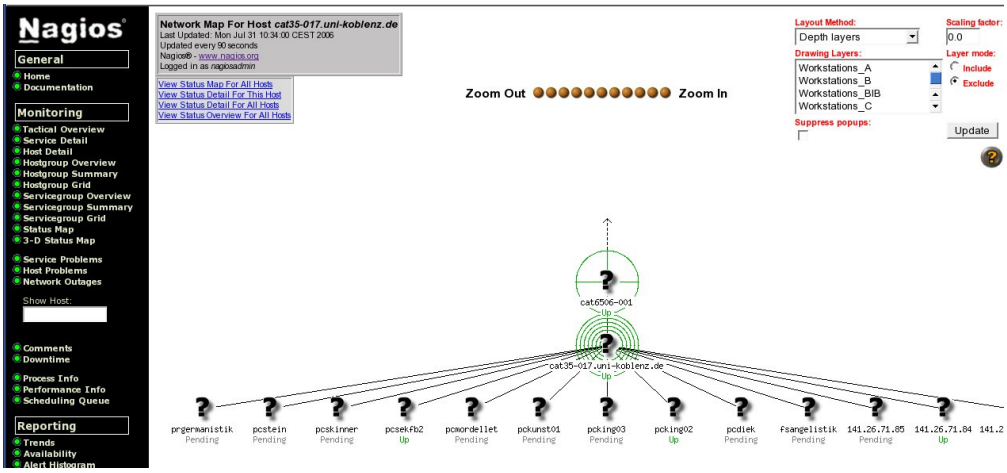

Abbildung 5.5: Baumdarstellung für den Switch Cat35-017 mit Eltern- und Kindknoten

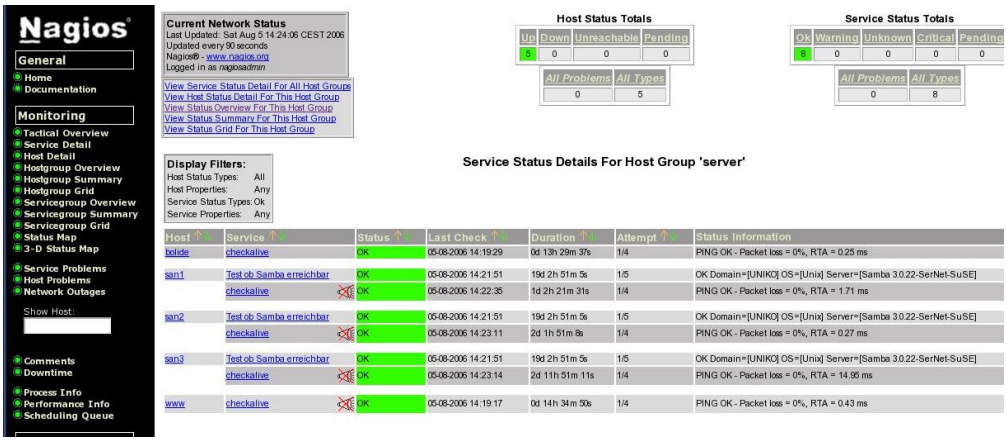

Abbildung 5.6: Darstellung verschiedener Servicechecks für die Gruppe der Server

```
01: define service{
02:   use                     generic-service
03:   hostgroup_name          switches
04:   service_description     PING-Switch
05:   contact_groups          switchadmins,imapwriter
06:   check_period            24x7
07:   max_check_attempts      5
08:   normal_check_interval   5
09:   retry_check_interval    1
10:   notifications_enabled   1
11:   notification_options    w,u,c,r
12:   notification_interval   10
13:   notification_period     24x7
14:   check_command           check_ping!100.0,30%!500.0,70%
15: }
```

Abbildung 5.7: Konfigurationsdatei - Service

UNKNOWN Das Plugin erhält Daten, für die kein eindeutiger Zustand definiert ist.

Einer dieser vier Zustände muß die erste Angabe in der Ausgabezeile des Plugins sein, damit Nagios die Daten verwerten kann. Nagios ist ohne Plugins wertlos. Nagios selbst kann die Antworten der Dienste nicht interpretieren und ist auf die Zustandsrückgabewerte der Plugins angewiesen. Der Rückgabewert beeinflusst das Verhalten von Nagios, um zum Beispiel Nachrichten zu versenden. Der Rest der Pluginausgabe sind Detailinformationen über den Zustand zur Darstellung in der Informationszeile der Detailansicht. Nagios selbst hat mit der eigentlichen Prüfung des Dienstes nichts zu tun. Das Plugin ist alleinig für die Korrektheit einer Dienstprüfung zuständig und liefert nur die Rückgabe an Nagios. Arbeitet ein Plugin nicht gemäß der gewünschten Anforderungen, steht es dem Nagiosadministrator frei, ein anderes Plugin zu verwenden. Es gibt keine feste Menge an Plugins. Ist für einen Dienst kein Plugin vorhanden, ist es mit jeder beliebigen Programmiersprache möglich, selbst ein Plugin zu entwickeln, welches den Vorstellungen des Administrators entspricht. Nagios ist in diesem Punkt völlig frei konfigurierbar.
Der Aufbau der Servicechecks ist ähnlich dem der Hostchecks. Die Abbildung 5.7 zeigt eine Servicecheckdefinition in Form eines PING zur Erreichbarkeitsprüfung der Switches. Der Kern des Objekts stellt der Parameter *check_command* dar. An dieser Stelle wird das Plugin aufgerufen mit den benötigten Parametern, die meist die Schwellwerte für die Zustände *WARNING* und *CRITICAL* angeben. Die folgenden Parameter sind innerhalb der Servicecheckdefinition notwendig, um Prüfung eines Dienstes zu steuern.

max_check_attempts verlangt einen Integerwert, der die Anzahl der Pluginaufrufe angibt, bis der Zustand in den Status *Critical* wechselt.

normal_check_intervall gibt den Zeitabstand in Minuten an, das die Ausführung des Plugins angestoßen werden soll.

retry_check_interval ist der Minutentakt, in dem die Wiederholung des Plugins erfolgen soll, wenn der Zustand gerade gewechselt hat. Ziel ist es, den neuen Zustand mehrfach verifizieren, um Übertragungsfehler oder eine kurzzeitigen Ausfall zu registrieren.

check_period ist eine von beliebig vielen frei definierbaren Zeiträumen, in denen der Servicecheck durchzuführen ist. Das gibt die Möglichkeit unterschiedliche Zeitfenster zu definieren, in denen das Plugin ausgeführt wird.

notification_period ist wiederum ein Zeitfenster, in welchem eine Benachrichtigung für das Plugin erfolgen soll.

contact_groups verlangt die Angabe mindestens einer der zu definierenden Kontaktgruppen, die für diese Prüfung zuständig sind und ggf. zu informieren sind.

Kommandokonfiguration

Die freie Definition von Kommandos (vgl. Abbildung 5.8 ist eine weitere herausragende Eigenschaft von Nagios. Die Anwender von Nagios sind nicht auf eine Sammlung von vorgegebenen Kommandos angewiesen, sondern es können beliebige Kommandos selbst definiert werden. So besteht die Möglichkeit über die Definition eines Kommandos jedes beliebige Programm unter UNIX, oder eigene Implementation anzusprechen. Für die Verwendung von Werten aus Nagios selbst, wie zum Beispiel der Hostname oder der aktuelle Status, stehen eine Reihe von Variablen zur Verfügung, die der Dokumentation zu entnehmen sind. Das anzusprechende Programm kann als Übergabewert diese Nagiosvariablen übergeben bekommen. Durch die Kommandodefinition ist eine Schnittstelle nach Außen gegeben für andere Tools, die die Daten von Nagios verwenden. Für einen Host- oder Servicecheck kann nicht nur ein Kommando angegeben werden, sondern beliebig viele, die gleichzeitig ausgeführt werden. Ähnlich bei der Verwendung von Plugins für Servicechecks, wird in den Kommandodefinitionen eine Kommandozeile mit variablen Parametern definiert. Es können alle auf der Maschine verfügbaren Tools angesprochen werden. Das auszuführende Kommando wird als String in der Datei eingetragen, als ob es in einer regulären Shell eingegeben wird. Die spezifischen Angaben, wie Rechnernamen oder Argumente, werden durch Variablen ersetzt. So ist es möglich, beliebige Kommandos für jeden Einsatzzweck zu definieren, die über die Option *check_command* bei den Servicechecks oder *service_notification_command* in der Kontaktdefinition verwendet werden können. Im Beispiel aus Abbildung 5.8 wird der Kommandoname *check_ping* in Zeile 2 vergeben. Zeile 3 ist das eigentliche Kommando, wie es, zum Beispiel in der *Bash*, mit den jeweiligen Werten, anzugeben wäre. In diesem Fall wird das Plugin *check_ping* aus dem Pluginordner von Nagios verwendet. Zufälligerweise ist der Kommandoname gleich dem Pluginnamen, wäre aber wie in den folgenden beiden Kommandos beliebig wählbar. Die Variablen $ARG1$ und $ARG2$ werden mit den Werten, nach dem Ausrufungszeichen, aus Zeile 14 der Abbildung 5.7 gefüllt. Die beiden folgenden Kommandos dienen dazu, ein eigens entwickeltes Perl-Script aufzurufen für weitere Datenbankeinträge. Die Bedeutung des Scripts wird in Abschnitt 6.4 erklärt. Die Zeilen 7 und 11 sind

```
01: define command{
02:   command_name check_ping
03:   command_line $USER1$/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
04: }
05: define command{
06:   command_name service_imapdata
07:   command_line $USER1$/imapdata.pl -H $HOSTADDRESS$ -S $SERVICESTATE$
08: }
09: define command{
10:   command_name host_imapdata
11:   command_line $USER1$/imapdata.pl -H $HOSTADDRESS$ -S $HOSTSTATE$
12: }
```

Abbildung 5.8: Konfigurationsdatei - Kommando

die eigentliche UNIX-Befehlzeile. Nagios ersetzt, die in Dollar eingefassten, Variablen mit den jeweiligen Werten des problembehafteten Gerätes. Die von vom Nagiosentwickler benutzen Variablenbezeichnungen sind alle einleuchtend gewählt und erklären sich von selbst.

Benachrichtigungssystem

Das Benachrichtigungssystem in Nagios bedarf eine detaillierten Konfiguration von Zuständigkeiten, Personengruppen und Zeitfenstern. Die Definition von Zeitfenstern (timeperiods) stellt eine wichtige Basis bei der Benachrichtigung dar. Nagios ist in der Lage durch Verwendung von Zeitfenstern Nachrichten und Checks nur zu vorgegebenen Zeiten durchzuführen. Das Ziel ist dabei Informationen nur in solchen Zeiten zu senden, in denen eine Reaktion des Systems und der Verwalter möglich bzw. notwendig ist. Wenn Services zu speziellen Zeiten nicht benötigt werden, ist es auch nicht notwendig Alarm zu geben, falls dieser in dem Zeitraum nicht verfügbar ist. Ebenso ist es unnötig einem Administrator eine Nachricht zukommen zu lassen, wenn er in dem Moment keine Schicht hat, sondern ein Kollege für den betroffenen Bereich zuständig ist. So erreicht den jeweiligen Mitarbeiter nur eine Nachricht, auf die er unmittelbar reagieren kann. Würden Nachrichten zu Zeiten versendet werden, in denen der Mitarbeiter diese nicht lesen kann, würde sich eine Menge an Nachrichten ansammeln, die alle das gleiche Problem beschreiben, aber

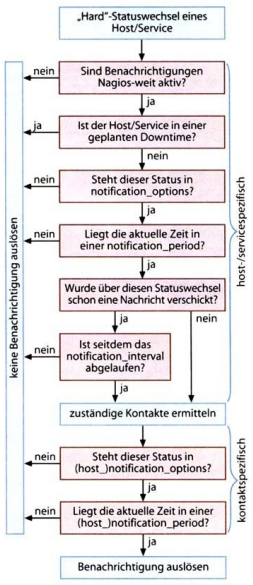

Abbildung 5.9: Nachrichtenfilter

die Mailbox verstopfen. Die Definition benötigt einen Namen zur Identifikation und die Angabe der sieben Wochentage. Für jeden Tag sind Uhrzeitbereiche anzugeben. Es besteht die Möglichkeit mehrere Bereiche, durch Komma getrennt, anzugeben, um zum Beispiel die Mittagspause zu berücksichtigen. Für jede Person, die an der Überwachung der Systeme beteiligt ist, wird ein Kontakt angelegt. Notwendig ist dies zum Einen für das Benachrichtigungssystem und zum Anderen für die Definition von Zugriffsrechten innerhalb des Webinterface. Nach der Betitelung und Beschreibung erfolgt die Angabe, welche der Zeitfensterdefinitionen für Benachrichtigungen bei Service- und Hostchecks gültig sind, gefolgt von den Angaben für die Zustände, bei denen eine Aktion erfolgen soll. Die Angaben entsprechen der Statusmeldungen von Nagios. Services unterschieden die Zustände:

w Warning

u Unknown

c Critical

r Recovery

Hostangaben differenzieren:

d Down

u Unreachable

r Recovery

Weiterhin ist zu definieren, wie die Kontaktperson die Nachricht erhalten soll. Hier wird ebenso in Host und Service unterschieden. Die Parameter erlauben eine Liste an Kommandos, die frei definierbar als Konfigurationsdatei in Nagios vorliegen. Für die Konfigurationen gibt es einige Standardvariablen, die verwendet werden können. Eine, die zwingen in jedem Kontakt anzugeben ist, ist die Email-Adresse.
Die eigentliche Brillanz von Nagios liegt im Nachrichtenfilter, welcher in Abbildung 5.9 in sehr ausführlicher Weise dargestellt ist. Grundlage der zur Auslösung eines Benachrichtigung, ist ein Statuswechsel. Es erfolgt eine Prüfung, ob Nachrichten generell erlaubt sind, und ob sich das Objekt nicht evtl. in einer Wartungsphase befindet. Für jeden Host und Service kann definiert werden, für welchen Zustand eine Auslösung zur Benachrichtigung erfolgen soll. Schliesslich wird geprüft, ob der aktuelle Zeitpunkt relevant für eine Nachricht ist. Die Zeiträume können in Tagen und mehrere Uhrzeitbereiche unterschieden werden. Desweiteren wird geprüft, ob bereits eine Benachrichtigung erfolgt ist, ob weitere erwünscht sind, bzw. das Benachrichtigungsinterval schon abgelaufen ist. Ab diesem Punkt werden die zuständigen Kontakte ermittelt. Wie in den Kontaktinformationen zu sehen, existieren hier ebenfalls die Parameter für den Zeitraum und die Art des Zustands. Diese können sich von Host/Service zu Kontakt unterscheiden und erlauben so eine Individualisierung bzgl. der Personen. Nachdem alle Filter durchlaufen sind, wird das oder die eingetragenen Kommandos für jeden Kontakt ausgeführt, die nach den Filtern übrig sind.

5.2.2 Systemkonfiguration

Die Systemkonfiguration von Nagios teilt sich in zwei Dateien auf. Die erste Konfiguration *nagios.cfg* betrifft im wesentlichen Angaben für das Verhalten von Nagios. Dazu gehören die Angabe für den Ablageort von Logdateien oder die Auflistung aller einzelnen Konfigurationsdateien oder die Angabe des Pfades[9], unter dem rekursiv nach Konfigurationsdateien gesucht werden soll. Die Datei *nagiosċfg* erlaubt eine detaillierte Unterscheidung, welche Aktionen von Nagios mitgeloggt werden, oder wann z.B. Überprüfungen durchgeführt werden sollen und in welchem Abstand. Ein weiterer wichtiger Punkt betrifft den Neustart des Nagiosdeamons. In der Regel verwerfen Programme beim Beenden die nachträglich, im Webinterface gemachten, Einstellungen und Informationen, die im Laufe des Programmdurchlaufs gesammelt worden sind. Nagios erlaubt Optionen, in denen dies nicht der Fall ist. Wird es gewünscht, dass alle Einstellungen, die nachträgliche im Interface gemacht worden sind, erhalten bleiben, so ist die Variable *use_retained_program_state=1* zu setzen. Ähnliches erlaubt die Angabe von *retain_state_information=1*, die zum Zeitpunkt des Ausschaltens des Deamons alle aktuellen Zustände der überwachten Geräte speichert. Derzeit erfolgt dies in einer Textdatei, die beim Neustart des Systems ausgelesen wird. So bleiben die gesammelten Daten und Einstellungen, trotz eines Systemausfalls erhalten.
Was in der einen Situation ein Segen ist, ist in anderen Fällen ein Fluch. Werden die Angaben in der Hauptkonfigurationsdatei geändert und das System neu gestartet, wird sich nichts am System ändern. Die Einstellungen, die sich das System im laufenden Betrieb merkt, haben Priorität gegenüber den Konfigurationsdateien. Angaben, die in der Konfigurationsdatei geändert werden, sollten auch über das Webinterface durchgeführt werden. Neu hinzukommende Konfigurationen für einen Host oder Service entnehmen zur Initialisierung die für sie angelegte Konfigurationsdatei. Alternativ ist die Angabe *use_retained_program_state=0* zu setzen, Nagios neu zu starten und anschließend, je nach Bedarf, der Parameter wieder auf eins zu setzen. Die erste Variante, der doppelten Änderung der Parameter, erscheint im ersten Moment aufwendiger, aber im Arbeitsrhytmus deutlich brauchbarer, da das System nicht beendet wird.
Die zweite Konfigurationsdatei, *cgi.cfg*, betrifft Einstellungen zu den verwendeten CGI-Scripten, die für die Darstellung der Daten zuständig sind und gleichzeitig eine Steuerung von Nagios im laufenden Betrieb ermöglichen. Die CGI-Scripte, die Nagios mitliefert, erlauben die Darstellung der Geräte und ihrer Historie. Viele Werte der Gerät- und Servicedefinitionen lassen sich über das Webinterface verändern. Nachrichten können deaktiviert oder Überprüfungen terminlich neu angesetzt werden. Wird ein Problem erkannt, besteht die Möglichkeit dem betreffenden Gerät einen Kommentar beizufügen, so dass die Kollegen informiert sind, dass sich bereits jemand dem Problem angenommen hat. Die differenzierte Darstellung von Systeminformationen und die Möglichkeit Nagios über das Web zu steuern wird über die Angabe der Kontaktgruppen ermöglicht.

[9] z.B. fg_dir=/etc/nagios/myconfigs

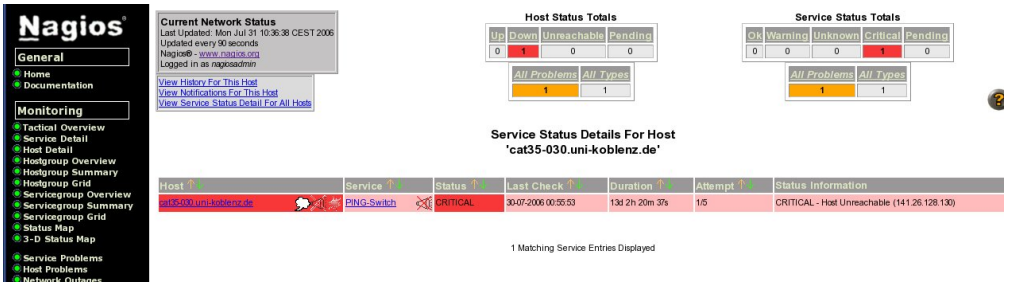

Abbildung 5.10: Darstellung eines Hosts, der durch ausbleiben des Ping-Checks, als abgeschaltet angezeigt wird.

5.2.3 Nagios-Weboberfläche

Die Vorstellung von einem Monitoringsystems lässt bei einem Laien sicherlich die Vermutung zu, dass es sich um ein System handelt, welches mit Hilfe eines Bildschirms die Zustände der überwachten Geräte permanent anzeigt. Dies Annahme ist sicherlich nicht falsch, wenn auch die Anwendung der Software in dieser Form nicht besonders effektiv ist. Ein gutes Monitoringsystem muss nicht ununterbrochen vom Systemverwalter über einen Bildschirm betrachtet werden, um im Falle eines Problems den optischen Hinweis zu erfassen, und reagieren zu können. Ein gutes System meldet sich bei dem zuständigen Verwalter und weist ihn auf den Fehler hin, so dass dieser im konkreten Fall, erst dann einen Blick auf den Monitor wirft.
Nagios bietet verschiedene Möglichkeiten, die überwachten Komponenten darzustellen. Eine gute Übersicht über die überwachten Objekte bietet die Abbildung *Hostgroup Summary*, auf Seite 77. Auf dieser werden alle definierten Hostgruppen angezeigt und wie viele der Objekte in einem guten oder kritischen Zustand sind. Die Aufmerksamkeit wird durch die bekannte Farbgebung Grün und Rot unterstrichen. Ein Mausklick auf einen rot markierten Zustand führt zur Darstellung aller kritischen Zustände dieser Gruppe. Abbildung 5.10 zeigt einen Beispielfall. In dieser ist zu erkennen, welcher Servicecheck ein Problem verursacht und wie lange dieses Problem bereits besteht. Das letzte Feld einer Anzeigezeile zeigt die Informationen für den Rückgabewert des Check-Plugins. Der Anwender hat in dieser Darstellung die Wahl, ob er Details bezüglich des Hosts, oder des zugehörigen Services betrachten möchte. In der Regel ist die Problemursache im Service zu suchen, welches anhand der Abbildung 5.11 in detaillierter Form zu sehen ist. An dieser Stelle ist es möglich, das Problem zu akzeptieren und einen Eintrag, in Form eines Kommentars, in das System zu machen. Das hat den Vorteil, dass jeder weitere Mitarbeiter dadurch in Kenntnis gesetzt ist, dass sich jemand mit dem Problem bereits beschäftigt hat. Nagios bietet in dieser Ansicht einige Kommandos an, die über das Webinterface steuerbar sind. Die Überprüfungen oder Benachrichtigungen können de- und aktiviert werden und Prüfungen können außer der Reihe angesetzt werden. Ist der Fehler akzeptiert, muss gehandelt werden. In der Regel ist zu prüfen, in wie weit es Abhängigkeiten der Geräte gibt, die an Problemsituation beteiligt sind, bzw. sollte die

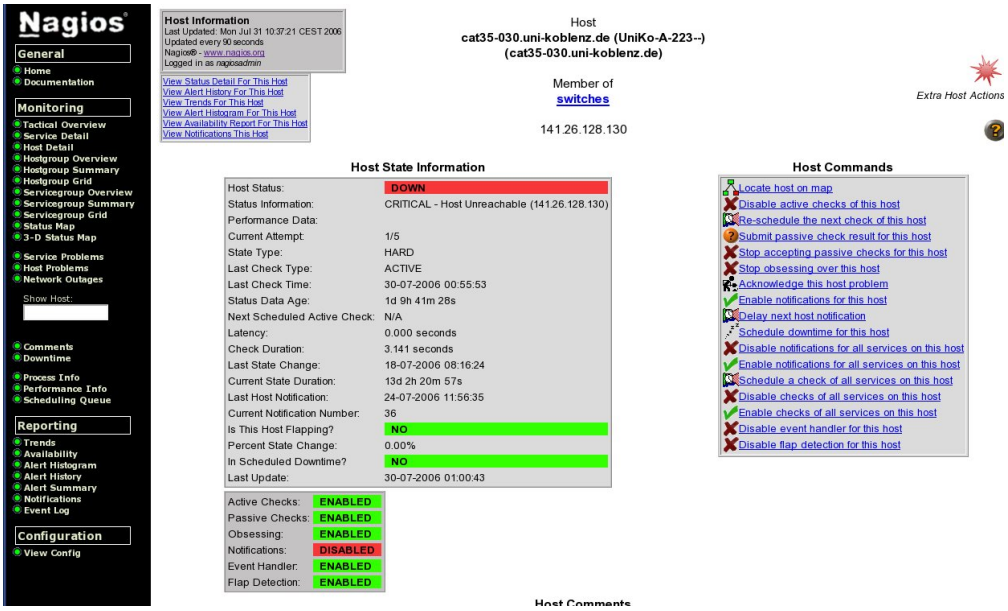

Abbildung 5.11: Ausführliche Anzeige des ausgefallenen Gerätes mit Steuerungsleiste an der rechten Seite.

Region bestimmbar sein, die beeinflusst ist. Zu diesem Zweck steht ein weiteres Kommando in der Detailansicht des Hosts zu Verfügung, der Menüpunkt *Locate host on map*. Dieser wechselt in die Ansicht 5.12, die den problembehafteten Host in der Baumdarstellung zeigt. Anhand des Elternknotens ist zu erkennen, dass dieser in Ordnung zu sein scheint und das Problem das Gerät selbst ist. Desweiteren zeigt diese Ansicht, dass derzeit kein Kindknoten an diesem Gerät zu finden ist und daher kein Host vererbt problembehaftet ist. Die Abhängigkeiten, wie sie in den Objekten durch den Parameter *parents* erzeugt werden, erlauben Nagios eine intelligente Berechnung. Fällt ein Switch aus, sind logischerweise die angeschlossenen Host nicht mehr erreichbar. Ein weniger ausgefeiltes System würde für jeden nicht erreichbaren Host eine Benachrichtigung versenden, da offensichtlich der Service-Check fehlschlägt. Aus den definierten Abhängigkeiten erkennt Nagios, dass eine Gruppe von Hosts an dem ausgefallenen Switch angeschlossen sind, und dieser Switch für den Ausfall verantwortlich sein kann. Aus diesem Grund erhalten diese Hosts nicht den Zustand *Down*, sondern *Unreachable*, da keine Aussage über diesen Host getroffen werden kann. Eine Benachrichtigung erfolgt nur für den ausgefallenen Switch. Durch dieses Verfahren wird die Benachrichtigungsflut gering gehalten. Für den Fall, dass die Kindknoten wiederum Switches enthalten, existiert in Nagios zusätzlich ein Menüpunkt *Network Outages*, welcher direkt den jeweiligen Switch nennt, der für den Ausfall eines ganzen Netzwerksegments verantwortlich ist und das Ausmaß besser zu erkennen ist.

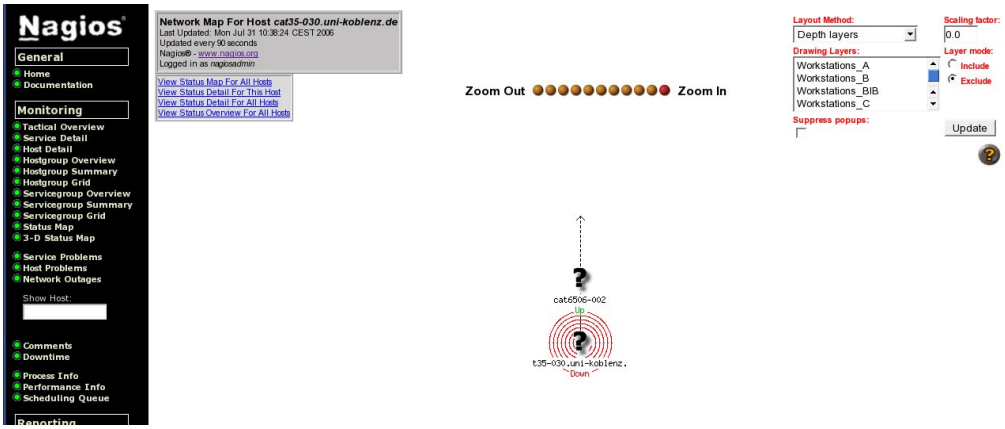

Abbildung 5.12: Ausgefallender Host mit zugehörigem funktionierendem Elternknoten

5.2.4 Kritik der Nagios-Weboberfläche

Ruft der Administrator das Webinterface von Nagios auf, sieht er links ein Menü und im Hauptfenster einige Informationen über die Version von Nagios und die Lizenzbestimmungen, aber keinerlei Informationen über aktuelle Probleme. Im Fehlerfall eines Gerätes sollte Nagios Problem auf der Startseite anzeigen, damit der Anwender mit möglichst wenigen Interaktionsschritten agieren kann. Nagios dient nicht als Zeitvertreib für den Administrator, um sich die Status der überwachten Komponenten anzusehen. Es ist ein System, welches für Probleme im Netzwerk entworfen ist, die es schnell zu lokalisieren und zu lösen gilt. Eine schnelle und aussagekräftige Informationskette ist daher das Wichtigste. Aufgrund dieses Mankos entsteht die Ergänzung eines optischen Warnsystems, welches ab Seite 104 im Kapitel 6 vorgestellt wird, und in die Startseite von Nagios integriert wird.

5.3 Multi Router Traffic Grapper

Das Tool MRTG[10] von Tobias Oetiker dient zur graphischen Darstellung der Messdaten von Netzwerkgeräte, die über SNMP auslesbar sind. Das Programm bietet eine Reihe von Anpassungsmöglichkeiten, so dass es nicht, wie ursprünglich vorgesehen, nur für die Netzwerkinterfaces von Routern, sondern zur Darstellung jeglicher möglicher Messdaten (Festplattenausnutzung, CPU-Load, etc) fähig ist, die über SNMP zur Verfügung gestellt werden. MRTG ist ein völlig eigenständiges Programm mit eigenen Konfigurationsdateien, die keinerlei Bezug zu Nagios haben. Für jedes zu Gerät, für das eine graphische Statistik erzeugt werden soll, wird eine Konfigurationsdatei angelegt. Mit Hilfe eines extra Konfigurationsprogramms von MRTG wird die Konfiguration erstellt.

[10] http://oss.oetiker.ch/mrtg/

Wird die Darstellung anderer Daten verlangt, muss die Konfiguration per Hand erzeugt werden. Die Konfigurationsdatei ist immer geräte- und messwertabhängig und für jedes Gerät einzeln anzulegen. Die Überwachung eines Gerätes für mehrere, unterschiedliche Messwerte wie CPU- oder Netzwerk- oder Festplattenauslastung in einer Konfigurationsdatei ist derzeit nicht möglich. Es ist daher notwendig für jede Messreihe eine eigene Konfiguration anzulegen. Bei einem sehr großen Netzwerk nimmt die Menge der Konfigurationen schnell ein unüberschaubares Maß an. Eine Möglichkeit, Ordnung in die Dateien zu bekommen, ist die Verteilung in Ordner.
Um die erwünschten Diagramme für die Messdaten, der zu überwachenden Geräte, zu bekommen, erfolgt ein Programmaufruf von MRTG mit der Konfigurationsdatei als Parameter. Der Aufruf ist für jede einzelne Konfigurationsdatei einzeln notwendig. Um den Vorgang zu automatisieren, besteht unter UNIX/Linux die Möglichkeit einen *Cronjob*[11] anzulegen, der in regelmäßigen Abständen das Programm mit einer Konfigurationsdatei aufruft. Nachteilig an diesem Verfahren ist es, dass bei der Überwachung vieler Geräte sehr viele Programmaufrufe notwendig sind. Das Einlesen der Konfiguration, sammeln der empfangenen Messdaten und die anschließende Erzeugung der HTML-Seiten und Grafiken benötigt Zeit. Bei einer großen Anzahl an zu überwachenden Geräten, die zum gleichen Zeitpunkt zu prüfen sind, gerät MRTG an seine Grenzen. Die sequenzielle Ausführung von MRTG erlaubt erst den Start des nachfolgenden Prozesses, wenn der Vorläufer fertig ist. In dieser Art der Anwendung stauen sich durch die Abfrage vieler Geräte eine große Menge an Prozessen. Aus diesem Grund ist der Aufruf in einem parallelisierten Modus zu empfehlen. Um zu gewährleisten, dass alle Konfigurationsdateien berücksichtigt werden, empfiehlt es sich MRTG über einen Cronjob zu steuern, der mit Hilfe einer Schleife ein komplettes Verzeichnis mit Konfigurationen abarbeitet.
MRTG verlangt in der Konfigurationsdatei ein *Working-Directory*[12], in dem alle Daten für die Auswertung und Speicherung in Textdateien abgelegt werden. Diese Dateien dienen als Basis für MRTG, um die Grafiken zu erzeugen, die ebenfalls in diesem Order abgelegt werden. Es ist darauf zu achten, das beim Einsatz von MRTG über eine große Anzahl von Geräten die Zeitdifferenz zwischen den Starts nicht zu gering gewählt ist, damit nicht der erste Switch wieder zu prüfen ist, wenn der letzte noch gar nicht angefangen hat. Die Abfragen durch SNMP brauchen Zeit, da alle Anfragen einzeln erfolgen. Zudem werden pro Interface 5 Grafiken erzeugt. Der erste Durchlauf einer Konfigurationsdatei dauert deutlich länger, da alle Dateien zur Speicherung der Messwerte, sowie die benötigten Webseiten angelegt werden. In den nachfolgenden Aufrufen werden lediglich die Grafiken erneuert. Der Server ist irgendwann überlastet und kann die Anfragen nicht mehr bewältigen. Im Szenario der Universität Koblenz werden derzeit ca. 60 Geräte überwacht, für die alle fünf Minuten Daten gesammelt werden. Der 3,0 GHz Dualrechner braucht ca. 2,5 Minuten für alle 60 Geräte.
Ein Kritikpunkt an MRTG, ist die Art der Datenabfrage. Unter Verwendung von *Ethereal* fällt auf, dass die Abfrage der Portdaten eines Switches in einzelnen Abfragen erfolgt.

[11] Ein Cronjob wird unter UNIX verwendet, Windows©nennt es *Sheduled Task*, um immer wiederkehrende Aufgaben zeitgemäß steuern zu können.
[12] Bezeichnet einen Order, der als Arbeitspfad dient, um temporäre und persistente Daten abzulegen
[12] www.ethereal.com/

Eine Alternative wäre der Einsatz von *snmpbulkget* oder *snmptable*. Das hätte zur Folge, dass für einen Switch die Informationen in einer Anfrage erfolgen würde und der Datenverkehr deutlich reduziert wäre. In Bezug auf diese Erkenntnis liegt es Nahe die Ermittlung der Performanzdaten in einer effizienteren Neuentwicklung zu ermitteln.

6 Systementwicklung und Implementationen

Nagios ist ein, in Fachkreisen, sehr gelobtes Werkzeug, welches bereits in einigen Unternehmen erfolgreich die Administratoren von Netzwerken unterstützt. Die Konfiguration aller Netzwerkkomponenten, Server und ggf. Arbeitsstationen ist in der Regel eine einmalige Aktion. Auch wenn dies oft nur einmalig durchgeführt werden muss, so nimmt es viel Zeit in Anspruch und der erfahrene Administrator weiß, dass viele Schritte automatisierbar sind. In stark wachsenden oder sich verändernden Umgebungen, wie einer Universität, nimmt die Pflege von Konfigurationsdateien sehr viel Zeit in Anspruch. Berücksichtigt man stressige Arbeitsumgebungen und deren Administratorenteams, die unweigerlich an Überlastung leiden, bleibt die Pflege eines solchen unterstützenden Systems, sicherlich oft aus Zeitgründen, auf der Strecke. Gerade dieses Umfeld ist auf ein stabiles Monitoring angewiesen, um die schwierige Situation zu entschärfen. Die unzuverlässige Pflege der Konfigurationen des Monitorings, erzeugt für die Betroffenen einen Teufelskreis des Chaos, obwohl der Einsatz von Systemen wie Nagios dem entgegen wirken und eine Entlastung des Teams erreichen soll. Aus diesen Erfahrungen heraus entsteht die Idee die Konfigurationen für Nagios so zu automatisieren, dass der Pflegeaufwand auf ein Minimum reduziert wird. Wie bereits in der Einleitung zu Netzwerkmanagement (Abschnitt 2) erwähnt, besteht ein Managementsystem nicht aus einem Werkzeug, sondern aus einer Sammlung von Tools, die sich ergänzen und gegenseitig Daten bereitstellen. Diese Intention wird auch in der vorliegenden Entwicklung aufgegriffen. Für die automatische Erkundung des Netzwerks wird das frei erhältliche Tool *Netdisco* verwendet, welches die Netzwerkkomponenten und die zugehörigen Beziehungen innerhalb der Topologie liefert. Eine eigene Implementation integriert die ermittelten Topologiedaten aus der Datenbank in das Monitoringsystem *Nagios*. Um einen besseren Überblick über die Auslastung der Switches zu bekommen, wird der Netzwerkverkehr jedes Switches aufgezeichnet, und über eine Webseite bereitgestellt. Für das Auslesen und Aufbereiten der Verkehrsdaten wird *MRTG* verwendet. Die Konfiguration von MRTG erfolgt ebenfalls vollautomatisch, wobei wiederum die Datenbankinformationen von Netdisco als Basis genommen werden. Abbildung 6.1 verdeutlicht das Zusammenspiel der drei Programme über die verwendetet PostgreSQL-Datenbank.
Für die Topologieerkundung war zuerst eine eigene Implementation geplant. Bei Recherchen zu diesem Thema wurde das Tool *Netdisco* im Internet gefunden. Der Beschreibung nach macht Netdisco exakt das, was die ursprüngliche Idee dieser Arbeit war, die Topologieerkundung einer unbekannten Umgebung. Da dieser Teil der Arbeit in diesem Sinn nicht mehr möglich war, wird das Thema erweitert in die automatische Konfiguration der Monitoringsoftware unter Berücksichtigung der Topologieänderungen.

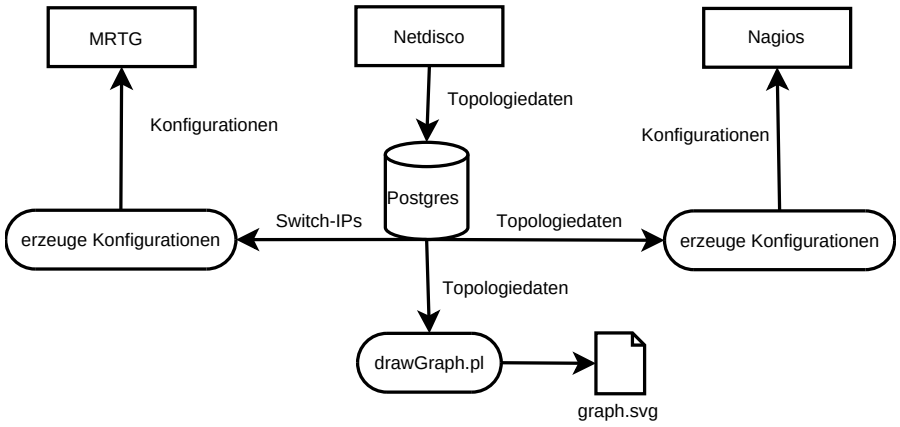

Abbildung 6.1: Beziehung von Topologieerfassung und Datenpräsentation

Netdisco übernimmt zu Beginn der Arbeit alleine die Topologieerkundung und erkennt den größten Teil der Netzwerkinfrastruktur. Die Verbindung der beiden Werkzeuge wird durch Entwicklung von Zusatztools realisiert. Die zusätzlichen Scripte sind notwendig, da weder Nagios, noch Netdisco Schnittstellen bereitstellen, die die beiden Programme verbinden würden. Erleichtert wird die Aufgabe dadurch, dass Netdisco den kompletten Datenbestand in einer PostgreSQL-Datenbank speichert, für die es zu fast jeder höheren Programmiersprache eine Schnittstelle gibt.
Unter heutigen Gesichtspunkten der Softwaretechnik sollte die Implementation in einer objektorientierten Sprache sein. Die erste Auswahl bietete C++ oder Java an. Da ich im Laufe des Studiums bereits beide Sprachen verwendet habe, lag die Verwendung einer der beiden nahe. Die Wahl fiel aber schließlich auf Perl, Version 5, da seit dieser Version die Scriptsprache objektorientierte Programmierung erlaubt und Perl eine sehr beliebte Sprache unter Systemadministratoren ist. Ein weiterer Punkt, der für Perl sprach, war die Tatsache, dass die Steuerung und Darstellung von Nagios in einem Webinterface erfolgt und dies durch CGI-Scripte realisiert wird. CGI-Scripte werden derzeit in Perl oder PHP entwickelt. Mit dem Hintergedanken, dass es eine Erweiterung innerhalb von Nagios geben soll, ist die CGI-Programmierung unumgänglich. Zudem ist die Implementation von Netdisco in Perl und da der Programmcode eventuell zu verändern ist, liegt es Nahe, die Sprache Perl zu beherrschen.
Die Integration der Topologiedaten in Nagios war als zweite Phase der Arbeit geplant. Die erste Phase verlangte eine graphische Darstellung der Netzwerktopologie in Form eines Diagramms. Zu diesem Zweck existiert das Open-Source Tool GraphViz[1]. Mit Hilfe dessen ist es möglich, Knoten und Kanten darzustellen, die unter Berücksichtigung ihrer Abhängigkeiten bestmöglich und automatisch angeordnet werden. Es wird ein Programm entwickelt, welches die Topologiedaten aus der Datenbank auswertet und unter Verwendung von GraphViz als Graph darstellt. Das komplette Netz von Switches und

[1] http://www.graphviz.org/

Computern besteht aus vielen Einzelverbindungen von Switch zu Switch und von Host zu Switch. Die Beziehungsinformationen sind für den weiteren Verlauf der Arbeit von essentieller Bedeutung, die immer wieder gebraucht werden sollen. Daher wird vor der Grapherzeugung ein *Kommandline-Tool* implementiert, welches die folgenden drei Nachbarschaftsinformationen liefert und als Basis für weitere Implementationen dient. Neben der Verwendung zur Topologiedarstellung, sind die erhaltenen Informationen auch im Arbeitsalltag, zur Fehlersuche, hilfreich.

Elternswitch einer Arbeitsstation Im täglichen Gebrauch ist es oft wichtig zu wissen, an welchen Switch ein Rechner aktuell angeschlossen ist. Daher ist es wichtig die nächste Netzwerkkomponenten, in der Regel der Switch, zu kennen, um zum Beispiel eine Fehlersuche fundiert zu beginnen.

Elternswitch eines Switches Um den Weg von Datenpaketen nachvollziehen zu können, müssen die Verbindungen der Switches zueinander ermittelbar sein. Durch diese Informationsdarstellung kann ein Pfad von Switch über Switch bis zur Wurzel des Netzwerks erzeugt werden.

Alle angeschlossenen Geräte an einem Switch Für eine weitreichendere Übersicht ist es hilfreich zu wissen, welche Nachbargeräte an jedem einzelnen Port des aktuellen Switches angeschlossen sind. Zudem ist die Information relevant, ob an einem Port eventuell Hubs eingebaut sind, sodass mehrere Rechner über einen Switchport angebunden sind. Der unkoordinierte Einsatz von Hubs birgt, wie in Abschnitt 3.6 auf Seite 51 gezeigt, Fehlerquellen, die das Netz erheblich stören können und erlaubt so eine leichtere Identifizierung von möglichen Hubs im Netzwerk.

Der schnelle Zugriff über die Konsolenabfrage und einer simplen Darstellung bietet oftmals schneller Informationen über einen speziellen Sachverhalt, als eine Graphik, die erst nach dem gewünschten Punkt zu überfliegen ist. Alle Routinen, die für diese Anwendung zu schreiben waren, werden von späteren Werkzeugen benötigt. Über diese drei Methoden war die Topologie bereits rudimentär darstellbar. Ein Grundstock an Methoden war somit gegeben und konnten in die Implementation zur Grapherzeugung integriert werden, um einen Gesamtüberblick der Topologie zu bekommen.

Grapherzeugung Bei einer geschätzten Knotenzahl von eintausend Stück für Computer und Switches, wird auf die graphische Darstellung der Computer verzichtet und nur das Netzwerk, bestehend aus den Switches, gezeichnet. Die Verwaltung von verbleibenden 120 Knoten und ihrer Kanten gestaltete sich trotzdem als schwieriges Unterfangen. Die Anordnung der Knoten lässt sich zwar realisieren, aber die zugehörigen Kanten erlauben keine zufriedenstellende Anpassung, die für den menschlichen Eindruck ein optimales Bild ergeben würden. Zur besseren Unterscheidung werden die Switches als Rechtecke, die Accesspoints als Kreise gezeichnet. Zwischen den Objekten markieren Pfeile die Abhängigkeiten. Für einen besseren Gesamtüberblick erlaubt GraphViz die Clusterung von Knoten. So war es möglich die Switches gebäudeweise zusammenzufassen, da diese Information im Locationeintrag gespeichert ist (Abschnitt 4.2). Das Ergebnis ist

ein Bild, welches eine Unterteilung in die verschiedenen Gebäude enthält, die wiederum die Switches und Accesspoints enthalten, die in diesem Gebäude positioniert sind. Da der Campus Koblenz derzeit durch eine reine Sterntopologie, oder genauer gesagt einen balancierten Baum, aufgebaut ist, verteilen sich die Kanten der Kindknoten auf zwei Knoten, den Backboneswitches. Wie bereits in Abschnitt 3.3 beschrieben, verbindet der Stern alle Kinder an einem Elternknoten. Das ist der Grund für die Unübersichtlichkeit bezüglich der vielen Kanten, die in den zwei Backbone-Knoten[2] eingehen. Das entstehende Bild verlangt vom Betrachter ein gewisses Maß an Gutmütigkeit bei der Verfolgung der jeweiligen Kante. Aufgrund der vielen Knoten und ihrer Anordnung entsteht ein Bild, welches selbst bei Verwendung eines Papiers im Standardformat DIN A0 die Grenzen deutlich überschreiten würde. Die Suche nach alternativen Werkzeugen bleibt erfolglos. Ohne Manipulation durch einen Anwender, kann, bei dieser hohen Anzahl von Knoten, kein automatischer, optimaler Graph erzeugt werden. Unter dem Gesichtspunkt, dass sich die Topologie und die zugehörigen Komponenten in ständigem Wandel befinden, wird auf die weitere Betrachtung des Problems verzichtet, da jeder Neuzugang oder Wechsel eines Switches einen neuen Ausdruck der Topologie verlangen würde. Auf die optimale Darstellung auf Papier, unter Berücksichtigung der Einsatzes von Nagios, welches ebenfalls eine visuelle Darstellung anbietet, wird verzichtet.

Berücksichtigung von Topologieänderungen Ein wesentlicher Aspekt bei automatischen Konfigurationen ist der Fall, dass sich einzelne Komponenten der Topologie verändern, in dem sie entfernt oder hinzugefügt werden, und wie diese Veränderung erkannt und berücksichtigt werden soll. Ein Computerprogramm kann nicht entscheiden, ob das Wegfallen eines Switches geplant ist, oder ob das Gerät während des Scanvorgangs einen Systemfehler hat, und aus diesem Grund nicht antwortet. Die automatische Topologieermittlung muss drei Situationen zuverlässig verarbeiten können:

Ein Switch wird der Topologie hinzugefügt In diesem Fall gilt es die folgenden Punkte zu beachten. Wird der Switch als Blatt eingefügt oder zwischen zwei bestehende Switches? Welche Protokolle für Nachbarschaftsinformationen beherrschen die betroffenen Komponenten? Sind alle an der Veränderung beteiligten Geräte CDP-fähig, ist es für Netdisco kein Problem, die betroffenen Komponenten zu berücksichtigen und in der Datenbank die Änderungen mit den neuen Zuständen korrekt zu vermerken. Wird das neue Gerät, welches kein CDP beherrscht, als Blatt eingefügt, bleibt der bisherige Teil der Topologie davon unberührt. Lediglich der Uplinkport des Elternswitches, der neuen Komponente, wird auf dem Port eine große Menge an Geräten registrieren, was für den Administrator ein Zeichen ist, dass ein weiteres Gerät eingebaut ist, aber die Topologieerfassung nicht negativ beeinflußt. Für diesen Fall besteht die Möglichkeit die Informationen des Spanning Trees zu nutzen, um den Kindknoten korrekt in die Topologie zu integrieren. Problematischer wird es, wenn er neue Switch innerhalb eines Pfades eingebaut wird und er keine Unterstützung für CDP hat. Der Switch kann wiederum durch

[2] Das zentrale Element wird auf dem Campus Koblenz durch den Trunk zweier Geräte gebildet, wie in Abschnitt 4.1 erläutert.

die Spanning Tree Daten an den Elternknoten angeschlossen werden, alle Kinder jedoch, sind in Zukunft von der Topologieerkennung von Netdisco ausgeschlossen, auch wenn sie CDP unterstützen. Alle betroffenen Kindknoten müssen in der weiteren Nutzung durch die Spanning Tree Angaben einander bekannt gemacht. Die Knoten werden nicht aus den Tabellen von Netdisco verschwinden, aber die Beziehungseinträge in der Tabelle „device_port“werden beim folgenden Netzwerkscan verschwinden.

Die Daten eines Switches verändern sich Jegliche Veränderungen werden durch regelmäßige SNMP-Abfragen der Switchdaten, auf dem aktuellen Stand gehalten. Veränderungen, egal ob automatisch oder durch einen Benutzereingriff, haben beim kommenden Netzwerkscan eine Veränderung der Datenbankeinträge zur Folge. An der Topologie ändert sich in der Regel nichts. Es darf keinen Unterschied für die Datenbankeinträge haben, ob das Gerät mit CDP- oder Spanning Tree Informationen in die Topologie angebunden wird.

Ein Switch wird aus dem Netzwerk entfernt Die Entnahme eines Switches aus der Topologie stellt für das Nachbargerät keine nennenswerte Bedeutung dar. Bei CDP-fähigen Geräten wird die Veränderung unmittelbar registriert. Sind die Nachbarschaftsinformationen nur durch den Spanning Tree verfügbar, werden auf bekannte Weise die Daten in den Tabellen abgeglichen.
Das Monitoringtool darf diese Veränderung nicht einfach hinnehmen und muss „Alarm schlagen“. Aus diesem Grund ist dieser Aspekt etwas kritisch zu betrachten. Die Veränderung der Topologie ist durch den Administrator geplant. Das Computerprogramm kann das nicht wissen. Demzufolge ist neben der Hardwaremodifikation auch die Software zu bearbeiten, um keine unnötigen Fehlermeldungen zu produzieren.

6.1 Basisbibliothek für die Topologieimplementationen

Die Datenbank dient als zentraler Informationsspeicher für die Netzwerkgeräte und deren Beziehungen, welche von Netdisco auf dem aktuellen Stand gehalten werden. Desweiteren ist die Korrektheit der Informationen von den Fähigkeiten der Netdiscoimplementation abhängig. Wie bereits erwähnt, sind Nagios und MRTG eigenständige Programme, die von Konfigurationsdateien abhängig sind. Für die Automation sind weitere Implementationen notwendig, welche die Daten aus der Datenbank weiterverarbeitet, die Beziehungen der Geräte untereinander ermitteln und diese Informationen, im jeweils verlangten Kontext, zur Verfügung stellen. Es ist zu erwarten, dass unterschiedliche Anwendungen entstehen, die ähnliche Methoden benötigen werden, die mehr oder weniger mit den Topologiedaten arbeiten. Aus diesem Grund wird eine Basis-Library entwickelt, die die wesentlichen, immer wieder verwendeten Methoden für alle Anwendungen bereitstellt.
Da die Programme mit Netzwerkgeräten zu tun haben, macht es Sinn eine Klasse zu entwerfen, die ein grundlegendes, netzwerkfähiges Gerät repräsentiert. Dafür wird eine Klasse *NetComponent.pm* entwickelt, welche die grundlegenden Variablen beinhaltet, die

ein Gerät beschreiben. Von dieser abgeleitet entstehen zwei weitere Klassen, die eine Spezifikation von Switch und Host bezüglich der Ortsangabe beinhalten. Der Grund ist die Art der Informationsgewinnung, da Switches die Informationen durch die SNMP-MIB bekommen und Hosts durch Sekundärinformationen einer Datenbank oder durch ihre Beziehungen zu einem Switch. Zu den Gerätemodulen entsteht zusätzlich ein weiteres Perlmodul, welches Methoden bereitstellt, die in verschiedenen Programmen verwendet werden und somit als Basisbibliothek betrachtet werden kann. Das Klassendiagramm in Abbildung 6.2 auf Seite 95 zeigt die verwendeten Klassen, die als Perlmodule bereitgestellt werden. Eine Methode spielt für alle Implementationen eine besondere Rolle, die Methode *doDeviceHash*. In ihr werden die Daten der Switches verarbeitet und die Beziehungen der Geräte untereinander in Objekten festgehalten. Alle Switches werden als Objekt angelegt und in einem Hash, das durch die Methode zurückgegeben wird, gespeichert. Gegebenenfalls verlangt es ein Programm, dass alle bekannten Hosts verfügbar sind. In diesem Fall gibt es eine weitere Methode, die ein Hash mit Objekten der Hosts erzeugt.

6.1.1 Aufbau der Gerätebeziehungen

Die Methode *doDeviceHash* muss mit einem Parameter aufgerufen werden in Form einer IP. Es sollte die Ip des Switches sein, der die Wurzel des Baumes, dem Backbone des Netzwerks, darstellt. Das erzeugte Objekt, bekommt als Vater eine „Null“zugewiesen, wodurch eine Prüfung auf die Wurzel möglich ist. Es folgt der Aufruf der Methode *go* mit den Parametern IP des aktuellen Gerätes und der IP des zugehörigen Elternknoten. Der Wurzelknoten hat als Elternknoten die "Null". Es folgt die Prüfung, ob die aktuelle IP in einer Liste der bereits betrachteten IPs enthalten ist. Ist dies nicht der Fall, wird sie in diese Liste eingetragen. Das ist notwendig, um zu vermeiden, dass Switches mehrfach untersucht und als Objekt angelegt werden. Der nächste Schritt ermittelt mit Hilfe der Methode *childFinder* alle IP-Adressen der Nachbarswitches, die an den aktuellen Switch angeschlossenen sind. Für jede gefundene IP wird als erstes wieder die Methode *go* mit der neuen IP aufgerufen. Es entsteht eine rekursive Erkundung des Netzwerks. Gelangt die Methode *go* an eine IP, die keine Kindknoten hat, wird das Objekt eines Switches, für die aktuelle IP erzeugt und ein Eintrag in das Hash gemacht. Es ist ein Blatt des Baums ermittelt worden. An diesem Punkt geht die Rekursion eine Ebene zurück und bearbeitet das nachfolgende Element. Als letztes wird das Objekt für die Wurzel erzeugt. Die Rekursion ist beendet und der Hash ist fertig. Der Schlüssel für den Hash sind die IP der Switches. Die eigentliche Objekterzeugung erfolgt in einer Methode *doSwitchObject*, die als Parameter die IP und die IP des Elternknotens benötigt. Die restlichen Angaben, wie Name des Gerätes, werden durch Datenbankaufrufe ermittelt.
Der Aufbau des Host-Hashs verlangt keine Rekursion. In diesem Fall wird die Tabelle *node_ip* verwendet, die eine Liste aller IPs enthält, die an den Switches registriert sind. Der Schlüssel ist wiederum die IP des Gerätes, die restlichen Angaben sind der Switch, an dem die IP zuletzt registriert ist und einige weitere Angaben, die aus der Datenbank entnommen werden.

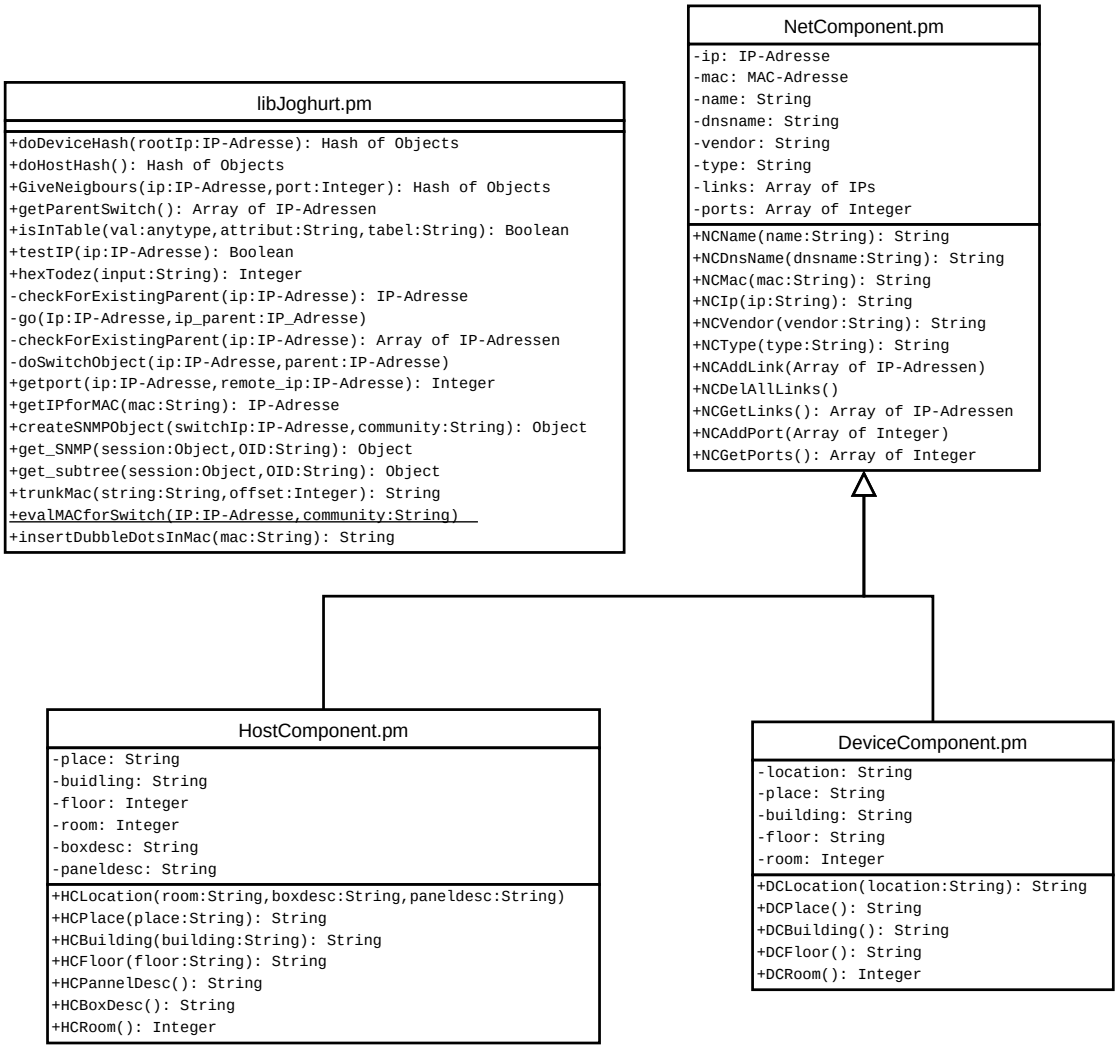

Abbildung 6.2: Klassendiagramm der Implementationsmodule

6.1.2 Datenaufbereitung für die Ortangabe

In der aktuellen Version unterscheiden sich die beiden Objektklassen für die Devices und Hosts lediglich in den Variablen für die Ortsangabe. Wie in Abschnitt 4.2 beschrieben wird die Ortangabe für einen Switch in der *Location* Variable der System-MIB gespeichert. Bei der Erzeugung der Objekts wird der Location-String aus der MIB ausgelesen und mit der Klassenmethode *DCLocation* bearbeitet. Der übergebene String wird geparsed und anhand des Trennzeichens „-“ist es möglich die einzelnen Werte von einander zu trennen und in die vorgesehenen Variablen des Objekts zu speichern.
Die Hostobjekte füllen die Ortsangaben auf zwei alternativen Wegen. Die primäre Weise verwendet die Angaben aus dem Elternswitch des Hosts. Auf diesem Weg ist der Platz, das Gebäude und der Flur bestimmt. Zusätzlich wird geprüft, ob die Tabelle *portmap*, in der die Beziehung der Switchports und der Pannelbeschriftung zu einander in Beziehung gesetzt werden, brauchbare Angaben enthält. Ist in der Tabelle ein verwertbarer Eintrag, hinsichtlich Raumangabe, für den Host, kann anhand dieser Angabe Flur und Raum durch die Tabelle gefüllt werden. Ohne Eintrag bleibt die Raumangabe leer und der Host wird dem Flur des Switches zugeordnet. In einzelnen Fällen kann das zu falschen Angaben führen, wenn mehrere Flure über einen Switch angebunden sind. Dieser Fehler muss derzeit noch akzeptiert werden.

6.2 Netdiscoergänzung - Spanning Tree

Netdisco ist alleine für die Erkundung des Netzwerks verantwortlich. Bislang basiert es komplett auf der CDP-Protokollunterstützung der Switches. Solange das Protokoll durchgängig verfügbar ist, entstehen keine Probleme bei der Erkundung. Bei der Verwendung von Hardware unterschiedlicher Hersteller und zusätzlich noch unterschiedlichen Preiskategorien entstehen Konstellationen, in denen vereinzelte Geräte das CDP-Protokoll nicht unterstützen. Eine Rücksichtnahme auf die Fähigkeiten der Hardware steht nicht zur Diskussion. Es gilt alternative Erkundungsmechanismen zu suchen, die entweder in Netdisco direkt integriert werden, oder als eigenständiges Programm die Datenbanktabelle manipulieren, um auf diese Weise die unbekannten Geräte zu integrieren. Als Alternative ist in Abschnitt 3.5.2 das *LLDP-Protokoll* vorgestellt worden. Da dieses Protokoll derzeit von nur wenigen Herstellern unterstützt wird, kann es für die Problemlösung nicht verwendet werden. Das dritte Protokoll ist der Spanning Tree. Jeder höherwertige Switch ist in der Lage am Spanning Tree teilzunehmen. Der Algorithmus ist für eine effizientere Nutzung schlicht notwendig. Die Daten des Algorithmus sind in den Switches gespeichert und durch SNMP abfragbar. Die Daten aus dem Spanning Tree Algorithmus sind nicht für die Topologieerfassung geplant gewesen und bieten daher lange nicht den Komfort, wie es das CDP oder LLDP tun, stellt aber die einzige Alternative dar zu den beiden Discoverprotokollen dar.

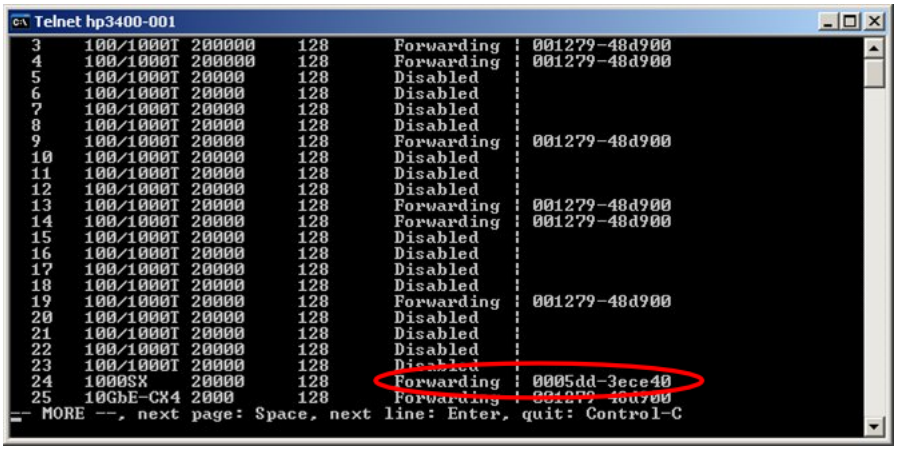

Abbildung 6.3: Ausgabe der STP-Daten des HP Switches HP3400-001 über Telnet

6.2.1 Ergänzung von Netdisco zur Speicherung von MAC-Adressen

Für die Ergänzung von Netdisco zur erfolgreichen Verwendung der Daten aus dem Spanning Tree Algorithmus, sind neben der eigentlichen Ermittlung der Spanning-Tree-Daten, noch weitere Vorbereitungen notwendig. Wie aus Abschnitt 3.6 bekannt, nutzt der Algorithmus nur Informationen auf OSI-Layer2, den MAC-Adressen der Geräte, um den Elternknoten zu identifizieren. In der Tabelle *device* aus Netdisco werden alle Switches mit ihren spezifischen Daten verzeichnet. Es existiert ein Attribut für die MAC-Adresse eines Gerätes. Aus nicht nachvollziehbaren Gründen, wird dieses Attribut bei den meisten Geräten nicht mit Werten versehen. Dieser Fehler muss mit einer Erweiterung korrigiert werden. Unter Verwendung der *Bridge-MIB* ist ein Zugriff auf die Variable *STPbase-BridgeAdress* möglich, die lediglich bei Bridges gesetzt ist und als Inhalt die Identifikation des Gerätes enthält. Der Rückgabewert ist die MAC-Adresse, die in die Tabelle *device* eingetragen wird. Der Code für das Programm ist in Perl geschrieben. Die Einpflegung dessen in den Sourcecode von Netdisco würde bei jedem Update von Netdisco notwendig sein. Daher wird lediglich eine Zeile im Netdisco-Source-Code ergänzt, die das Zusatzprogramm aufruft. Zu beachten ist, dass die Ausführung des Zusatzprogramms, nach jedem Durchlauf von Netdisco erfolgen muss, wenn Netdisco ein Update auf der Tabelle *device* vollzieht, da die MAC-Angaben offensichtlich anders ermittelt werden und bei erfolgloser Ermittlung einer MAC-Adresse, diese mit einem leeren String überschrieben wird.

6.2.2 Nachbarschaftserkundung mit Hilfe des Spanning Tree

In Kapitel 3.5.2 wird die Vorgehensweise zur Nachbarschaftserkundung über die Protokolle CDP und LLDP erklärt. Wie in Abschnitt 5.1 bereits erläutert, besteht ein Problem im Umgang mit Netdisco, sobald die Topologie aus Geräten zusammengesetzt wird, von denen ein Teil aus nicht-CDP-fähigen Geräten besteht. Die Menge dieser Switches nimmt

seit der Verabschiedung des LLDP-Standards zu, da einige Hersteller sich dem Standard anpassen. Desweiteren gibt es Switches, die nur die Funktionalitäten wie Spanning Tree oder VLANs verwenden, aber keine Komfortfunktionen wie CDP. Bei der Erkundung durch den regulären Netdisco-Scan werden diese Geräte nicht gefunden. Die Idee des Netdiscoalgorithmus beruht darauf, dass alle Nachbargeräte eines Switches in der MIB des CDP-Caches eingetragen sind.
Die Idee zur Problemlösung ist, die Erkundung durch Netdisco als Basis durchzuführen und im weiteren Verlauf die gesammelten Daten durch ein ergänzendes Programm zu erweitern. Da ein Großteil der Hardware CDP unterstützt, und die Datenextraktion in der MIB einfach und die Informationen sehr ausführlich sind, ist es sinnvoll die Daten zu verwenden. Auf diese Weise entsteht ein Grundstock des Netzwerks, den es in einem weiteren Schritt nur zu Ergänzen gilt. Die restlichen Geräte werden durch die Spanning Tree Daten der jeweiligen Switches an das bislang erforschte Netzwerk angebunden. Zuerst sind die Switches mit ihrer IP-Adresse ausfindig zu machen. Prinzipiell verlangt es nach einer einfachen Prüfung eines jeden gefundenen Hosts, ob dieser die Bridgingfunktionen untersucht. Die Bridgingfähigkeit ist durch Auslesen der Einträge in der Bridging-MIB entscheidbar, da einfache Hosts die Variablen nicht anbieten. Es sind zwei Wege denkbar, die die Ermittlung der Switches ermöglichen. Der erste nutzt die Ergebnisse aus den Netdiscotabellen. Netdisco bietet die Tabelle *node_ip* an, in der alle, von Netdisco gefundenen Hosts, mit einer IP verzeichnet sind. Durch eine simple Prüfung der IP-Adresse, ob die Bridgingfunktion aktiviert ist, kann entschieden werden, ob einer der bereits gefunden Host ein Switch ist, der bislang noch nicht als solcher deklariert worden ist. Die alternative Vorgehensweise wird auch von vielen kommerziellen Produkten, wie in Abschnitt 2.3 gezeigt, verwendet. Der Administrator gibt vor dem Scan einen IP-Adressbereich an, der seiner Ansicht nach alle verwendeten Switches enthält. So soll sicher gestellt werden, dass jeder Switch gefunden wird, da ein Administrator wissen sollte, in welchem IP-Bereich seine Switches liegen.
Der für die Implementation relevante Eintrag ist der Elternknoten des als Switch entlarvten Hosts. Die Abbildung 6.3 zeigt die Einträge des RZ-Switches *HP3400-001*, der im Serverraum der Universität Koblenz steht. Der Elternknoten zu diesem Switch wird am Port mit der Nummer 24 gefunden. Der Rot markierte Eintrag zeigt eine MAC-Adresse, die unterschiedlich ist zu den übrigen. Jeder Port, der zur Weiterleitung von Datenpaketen aktiviert ist (Forwarding) hat für diesen Port eine MAC-Adresse registriert. Standardweiterleitungsports verwenden die MAC-Adresse des Switches selbst, deswegen sind die vielen gleichen Adressen zu erkennen, der Elternknoten wird mit dessen MAC-Adresse registriert. Diese Information wird sich zu Nutze gemacht, um einen Switch an den Elternknoten anzubinden. Die Einträge der MIB speichern sogar die Portnummer des Elternknotens. Da der Elternknoten nicht durch die MAC-Adresse ansprechbar ist, sind die bisherigen Informationen aus der *device-Tabelle* notwendig, um zu der MAC-Adresse die IP-Adresse zu finden.

6.2.3 Implementation Nachbarschaftserkundung mit dem Spanning Tree

In der aktuellen Implementation wird die erst genannte Vorgehensweise zur Switchfindung verwendet. Es wird die *node_ip-Tabelle* der Reihe nach abgearbeitet und jede enthaltene IP-Adresse als potentieller Switch betrachtet. Die erste Phase überprüft jede IP-Adresse, ob das jeweilige Gerät mittels SNMP ansprechbar ist und es Einträge in der Bridging-MIB gibt. Daraus lässt sich schließen, dass dieses Gerät eine Bridgingfunktion haben muss und daher in die Klasse der Switches gehört. Jedes, als Switch ermittelte Gerät, wird in eine Liste eingetragen, die im Anschluss abgearbeitet wird. Die als Switches entlarvten Geräte werden mit SNMP nach wenigen relevanten Angaben durchsucht, die es erlauben den Switch in die Topologie einzubauen.

- sysName
- BrideMAC
- STPbaseBridgeAdress
- STProotPort
- STPdesignatedRoot
- STPdesignatedRootPort

Die Daten dieser Auflistung in Kombination derer in der bestehenden *device*-Tabelle erlauben es, die Beziehung eines neu entdeckten Gerätes mit bereits registierten Gerät herzustellen. Die neu ermittelten Switches werden nicht in die *device*-Tabelle direkt eingetragen, sondern in eine eigene Tabelle. Leider arbeitet Netdisco bei der Erkundung nur mit den selbst ermittelten Nachbarn unter Miteinbeziehung einer Textdatei (Abbildung 6.4), die Nachbarschaftsinformationen beinhaltet. Das bedeutet, dass Netdisco die Einträge in der aktuellen Datenbank bei der Erkundung nicht berücksichtigt. Bei jedem Erkundungslauf werden die CDP-fähigen Geräte Hop für Hop von einem frei wählbaren Wurzelknoten erkundet, unter Berücksichtigung einer Textdatei, die anderweitige Beziehungen beinhaltet. Angaben, die extern in die Tabelle der Devices eingepflegt werden, werden nicht berücksichtigt und verschwinden in einem, in der Konfigurationsdatei von Netdisco, frei wählbaren Zeitraum. Die Geräte blieben somit in der Tabelle *device* erhalten, aber die Beziehungsangaben in der Tabelle *device_port* werden unmittelbar entfernt. Netdisco registriert, dass es Einträge gibt, die bei dem Scanvorgang durch CDP nicht gefunden worden sind und weist darauf hin, ist aber nicht in der Lage die alten Informationen aus der Datenbank miteinzubeziehen. Zu diesem Zweck hat der Entwickler einen Ausweg integriert. Es ist möglich, mit Angabe der Textdatei (Abbildung 6.4) Beziehungsinformationen extern mitzuteilen. Zeile 1 beschreibt das einzubeziehende Gerät mit der IP-Adresse *141.26.128.165.* gefolgt von Zeile 2 mit der Angabe des verwendeten Links. Zu sehen ist, dass an Port 25 des Gerätes aus Zeile 2 der Nachbar mit der IP *141.26.128.80* an dessen Port *2.4* angeschlossen ist. Die folgenden Zeilen zeigen weitere Geräte, die auf diese Weise eingebunden sind. Um die volle Funktionalität von Netdisco nutzen zu können, ist dies vorerst die einzige Möglichkeit nicht-CDP-fähige Geräte

```
01: 141.26.128.165
02: link:25,141.26.128.80,2.4
03: 141.26.128.117
04: link:GigabitEthernet0/1,141.26.128.80,3.7
05: 141.26.128.161
06: link:49,141.26.128.136,GigabitEthernet0/2
```

Abbildung 6.4: Topologiedatei für nicht-CDP-fähige Netzwerkgeräte

in den Scanvorgang zu integrieren. Zusätzlich muss jedes Gerät, dessen Nachbarschaft durch die Textdatei Netdisco bekannt gegeben wird, mit einem einzelnen Scan für das neue Gerät, in die *device*-Tabelle integriert werden. Diese Einträge sind keine Herausforderung, da es simple SNMP-Abfragen sind und die Ermittlung nur einmalig erfolgt. Im Zuge der notwendigen Erzeugung der Textdatei, wird dieser Integrationsschritt durch einen Netdiscoaufruf erledigt und verlangt so keinen weiteren Programmcode.
Das Ergänzungsscript zu Netdisco, *netdisco_stp.pl*, unterscheidet drei Parameter. Der erste war notwendig, um die fehlenden MAC-Adressen in der Tabelle *device* zu ergänzen. Ein Fehler, den es zu kompensieren galt. Der zweite Parameter dient der Ermittlung der Hosts, die eine Bridgingfunktion haben auf Basis der Tabelle *node_ip*. Aus Performanzgründen wird derzeit nur die Netzadressen 141.26.128.x betrachtet, da alle verwendeten Switches des Campus Koblenz in einem VLAN mit diesem Adressbereich konfiguriert sind. Die folgende Liste zeigt die notwendigen Variablen, die der Programmcode verarbeitet.

- Name
- Ortsangabe
- MAC-Adresse zur Identifikation
- Designated Root
- Rootport
- Remote MAC
- Remote Port

Die ersten fünf Angaben sind durch einfache SNMP-GET Anfragen zu ermitteln. Die beiden restlichen, aber bedeutenden Werte, verlangen einige Umwege. Um die MAC-Adresse des Elternknotens zu bekommen, wird sich der MIB-Variablen *STPPortDesignatedBridge* mit der *OID = .1.3.6.1.2.1.17.2.15.1.8* bedient. Diese enthält eine Tabelle, die als Schlüssel die interne Identifikation jedes Ports des Switches verwendet und als Wert eine MAC-Adresse liefert. Für den Port, der den Uplink zu einem Elternknoten darstellt, ist es die Adresse des gesuchten Elternknotens. Die restlichen Werte unterscheiden sich je nach Hersteller und Gerätegeneration. Einmal ist es die eigene Adresse der

Bridge, mal gar nichts und wiederum mal ist es eine MAC bestehend aus Nullen. Anhand dieses Wissens, ist die Adresse des Elternknotens identifizierbar. Die zugehörige Portnummer stellt eine kleine Herausforderung dar. Spanning Tree verwaltet nur Elterninformationen, daher ist der Elternknoten keine Hilfe bzgl. seines Kindes. Die MIB-Variable *STPPortDesignatedPort* mit der *OID = .1.3.6.1.2.1.17.2.15.1.9* liefert zu der internen Identifikation eine Portnummer in Hexadezimalschreibweise. Nach deren Umrechnung in das Dezimalsystem wird diese als die interne Port-ID des Elternknotens entlarvt. Nach Ermittlung der IP-Adresse des Elternknotens wird eine neue SNMP-Session erzeugt und durch Verknüpfung der MIB-Variablen *BasePortIfIndex*, welche die gerade ermittelte interne Port-ID nutzt, und *ifName*, die den Portnamen bereitstellt, ist die benötigte Portnummer des Elternknotens bekannt.
An dieser Stelle gilt es noch eine weitere Hürde zu nehmen, da CDP nicht die Portnamen des jeweiligen Switches verwendet, sondern aussagekräftige Namen wie *FastEthernet0/3*, anstatt Fa0/3, wie es die Standard-MIB bereitstellt. Diese Berücksichtigung ist notwendig, da Netdisco die Datenbank nach den CDP-Angaben füllt. Die Geräte, die auf diese Weise in Netdisco integriert werden, haben keine Einträge im CDP-Cache und kennen diese Art der Portbezeichnung nicht. Daher sind die Angaben der Interface-MIB-Variablen *ifName* auf diese Darstellung zu übersetzen, da sonst keine Verbindung zu den bisherigen Geräten hergestellt werden kann.
Die Daten der Geräte, die über Spanning Tree integriert werden, stehen in einer extra Tabelle. Um diese nun in eine für Netdisco verwertbares Format zu bringen, existiert eine weitere Option, die lediglich die Textdatei, mit weiteren Topologieinformationen für Netdisco, erzeugt. Dies muss separat geschehen. Jedes Gerät, welches durch Spanning Tree eingebunden wird, muss bei jedem Netdiscoscan in der Topologiedatei stehen, da es sonst als nicht erreichbar deklariert wird. Daher ist es notwendig, dass eine eigene Tabelle sowohl die früh gefundenen Geräte, also auch die in späteren Erkundungen gefunden Geräte sammelt, die zum Beispiel als Kind an einem nicht-CDP-fähigen Switch angeschlossen sind.

6.2.4 Ermittlung von MAC-IP Adresspaaren

Eine wichtige Frage gilt es noch zu klären. Wie bekommt das Programm zu einer MAC-Adresse die zugehörige IP-Adresse. Switches nutzen bei der Datenweiterleitung, nur Adressen auf Ebene *Layer2*, den MAC-Adressen, und kommen daher mit der IP-Adresse der Kommunikationspartner nie in Berührung. Aus Abschnitt 3.5.2 ist bekannt, dass CDP als Protokoll zwischen Switches funktioniert und im Datenteil der Pakete die IP des Gerätes übermittelt. Durch diesen Komfort, den Cisco bereit gestellt hat, war es möglich die IP-Adresse des Nachbarswitches in jedem Switch zu speichern. Im Gegensatz dazu, beinhalten die Datenpakete des Spanning Trees (vgl. Abschnitt 3.6) keine Informationen hinsichtlich der IP-Adressen, da dies kein Mechanismus ist, der dem Administrator einen Komfort bieten soll, sondern einen funktionalen Nutzen, den der Schleifenvermeidung, hat. Der Spanning Tree braucht für seinen Dienst keine IP-Adressen, sondern nur die MAC-Adresse. Jedoch bieten die MAC-Adressen des Elternknotens, eines jeden Switches, eine Möglichkeit Zusammenhänge darzustellen.

Die Lösung ist der ARP-Cache der Rechner (vgl. Abbschnitt 3.4). Dieser Cache bietet eine Tabelle an, die eine MAC-Adresse auf die zugehörige IP-Adresse abbildet. Die IP-Adresse eines Switches steht im ARP-Cache desjenigen Hosts, der zum Beispiel mittels Telnet[3] kommuniziert hat. Es existieren also eine Reihe von Hosts, die die IP-Adressen der Switches kennen. Mit der derzeitigen Funktionalität von Netdisco wird lediglich der ARP-Cache von Switches ausgelesen. Das hat zur Folge, dass lediglich die Hosts mit ihrer IP in der Datenbank eingetragen werden, die mit einem Switch kommuniziert haben. Aber selbst der Cache, dieser wenigen gefundenen Arbeitsstationen, wird von Netdisco nicht näher untersucht, was zu deutlich mehr IP-MAC Paaren führen würde. Da Switches selbst untereinander nicht auf IP-Ebene kommunizieren, auch nicht durch das CDP-Protokoll, sind die IP-Adressen der Switches nicht in den ARP-Caches der Switches enthalten.
Das Problem, dass zu den MAC-Adressen der durch Spanning Tree gefundenen Switches, die IP-Adressen gesucht werden müssen, ist abhängig von der Netzwerkerkundung durch den Netdiscoalgorithmus. Die alternative Vorgehensweise, einen IP-Adressbereich anzugeben, hat dieses Problem nicht. Bei der Untersuchung jeder IP des Adressbereichs wird ein Switch durch die Bridging-MIB identifiziert und gibt unmittelbar für das Gerät die MAC- und IP-Adresse bekannt. Die Beziehungen können im weiteren Verlauf direkt hergestellt werden. Diese Alternative hat den Nachteil, dass alle IP-Adressen zu prüfen sind und dies ggf. deutlich länger dauert, erspart dafür die aufwendige Suche der IP-Adressen.

6.2.5 VLANs als Lösung zur Entdeckung unbekannter Switches

Beim Einsatz von Netdisco, in der Campustopologie, werden sehr viele Hosts gefunden, die in den Tabellen *node* und *node_ip* erscheinen. Das liegt daran, dass der verwendete Router ein CDP-fähiges Gerät ist und in der Erkundung durch Netdisco miteinbezogen wird. Aus diesem Gerät wird der Großteil der IP-Adressen für eine MAC-Adresse ermittelt. Das liegt daran, dass ein Router auf Layer3-Ebene arbeitet und die Pakete nach der IP untersucht, um das Routing in das Internet zu ermöglichen. Jeder Rechner, der einmal Pakete über den Router versendet hat, wird durch Netdisco ermittelt. Das erklärt nicht, warum auch jeder Switch in der Datenbank erscheint. Der Switch selbst sendet keine eigenen Pakete über den Router ins Internet, da er die Pakete von Hosts auf Layer2 verarbeitet und keinen eigenen Kontakt mit dem Internet aufnimmt. Die Lösung ist im logischen Aufbau der Campustopologie zu finden. Switches befinden sich zur Verwaltung in einem VLAN. Bei der Kommunikation eines Hosts, der nicht in diesem VLAN ist, mit einem Switch ist es notwendig, dass ein Routing zwischen den beiden VLANs erfolgt. Auf diese Weise wird die IP jedes Switches dem Router bekannt. Würden keine VLANs eingesetzt werden, und es wäre eine direkte Kommunikation von Host zu Switch möglich, würden die IPs der Switches nicht im Router eingetragen und Netdisco könnte die IP-Adressen nicht ermitteln.

[3] Telnet ist ein Programm, welches eine Verbindung zu einem Host ermöglicht und ein Terminalfunktion bietet. So ist es möglich von einem Fremdrechner ein anderes Gerät über ein Netzwerk zu bedienen.

6.3 Automatische Objekterzeugung

Die Vorgehensweise bei der automatischen Objekterzeugung könnte als noch nicht ausgereift bezeichnet werden, bietet aber einen ersten lauffähigen Ansatz. Es war ein geeigneter Weg zu finden, um die Gerätedaten aus der Datenbank zu verarbeiten und die Ergebnisse den Tools Nagios und MRTG bereit zu stellen. Ein Grund für die Wahl der beiden Werkzeuge war die Möglichkeit, dass die Konfigurationen als reine Textdateien vorliegen. So ist es einfach, die Informationen in Textdateien zu schreiben, die von den Tools eingelesen werden. Es stellt sich nun die Frage, wann und wie die Konfigurationsdateien erzeugt werden sollen. Für MRTG war die Lösung schnell gefunden. Ein Script prüft in einem festen Zeitabstand, ob jeder Eintrag der Datenbanktabelle *device* eine passende Konfigurationsdatei im MRTG-Ordner hat. Die Konfigurationsdatei wird mit der IP-Adresse des Gerätes betitelt. Auf diese Weise ist ein Vergleich leicht zu realisieren. Es ist nicht zu erwarten, dass sich die Konfigurationen für MRTG eines Switches ändern. Daher reicht die Prüfung auf Existenz.
Die Konfigurationserzeugung für Nagios gestaltet sich deutlich komplizierter. Vor allem ist zu berücksichtigen, dass Konfigurationen für Nagios nicht automatisch gelöscht werden dürfen. Würde ein Gerät ausfallen und ein Automatismus würde erkenne, dass das Gerät nicht mehr existiert, besteht die Gefahr, dass die Konfiguration einfach gelöscht wird. Dementsprechend würde auch die Überwachung aus Nagios verschwinden und die Administratoren werden auf den Ausfall nicht aufmerksam gemacht. Daher habe ich mich dazu entschieden, Konfigurationen für Nagios nicht automatisch zu löschen, sondern jede Konfiguration zu überschreiben, falls sie existiert. So werden Änderungen, zum Beispiel ein neuer Elternknoten, berücksichtigt, aber kein Gerät geht verloren. Der Vorgang wird durch ein Shellscript gesteuert, welches alle fünfzehn Minuten durch einen Cronjob angestoßen wird. Das erste der beiden im Shellscript enthaltenen Perlscripte ist dafür zuständig die IP-Tabelle der Datenbank abzusuchen, um für jede enthaltene IP den derzeitigen Rechnernamen mittels *nslookup* zu ermitteln. Dieser Vorgang ist notwendig, um die Datenbankeinträge der Hostnamen auf dem aktuellen Stand zu halten. Prinzipiell ist dieser Scan in einer Umgebung mit festzugeordneten IP-Adressen nur in den ersten Tagen des Einsatzes der Monitoringumgebung notwendig. Danach sollten alle Rechnernamen bekannt sein. Unter dem Aspekt, dass es sich um eine dynamische Umgebung handelt, wird erwartet, dass regelmäßig neue Rechner hinzukommen, die es zu erfassen gilt. Wird das Script seltener aufgerufen, ist darauf zu achten, dass es während der regulären Arbeitszeiten erfolgt, da zum Beispiel Nachts viele Computer ausgeschaltet sind.
Das zweite Script *doNagiosObjects.pl* ist der Kern des Automatismus. Das Programm verlangt als Parameter den Root-Switch in Form der IP oder des Namens. Nach einer Prüfung, ob dieser in der Tabelle der Switches existiert, erfolgt die Erzeugung der beiden Hashes für die Switches und die Hosts. Bei der Objekterzeugung werden alle Switches und Hosts berücksichtigt, die in den Datenbanktabellen existieren. Es folgen zwei Methodenaufrufe, die einmal die Erzeugung der Deviceobjekte und einmal die der Hostobjekte anstoßen. Die Daten und Beziehungen der Geräte untereinander liegen bereits in dem Hash vor, welches durch die Methoden des Basisbibliothek erzeugt wird. Es sind keine Berechnungen über Nachbarschaften mehr notwendig. Der Kern des Scripts ist prinzipi-

ell nur der Aufruf zur Hasherzeugung und die sequenzielle Erzeugung von Textdateien, den Konfigurationen der Hosts und Switches, in vorgegebene Order. Nagios erlaubt die Verarbeitung von Textdateien, wodurch eine einfache Erzeugung durch ein Perlscript möglich ist. Die wesentliche Aufgabe des Hostobjektes besteht darin, den Namen und die IP-Adresse bereitzustellen und durch den Parameter *parents* Beziehungen zu anderen Objekten herzustellen. Desweiteren beinhaltet die Objektdefinition die Nagiosparameter wann und für welchen Zustand Benachrichtigungen erfolgen sollen. Ein letzter wichtiger Parameter beinhaltet die Kontaktgruppe, über die die Zuständigkeiten anhand der Kontaktpersonen realisiert sind. In der aktuellen Version, zur Erzeugung der Konfigurationen, werden die alten Dateien überschrieben. Dadurch gehen keine Konfigurationen verloren, die physisch nicht mehr gefunden worden sind. Dadurch, dass die alte Datei noch existiert und weiterhin auch in Nagios, wird Nagios einen Alarm ausgeben, dass diese Komponente nicht mehr am Netzwerk angeschlossen ist. Für eine beabsichtigte Entfernung eines Gerätes sind sowohl die Datenbankeinträge, als auch die Konfigurationsdatei für Nagios durch den Administrator zu löschen.

6.4 Optisches Warnsystem Campus Koblenz

Ein großer Kritikpunkt von mir an Nagios ist, dass auf der Startseite keine Informationen über Problemzustände angezeigt werden. Erst durch Verwendung der Menüeinträge links gelangt der Administrator zu ersten Informationen. Bei diesem Menü muss er sich zusätzlich noch entscheiden, welche Darstellung er bevorzugt, ohne eine Ahnung von einem Problem zu haben, ob es sich um einen Service oder einen Host handelt. Aus diesem Grund wird eine Webseite entwickelt, die den Startbildschirm in Nagios ersetzt und ohne eine Aktion des Benutzers den Gesamtstatus des Netzwerks darstellt. Die Informationen über die Lizenzbestimmungen sind im Anwendungsbetrieb weniger interessant und werden weggelassen. Die Entwicklung einer graphischen Erweiterung für Nagios stellt eine interessante Herausforderung dar. Die Vorgabe der Arbeit verlangt ein hierarchisches System, bei dem die oberste Ebene durch ein photorealistisches Luftbild des Campus Koblenz präsentiert werden soll. Auf diesem Bild soll ein optischer Vermerk für problembehaftete Geräte erzeugt werden. Für den Anwender soll diese Idee ein intuitiver Leitfaden bei der Fehler- und Problemlösung sein. Durch Anwahl des jeweiligen Gebäudes wird die nächste Ebene dargestellt, das Innere des Gebäudes.
Für die Realisierung sind zwei Teilkomponenten notwendig. Der erste Teil nutzt das Konzept der frei definierbaren Kommandos in Nagios, um die Probleminformationen, die Nagios mitteilt, der Erweiterung bekannt zu machen. Die Idee ist die, das Benachrichtigungssystem von Nagios zu nutzen, um eine Datenbanktabelle mit Informationen zu füllen, welches Gerät mit einem Problem behaftet ist (vgl. Abbildung 5.1). Alternativ könnte die temporäre Datei von Nagios geparsed werden, um an die gleichen Informationen heranzukommen. Das Parsen von Textdateien ist immer mit Sonderfällen und Problemen behaftet und wird nicht weiter verfolgt, da die Alternative dazu viel elegan-

Abbildung 6.5: Darstellung der Switches, mit Zustand, im A-Gebäude des Campus Koblenz.

ter erscheint. Im ersten Schritt wird eine Tabelle *imap_data*[4] in Postgres eingefügt, sie speichert die Daten der problembehafteten Geräte. Der zweite Schritt verlangt die Realisierung eines Programms, welches die Parameter IP, Location und Status übergeben bekommt, und die Angaben in die Datenbank schreibt. Der Aufruf des Programms erfolgt durch ein neu zu definierendes Nagioskommando (vgl. Abbildung 5.8). Nagios erlaubt es, bei einem Event nicht nur eine Kontaktgruppe zu benachrichtigen, sondern mehrere. Diese Idee wird sich zu Nutze gemacht. Bei einer Benachrichtigung für einen Host oder Service wird in Zukunft nicht nur eine EMail verschickt, sondern im gleichen Zug ein Eintrag in die Datenbank gemacht. Das Nagioskommando *contact_groups* definiert welche Kontaktgruppe für den aktuellen Host- oder Servicecheck zu benachrichtigen ist. Auf diese Weise ist eine Differenzierung möglich, für welche Geräte der Datenbankeintrag erfolgen soll. Für diesen Fall wird nicht nur die Kontaktgruppe mit den Administratoren in der Konfigurationsdatei angegeben, sondern eine weitere Gruppe, die nur eine virtuelle Kontaktperson hat. Die virtuelle Kontaktperson erhält keine EMail, sondern ruft das

[4]Die Bezeichnung soll zum Ausdruck geben, dass es sich um eine Image-Map handelt, die mit Daten aus der Datenbank erzeugt wird.

Programm auf, welches die Daten der eigentlichen Nachricht entsprechend in die Datenbank schreibt. Erfolgt eine Nachricht, dass der Zustand wieder im grünen Bereich liegt, ist es möglich durch den Zustand *OK* die Daten aus der Tabelle wieder zu entfernen.
Die Visualisierung der Datenbankeinträge wird durch eine Webseite realisiert, die durch ein CGI-Script erzeugt wird. Das Script liest die Daten aus der Tabelle aus und kann daraus ermitteln wie viele Rechner pro Gebäude ausgeschaltet sind, oder ob ein Service den Zustand *WARNING* oder *CRITICAL* hat. Anhand dieser Angaben wird unter Verwendung des *GD-Toolkits*[5] ein Photo des Campus mit farblich unterschiedenen Ausrufezeichen ergänzt, das zusätzlich die Anzahl betroffener Geräte pro Gebäude enthält. Die Abbildung 5.1 zeigt ein Beispiel mit zwei Probleme im Gebäude rechts unten. Durch Definition einer *Image Map* werden Hyperlinks auf dem Bild erzeugt, welche eine Navigation in die tieferen Ebenen erlauben.
Nach der Wahl für ein Gebäude wird auf die photorealistische Darstellung verzichtet. Unter Verwendung simpler HTML-Tabellen werden die Geschosse dargestellt, in deren Zellen die Switches aufgeführt sind. Die Darstellung beschränkt sich auf Switches und Accesspoints, die farblich durch ihren aktuellen Zustand unterschieden werden. Die Abbildung 6.5 zeigt das gerade erläuterte Beispiel in differenzierter Darstellung, in der die Probleme auf den Etagen unterschieden werden. Da Nagios für den Zustand *Down* und *Critcal* beidesmal die Farbe Rot verwendet, wir in für diese Implementation der Zustand *Critcal* orange markiert, zur besseren Unterscheidung. Grün gilt sowohl für den Hoststatus, also auch für den Servicestatus.

Grün Das Gerät ist an und alle Services laufen zufriedenstellend.

Gelb Ein Service meldet den Zustand *WARNING*.

Orange Ein Service meldet einen *CRITICAL*-Zustand.

Rot Das Gerät ist nicht erreichbar.

Diese Farbcodierung wird sowohl in der photorealistischen Darstellung des Campus, als auch in der Darstellung der einzelnen Etagen verwendet. Die optische Darstellung erweist sich in der Praxis als schnelle Navigationshilfe. Um eine weitere Optimierung zu erlangen wird die EMailbenachrichtigung modifiziert. Bislang bestand die EMail lediglich aus den sachlichen Angaben, welches Gerät welches Problem hat. Es wird ein Hyperlink in die EMail integriert, die auf die photorealistische Darstellung verweist. Für einen Servicefehler, wird eine extra Nachricht versendet, die einen weiteren Link zur Detailansicht in Nagios beinhaltet. Das reduziert die Menge der notwendigen Mausklicks und beschleunigt die Fehlersuche, da der Anwender von der EMail direkt zu dem problembehafteten Gerät innerhalb von Nagios geführt wird.
Eine weitere Ergänzung ist die Differenzierung der jeweiligen Flure. Der Flur ist im Regelfall mit einem Switch ausgestattet, die Ausnahmen sind zwei oder drei Geräte. Die weitere Unterteilung bietet keine neue Information. Interessant wird es bei der Miteinbeziehung der einzelnen Arbeitsplätze jedes Flurs. Auf diese Weise erhält man einen

[5] GD Graphics Library, http://www.boutell.com/gd/

visuellen Eindruck darüber, welche Rechner auf dieser Etage angeschlossen sind und zum gegebenen Zeitpunkt angeschaltet sind. Eine Überwachung der Rechner mit einem Dienst ist nicht von Bedeutung, da ein Arbeitsplatzrechner keine Dienste anbietet. Relevant wären vereinzelte Server, die unter Umständen auf den Fluren untergebracht sind. Da für den derzeitigen Einsatzzweck keine bedeutenden Informationen hinzukommen, wird auf die Darstellung der Flure verzichtet.

6.5 Integration von Performancedaten in Nagios

Für die detailliertere Überwachung der Switches hinsichtlich ihrer Auslastung wird die Anforderung gestellt, für jeden Port den Datenverkehr zu messen. Dies unterstützt die Administratoren bei Entscheidungen, hinsichtlich der Auslastung der Switches. Realisiert wird dies durch Abfrage der Daten mittels SNMP aus jedem Switch. Um den Implementationsaufwand geringer zu halten, wird auf das Werkzeug MRTG[6] zurückgegriffen. Mit Hilfe eines einfachen Perl-Scripts wird die Tabelle *device* mit den Vorhanden MRTG-Konfigurationen verglichen. Existiert ein Eintrag, der noch keine MRTG-Konfiguration besitzt, werden die notwendigen Dateien für MRTG erzeugt. MRTG selbst wird durch einen Cronjob alle fünf Minuten aufgerufen und arbeitet alle verfügbaren Konfigurationen für die Switches ab. Die Kunst besteht darin, neue und ausscheidende Geräte ohne Benutzerinteraktion zu berücksichtigen und die Konfiguration zu erzeugen bzw. zu verwerfen. Die Erzeugung der Konfiguration für ein einzelnes Gerät hat keine langwierige Prozedur zur Folge, widerspricht aber dem geforderten automatisierten Prozess. Betrachtet man die vielen einzelnen Schritte, die die entwickelte Monitoringumgebung verlangt, nimmt der Aufwand pro Gerät einige Zeit in Anspruch. Daher ist die Automatisierung durchaus notwendig.
Realisiert wird die Automation durch regelmäßiges Ansteuern eines Perlprogramms. Die Aufgabe besteht darin, die Tabelle der Switches mit dem Ordner der MRTG-Konfigurationen zu prüfen. Existiert ein Eintrag, der noch keine MRTG-Konfiguration besitzt, wird diese angelegt. Die Konfiguration wird erzeugt und steht beim kommenden Start von MRTG zur Verfügung und wird abgearbeitet. Bei der ersten Verwendung einer neuen Konfiguration entsteht ein Ordner für die Daten des neuen Gerätes, in dem sowohl alle ermittelten Daten, als auch die zugehörigen Grafiken und HTML-Dateien abgelegt werden. In jedem weiteren Aufruf werden die Dateien aktualisiert. Es ist sinnvoll die Daten eines jeden Gerätes in einem separaten Ordner unterzubringen, um die Übersichtlichkeit innerhalb des Dateibaums zu erhalten.
Die Performancedaten sind bereitgestellt und zur Ansicht in einer Webseite verfügbar. Abbildung 6.6 zeigt ein Beispiel für einen Switch des Campus Koblenz. Die Einbindung der Anzeigeoption in Nagios wird durch eine Anpassung der Objektdefinition erreicht. Nagios bietet zu den herkömmlichen Konfigurationen eines Gerätes die Definition von *Erweiterten Host-Informationen*. Neben der Möglichkeit auf spezielle Icons für das Gerät zu verweisen, existieren zwei Parameter für die Angabe von URLs.

[6] http://oss.oetiker.ch/mrtg/

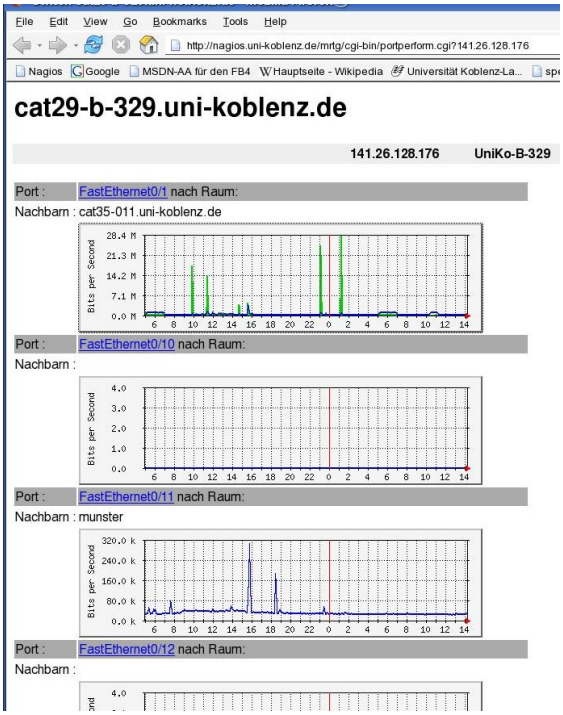

Abbildung 6.6: Graphische Darstellung des Netzwerkverkehrs pro Port. inkl. Angabe der angeschlossenen Geräte.

notes_url Erlaubt die Angabe einer beliebigen Website und wird, falls definiert, mit einem kleinen Notizblock bei der Darstellung zur Geltung gebracht.

action_url Dient dazu auf ein Script zu verweisen, welches eine für den Host relevante Aktion ausführt.

Prinzipiell sind beide Angaben mit einem beliebigen Verweis nutzbar, einem simplen Link, auch wenn es ursprünglich vom Nagios-Entwickler anders vorgesehen ist. Daher ist es durchaus möglich beide Parameter mit einem Script zu verbinden oder mit einfachen Websites. Für die Erweiterung der Performanzdaten wird der Parameter *action_url* verwendet. Dieser verweist auf ein CGI-Script, welches alle Ports des betroffenen Gerätes auflistet und die dazu gehörigen Nachbarn und die Grafik des aktuellen Verkehrs darstellt. Auf diese Weise ist eine ansprechende Präsentation des Netzwerkverkehrs der angeschlossenen Nachbarn erzielt.
Im optimalen Fall ist an den Ports nur ein Gerät angeschlossen und kein Hub dazwischen gesteckt. Bei einzelnen Geräten kann der echte Verkehr für den einen Client analysiert werden. Bei mehreren Geräten ist die Angabe wenig aussagekräftig. Es ist daher zu Empfehlen auf die Verwendung von Hubs und Layer2-Switches zu verzichten.

6.6 Erfahrungen mit dem entwickelten System

Nach der erfolgreichen Erkundung der Topologie und der angeschlossenen Hosts erfolgte unmittelbar die Entwicklung zur automatischen Erzeugung der Nagioskonfigurationsdateien. Ab diesem Punkt war Nagios in der Lage erste Überwachungen durchzuführen. Für jedes Hostobjekt wird ein Standardservice angelegt, ein simpler PING[7]. Durch die regelmäßigen Überprüfungen war es somit möglich den Ausfall eines Rechners zu melden. Zu diesem Zeitpunkt existierte bereits das Kommandozeilentool, welches die Nachbarschaften eines Switches darstellen konnte, bzw. die Elternknoten eines Hosts oder Switches. Der Einsatz der Umgebung, während der Entwicklung, hat bereits in einigen Situationen wertvolle Dienste geleistet.
Im Rechnerraum C207 meldeten drei Rechner, dass sie nur einen eingeschränkten Zugriff auf das Netzwerk haben. Die Rechner selbst waren erreichbar, aber eine Anmeldung über das Netzwerk nicht möglich. Erste Überprüfungen an den Rechnern konnten die Situation nicht erklären. Unter Verwendung des Kommandozeilentools stellt sich heraus, das genau diese drei Rechner am gleichen Port des Etagenswitches angeschlossen waren. Das führte zu der Überlegung, dass zwischen den drei Rechnern ein Hub eingebaut sein muß. Im weiteren Verlauf stellt sich heraus, dass zum Zeitpunkt der Inbetriebnahme der drei Rechner nicht genug Ports am Switch zur Verfügung standen und im Kabelkanal des Tisches der Hub installiert wurde. Der Hub war schlicht kaputt gegangen und die Rechner nach Anschluss, an je einen eigenen Port des Etagenswitches, wieder funktionsfähig.
Der nächste Nutzen betraf die Meldung aus Nagios, dass der Switch HP2650-G-333, der die Institute Chemie und Physik anbindet, eine Warnmeldung anzeigte. Ein beträchtlich hoher Anteil an duplizierten Paketen wurden für das Gerät registriert. Diese Information hat es zu diesem Zeitpunkt noch nie gegeben, da diese Information bislang noch nie den Switch verlassen hatte. Es war bekannt, dass des öfteren Meldungen aus diesen beiden Instituten im Rechenzentrums eingingen, die das Netzwerk als unerträglich langsam beschrieben. Da sich in früheren Zeiten keine Lösung finden lies, wurde der damalige Switch gegen einen neueren ausgetauscht, was offensichtlich keine nennenswerte Verbesserungen erreichte. Die Information der duplizierten Pakete lässt, wie aus Abschnitt 3.6 bekannt, auf den Einsatz von L1/L2-Geräte schließen. Dies war auch der Fall und erlaubte somit eine fundierte Antwort auf die Meldung hinsichtlich der ungünstigen Performanz des Netzwerks.
Das Rechenzentrum bindet seit kurzer Zeit das neue Studentenwohnheim über eine Richtfunkstrecke an das lokale Netzwerk an. Die Verwaltung des Wohnheimnetzwerks obliegt eigenen Mitarbeitern. Das Rechenzentrum ist lediglich für die Bereitstellung der Funkstrecke zuständig. Als die Internetverbindung des Wohnheims eines Tages nicht funktionierte und der örtliche Mitarbeiter nicht zu Gegen war, wurde der Ausfall der Funkstrecke vermutet. Nach einer gezielten Betrachtung der Daten aus Nagios war zu erkennen, dass beide Antennen in Bereitschaft waren, aber der Router des Wohnheims

[7] Ein Ping ist ein Computerprogramm, welches die Erreichbarkeit eines Rechners von einem Computer aus über ein IP-Netzwerk mittels des ICMP-Requests testet und zugleich die Antwortzeit liefert.

nicht antwortete. Für diesen Fall war eine schnelle und begründete Aussage für die Betroffenen möglich gewesen.
Im Arbeitsalltag unterstützt Nagios derzeit die Administratoren der Poolraumdrucker. Die Drucker werden sehr viel benutzt und verbrauchen dementsprechend viel Papier und Toner. Um einen reibungslosen Ablauf zu ermöglichen, prüft Nagios mit Hilfe eines Plugins zur Druckerprüfung, ob Papierstaus vorliegen, der Toner zu neige geht, oder sonstige Probleme vorliegen, die über SNMP ermittelbar sind. Mit Hilfe des Mechanismus werden die zuständigen Mitarbeiter unmittelbar benachrichtigt und gewährleistet für die Studierenden ein reibungsloses drucken. Natürlich wird Nagios für alle Switches und Server zur Überwachung verwendet. Da die Verfügbarkeit der Server und Switches deutlich über 99% liegt, sind für diese Geräte noch kaum Erfahrungswerte gesammelt worden.

6.7 Gedanken zur Weiterentwicklung

Der Einsatz der Umgebung, bereits während der Entwicklung, verlangt trotz vorheriger Überlegungen immer wieder Modifikationen die für einen brauchbaren Einsatz des Systems unabdingbar sind. Bei der Entwicklung des Gesamtsystems entstehen ununterbrochen Ideen zur Weiterentwicklung. Teilweise sind diese von so enormer Wichtigkeit, dass sie unmittelbar umgesetzt werden müssen. Andererseits entstehen Ideen, die eine grundlegende Umgestaltung verlangen, mit einem erhöhten Zeit- und Planungsaufwand. Besonders der aktive Einsatz in Problemsituationen verdeutlicht, in welche Richtungen eine Weiterentwicklung notwendig ist, um das System leistungsfähiger zumachen. So fundiert ein System in der Planungsphase durchdacht wird, so offenbart oftmals erst der praktische Einsatz Situationen, die durch Systemplaner, die nicht im täglichen Umgang mit der Materie sind, einfach nicht vorhersehbar sind. Teilweise zeigt erst der reale Einsatz, welche Vorgehensweise und Steuerungsmechanismen die besten Ergebnisse bringen. Die Erfahrungen mit dem System gilt es festzuhalten, um eine praxisorientierte Weiterentwicklung zu ermöglichen, wodurch das System in der Zukunft leistungsstärker wird.

6.7.1 Erkennung von Hubs und Layer2-Switches

Bereits im frühen Stadium der Arbeit stellt sich die Frage, ob einfache Layer1- und Layer2-Switches lokalisierbar sind. Diese Geräte sind nicht durch eine MAC- oder IP-Adresse identifizierbar. Es besteht zu dem keine Möglichkeit irgendwelche Informationen aus den Geräten zu extrahieren. Eine sinnvolle Aussage, welche Hosts an einem solchen Gerät angeschlossen sind, ist nicht so einfach möglich. Unter Miteinbeziehung der managbaren Switches sind zwei Angaben möglich.

- An dem Port eines managebaren Switches sind mehrere aktive Host angeschlossen. Folglich muss an diesem Port ein Hub angeschlossen sein.
- Der Port arbeitet in Half- oder Fulldublex Modus. So ist es wahrscheinlich, dass es sich um einen Hub, oder einen Layer2-Switch handelt.

Es ist nicht entscheidbar, ob es sich um einen großen 24-Port HUB/L2-Switch oder um mehrere Fünf- oder Achtportgeräte handelt. Eine Idee erachtet die Möglichkeit an die jeweiligen Geräte das gleiche Datenpaket zu senden, und die Antwortzeiten zu messen. Über die Zeitdifferenzen erhofft man sich eine Aussage treffen zu können, ob verschiedene Rechner ähnliche Zeiten vorweisen, die somit am gleichen Gerät angeschlossen sein sollten.

6.7.2 Erkundung von Hostdiensten

Für die Gewinnung der Spanning Tree Daten wird ein ganzer IP-Adressbereich abgescannt. Dieser IP-Bereich kann auf das komplette Netzwerk ausgedehnt werden. Der Vorgang würde einige Minuten länger dauern, was in den Mitternachtstunden kein nennenswertes Problem ist. Der Vorteil ist, auch falsch konfigurierte Geräte in anderen IP-Bereichen zu finden. Da für dieses Feature der gesamte Bereich getestet wird, ist es keine nennenswerte Mehrarbeit, eine Auswahl an Services auf jede IP-Adresse zu prüfen. Der Vorteil ist die automatische Erkundung des Netzwerks nach Serverdiensten, die ähnlich automatisch in Nagios integrierbar sind. Auf diese Weise können alle Server überwacht werden, ohne eine aufwendige Konfiguration für jeden einzelnen per Hand zu erstellen. Vor allem werden neue Dienste auch ohne Anmeldung im Rechenzentrum gefunden. Dienste bieten immer wieder Angriffspunkte aus dem Internet, die bei Erfolg das Netzwerk belasten können. Um die Automation zu kontrollieren ist es nicht einmal notwendig, dass die Konfigurationen unmittelbar in das System integriert werden, sondern die reine Erzeugung wird viel Arbeit abnehmen. Die Verschiebung der Konfiguration in den vorgesehenen Ordner der Nagioskonfigurationen per Hand ist eine Kleinigkeit.

6.7.3 Optimierung der Topologie durch Spanning Tree

In Kapitel 3.6 ist die Funktionsweise des Spanning Trees erläutert. Die Aussage, dass der Spanning Tree Schleifen in einem Netzwerk vermeidet, kann sich zu Nutze gemacht werden, um eine redundante Verkabelung unter den Switches zu erreichen. Der Vorteil der Redundanz liegt auf der Hand. Trotz des Ausfalls einer Leitung oder Netzwerkkomponente, bleibt das Netzwerk in Takt und kann ohne Beeinträchtigung der Anwender wieder in einen konsistenten Zustand gebracht werden. Es wäre zu analysieren, wie eine zusätzliche Verkabelung der bestehenden Geräte eine redundante Anbindung jedes Switches ermöglicht. Durch den Spanning Tree, sollte es möglich sein, die optimale Sterntopologie zu erhalten. Im Fehlerfall darf sich die logische Topologie wandeln, da eine geringe Performanzeinbuße im Gegensatz zu einem Totalausfall tolerierbar ist. Für diese Konstruktion bedarf es sicherlich einer manuellen Anpassung der Switchidentifikatoren, bzw Prioritäten, um den Stern logisch zu erhalten, um nicht die Switches anhand ihrer selbst vergebenen Priorität aufstellen zu müssen.

6.7.4 Erkennung von Trunks

Bisher diente Trunking der Erhöhung der Leitungskapazität bei Switches, deren Uplinkport zu schwach bemessen ist. Die Bündelung mehrere Leitungen zweier Geräte erlaubt

so einen besseren Datendurchsatz. Wird Trunking eingesetzt, erwartet der Administrator, dass die Performanz des Uplinks besser ist. Eine echte Kontrolle gibt es in der Regel nicht. Wie in Abschnitt 5.2 bereits für das Monitoring mittels Nagios erwähnt, macht es keinen Sinn permanent einen Bildschirm zu betrachten, ob ein Fehler auftritt. Ebensowenig ist es unnötig, dass der Administrator regelmäßig alle Switches kontrolliert, ob an den jeweiligen Trunkports vergleichbar viele Daten weitergeleitet werden. Unter Umständen kann ein Port des Trunks kaputt gehen, oder eine Datenkabel verletzt werden. Diese Fehlererkennung kann automatisch erfolgen, indem Trunks bei der Nachbarschaftserkennung mit berücksichtigt werden. So ist zum Einen ein optisches Bild der Leitungskapazitäten möglich und zum Anderen kann die Funktionsfähigkeit der Trunks automatisch geprüft und gemeldet werden. Gerade hinsichtlich der Performanzanalyse spielen Trunks eine wichtige Rolle, um die Datenlast individuell anzupassen.

6.7.5 Portbasierte Informationen der Switches

Die in Abschnitt 6.5 beschriebenen Performanzdaten der Switchports sind ein erster Schritt in die Kontrolle des Netzwerks. Allerdings hat die bisherige Anwendung nur statistischen Charakter. Es werden auf den jeweiligen Switch angepassten Webseiten nur der ein- und ausgehende Datenverkehr dargestellt. Was genau an dem jeweiligen Switch und dessen einzelner Ports passiert, bzw. warum etwas nicht passiert ist nicht zu erkennen. Um detailliertere Informationen zu bekommen existiert RMON[8]. Es stellt eine Menge statistischer Daten bereit, die für die Fehleranalyse sehr hilfreich sind. Der Zugriff auf die Daten über das Webinterface soll in Zukunft möglich sein. Desweiteren wäre es denkbar, markante Angaben, wie erkannte Kollisionen für einen Port, auf dem Webinterface direkt visuell hervorzuheben. Bedenkt man die Fähigkeiten von *CiscoView* aus Abschnitt 2.3, in dem jeder Switch visuell in einem Webinterface darstellbar ist, so ist eine besser Fern-Fehleranalyse möglich. Ein Switch muss nicht photorealistisch dargestellt sein, aber zumindest sollte ein einfacher Zugriff auf möglichst viele Daten, die ein Gerät bereitstellt, möglich sein. Der umständliche und teilweise unterschiedliche Weg der Datenermittlung über Telnet aus den verschiedenen Geräten, könnte so minimiert werden.

[8] Remote Monitoring

7 Resümee

Die ursprünglich gesetzten Ziele sind erreicht. Durch den Einsatz von Netdisco, mit der Spanning-Tree-Erweiterung, ist es möglich die komplette Campustopologie zu erfassen und in einer Datenbank bereitzustellen. Desweiteren werden durch Netdisco die angeschlossenen Hosts und deren Elternknoten ermittelt. Durch den Einsatz der SQL-Datenbank stehen die Daten für jegliche Anwendung zur weiteren Verarbeitung zur Verfügung. Die Darstellung erfolgt wahlweise über ein Kommandozeilentool, eine Grafik im SVG-Format unter der Verwendung von GraphViz oder anhand der Einbettung in Nagios.
Die Probleme bezüglich des CDP-Protokolls sind temporär gelöst, stellen aber keine dauerhafte Lösung dar. Wenn sich der Trend weiterentwickelt, dass CDP nur noch von Ciscogeräten unterstützt wird, wird die Funktionalität von Netdisco verloren gehen. Eine alternative Erkundung mittels LLDP ist möglich, bedeutet aber, dass die Unterstützung von Ciscohardware entfällt, da nicht zu erwarten ist, dass sich Cisco dem LLDP-Standard unterwirft. Der einzig gemeinsame Nenner hinsichtlich Nachbarschaften ist der Spanning Tree. Diese Funktionalität wird weiterhin von allen Geräten unterstützt. Eine Neuentwicklung von Netdisco auf Basis von Spanning Tree ist sicherlich die zukunftssicherste Variante. Die Ermittlung der Hosts anhand der Forwardingtabellen und der ARP-Caches sollte kein nennenswertes Problem in einer Neuentwicklung sein. Der Aufbau der Datenbanktabellen ist sinnvoll gewählt und kann mit einigen Ergänzungen weiter verwendet werden.
Nagios stellt ein sehr anpassbares System dar. Die Wahl war für den Einsatzzweck die Richtige. Durch die Verwendung simpler Textdateien ist es für externe Programme einfach möglich die Konfigurationsdateien zu generieren und ggf. auch nach Inhalten zu prüfen. Bei der Entwicklung entschied ich mich für die Neuerzeugung jeder Datei, anstatt jede Datei zu prüfen, ob eine Änderung zu verzeichnen ist. Es ist zu erwarten, dass eine weitreichende Prüfung so vieler Dateien deutlich länger dauert, als eine Neuerzeugung. Die Stabilität des Systems ist während dieses Vorgangs nicht gefährdet. Die Erzeugung der Objekte funktioniert zufriedenstellend durch ein Perlscript. Die kompletten Konfigurationsmöglichkeiten von Nagios sind noch nicht voll ausgeschöpft und erlauben eine eigenständige Arbeit bzw. eine wochenlange Auseinandersetzung mit den Möglichkeiten, die das System bietet.
Die schwierigsten Hindernisse waren die Abbildung der Switchdaten in verwertbare Zusammenhänge, um eine stabile, logische Darstellung der Topologie zu erlangen. Die Hardwareprobleme hinsichtlich der Protokollunterstützung verlangten einige Überarbeitungen während der Entwicklung, da die Probleme nach und nach aufgetreten sind. Besonders der unmittelbare Einsatz der Umgebung in einem realen Netzwerk zeigte Probleme, die in theoretischen Überlegungen oder simulierten Umgebungen evtl. nie aufgetreten wären.

Der Einsatz einiger älteren Switches für eine Testumgebung erlaubt es, Tests in einem realen Netzwerk, welches an dem bestehenden angeschlossen ist, durchzuführen, ohne dieses zu stören.
Eine merkbare Verbesserung ist das optische Warnsystem, welches in die Startseite von Nagios integriert worden ist. Die Auswahl der überwachten Geräte erfolgte bislang durch Namen, die es in einer Liste zu finden gilt. Auch wenn diese alphabetisch sortiert ist, erlaubt die optische Trennung in Gebäude und Flure eine deutlich schnellere Navigation. Das Auftreten der kumulierten Marker innerhalb der Grafik erlaubt erste Annahmen bzgl. des Problems. In der ursprünglichen Variante können keine Beziehungen zwischen den Geräten in der Listendarstellung festgemacht werden. Die schnelle Orientierung wird in naher Zukunft eine Erweiterung zur Anzeige der Serverüberwachung verlangen.
Die Ergänzung der Switchkonfigurationen zur Anzeige der Performanzdaten jedes Switches pro Port erlaubt einige Vermutungen anzustellen. Einige Geräte verzeichnen weder angeschlossene Hosts, noch Datenverkehr über einen längeren Zeitraum. Daher ist eine Optimierung der Portverfügbarkeit in einigen Bereichen anzusetzen, um die teuren Geräte effektiver einzusetzen. Einige Flure sind derzeit auf L1/L2-Switches angewiesen, die durch eine Umverteilung ersetzt werden können. Die bisherige Annahme, neue Switches kaufen zu müssen, kann vorerst verworfen werden.

A Installationshilfe

Das entwickelte System verwendet sowohl fertige Produkte, als auch eigene Entwicklungen. Die Open Source Gemeinde stellt einige brauchbaren Tools zur Verfügung. Von diesem riesigen Reservoir wird verwendet:

- Netdisco von Max Baker, http://www.netdisco.org
- Nagios von Ethan Galstad, http://www.nagios.org
- MRTG von Tobias Oetiker, http://oss.oetiker.ch/mrtg/

A.0.6 Systemkonfiguration

Als Betriebssystem wird auf dem Server SUSE[1] in der Version 10.0 verwendet. Als Server kommt ein Standard 1HE Gerät zum Einsatz mit 2GByte Arbeitsspeicher und einem Dualcore Pentium 4 Prozessor. Die erste Installation funktionierte auf einem Standard-PC mit zwei Pentium 2 450MHz Prozessoren und 500 MByte Arbeitsspeicher. Der Umzug des Systems auf die stärkere Maschine war erst notwendig, seit die Performancechecks durch MRTG eingesetzt worden sind. Das reine Netdisco-Nagios Gespann ist mit dem schwächeren System ausgekommen.

Distributionssoftware

Die Verwendung von Programmen, die bei der Distribution mitgeliefert werden erspart viel Arbeit. Deswegen werden die Programme *Apache2*, *Postgres8.03*, *MRTG*, *GraphViz* und *net_snmp* von der Suse-CD installiert. Der Apache wird von Nagios und MRTG verwendet. Beide Tools sind bezüglich der Version unproblematisch. Netdisco bietet zum Zeitpunkt der Arbeit keine Möglichkeit mit Apache2 zu arbeiten. Bezüglich der Datenbank scheint Netdisco keine Einschränkungen bei der Version zu haben.

Firewall

Da das System von Außen zu erreichen ist, sollte der Server mit einer aktiven Firewall ausgestattet sein. Prinzipiell braucht nur der Port 80 für den Webserver, der von Nagios verwendet wird, und SSH von Außen zugänglich zu sein. Netdisco, MRTG und der Datenbankserver begnügen sich mit lokalen Zugriffsrechten. Falls jedoch zur Pflege

[1] http://www.suse.de

des Systems externen Zugriffe notwendig sind, sollte der Port 5432, für Postgres, offen sein. Für eine externen Betrachtung der Datenbanktabellen empfiehlt sich das Tool PGAdmin3 auf einem Clientrechner.

Benutzer und Gruppen

Um möglichst Nahe an den Installationsanleitungen der Programme zu bleiben, emphielt es sich die Benutzer *netdisco* und *nagios* anzulegen. Desweiteren werden die Gruppen *nagios*, *netdisco* und *mrtg* gebraucht. Absolut wichtig ist die Gruppe *nagcmd*, die für die Ausführung der CGI-Scripte im Nagioswebinterface notwendig ist. In diese Gruppe muss mindestens der User, unter dem der Apache läuft. Unter Suse ist dies *wwwrun*.

Verzeichnisstrukturen

Da an der Universität Koblenz der Unixpfad */usr/local/* für alle Rechner ein Netzlaufwerk ist, werden die Programme nicht in den Standardpfad kopiert. Bei der Entwicklung habe ich mich für den Pfad */usr/monitor* entschieden. Alternativ hätte man den Ordner unter */opt* legen können. Es werden folgende Pfade benötigt:

- /usr/monitor/nagios
- /usr/monitor/netdisco
- /usr/monitor/mrtg
- /usr/monitor/joghurt
- /usr/monitor/joghurt/cgi
- /usr/monitor/joghurt/http

Der Ordner *joghurt* enthält die eigenen Implementationen. Der Name hat keine systemspezifische Bedeutung, da ich keinen brauchbaren Name gefunden hatte. Der Name ist ein Relikt aus meiner Zeit der LAN-Partys.

A.0.7 Netdisco

Netdisco ist in Perl implementiert und verlangt daher keine Kompilierung. Relevant ist die Konfigurationsdatei *netdisco.conf*, in der zum Einen der Homepfad zu ändern (*/usr/monitor/netdisco*) und zum Anderen die Einstellungen wie der Community-String auf das aktuelle Passwort zu setzen ist. Im gleichen Zug ist für die Datenbank ein das Passwort und der Datenbankbenutzer zu setzen.
Um die Konfiguration in Postgres zu erleichtert sind im Order *sql* im Netdiscopfad einige Scripte, die die Datenbanktabellen anlegen. In diesem Zug wird auch der Datenbankbenutzer angelegt. Bei der PostgreSQL-Konfigurationen sollte in der postgresql.conf-Datei der Listenport gesetzt werden und die Zugriffsrechte in der pg_hba.conf Datei.
Netdisco ist in Perl geschrieben und verwendet einige Module. Man sollte möglichst viele

Module von der Distribution verwenden, da so einige Pfadangaben bereits richtig gesetzt sind. Bei meiner Distribution waren das die Module:

- DBD_Pg
- db_devel für die db.h

Die folgenden Module sind alle über CPAN zu beziehen:

- SNMP::info
- Digest::Md5
- DBI
- DB_File
- Compress::Zlib
- Graph
- GraphViz
- Net::NBName

Die letzten drei werden zwar nicht direkt verwendet, führen aber bei irgendwelchen Prüfungen in Netdisco zu einem Programmabbruch. Hingegen der Webanleitung zu Netdisco [Netdisco] wird auf Apache verzichtet, da dieser lediglich für die Netdisco-Weboberfläche gebraucht wird.

A.0.8 MRTG

MRTG ist sehr anspruchslos bei der Konfiguration. Wenn es durch die Distribution installiert ist, sind lediglich die Pfade der von mir implementierten Scripte und die Apache2-Konfiguration (Abbildung A.1) bzgl. des Ordners der erzeugten Webseiten anzupassen.

A.0.9 Nagios

Nagios ist bei einigen Distributionen enthalten, jedoch schreitet die Entwicklung in den letzten Monaten enorm voran, sodass zum derzeitigen Zeitpunkt die Version 2.5 verfügbar ist. Entscheidet man sich dazu die aktuelle Version zu nutzen, ist eine Kompilierung notwendig. Einziges Paket, welches nach zu installieren ist, ist das Paket *gd_devel* der GD-Grafikbibliothek zur Erzeugung der Maps. Der Kompilierungsvorgang (Abbildung A.2) ist, unter Suse, unproblematisch in wenigen Schritten durchgeführt. Der Vorgang ist aus [Barth05]. Ich kann dieses Buch nur jedem empfehlen, der Nagios bislang noch nicht eingesetzt hat und keinerlei Erfahrung mit den einzelnen Konfigurationen hat. Die folgenden Anpassungen setzen die Kenntnisse aus dem Buch voraus. Da die Konfigurationsdateien an einer anderen Stelle abgelegt sind, kann, bei einem Update von

```
ScriptAlias /mrtg/cgi-bin /usr/monitor/joghurt/cgi/
<Directory "/usr/monitor/joghurt/cgi/">
        AllowOverride
        Options ExecCGI
        SetHandler cgi-script
        Order allow,deny
        Allow from all
</Directory>
Alias /mrtg /usr/monitor/mrtg/mrtghttp/
<Directory "/usr/monitor/mrtg/mrtghttp/>
        Options All
        AllowOverride
        Order allow,deny
        Allow from all
</Directory>
```

Abbildung A.1: Apache2-Konfiguration für MRTG-Webseiten

Nagios, exakt diese Reihenfolge eingehalten werden, ohne das die System-Einstellungen verloren gehen. Zu dem Nagios-Sourcecode sind noch zusätzlich die Plugins (Abbildung A.2) zu installieren, die extra herunterzuladen sind. Bei der Sourcecodeübersetzung von Nagios besteht die Möglichkeit einen Satz Beispielkonfigurationen zu erzeugen. Alternativ können die Konfigurationen aus meiner Umgebung verwendet werden, die in den Ordner */etc/nagios/myconfigs* zu kopieren sind. Der Inhalt des Ordners ist in Abbildung A.3 abgebildet. Die Order aus Zeile 6,8 und 11 werden automatisch mit Inhalten durch das Programm *doNagiosObjects.pl* gefüllt und brauchen nicht kopiert zu werden. Hingegen Ordner *externhosts* in Zeile 7 enthält Hostdefinitionen, die nicht automatisch erzeugt werden. Dieser beinhaltet derzeit die Konfiguration, um den SPSS-Server in Landau prüfen zu können. Alle Konfigurationen, bis auf die Hostkonfigurationen, sind im Anhang B aufgelistet. Desweiteren sind die Dateien nagios.cfg und resource.cfg entsprechend der eigenen Installation anzupassen. Für weitere Konfigurationsanweisungen sei auf [Barth05] Seiten 25 bis 40, sowie 58 - 60 verwiesen.

A.0.10 eigene Implementationen

Der Pfad */usr/monitor/joghurt/* enthält die ergänzenden Implementationen. Die Perlmodule sowie einige Programme sind in vorangegangenen Kapitel bereist erläutert worden. *addHostTable.pl* wird derzeit noch verwendet, um auf Basis des DNS zu IP-Adressen die Hostnamen zu finden. Gesteuert wird dies und *doNagiosObjects.pl* durch das Shellscript *nagObj.sh*. Im Ordern *cgi* sind die CGI-Scripte und HTML-Templates für die Performanzseiten der Switches. Der Ordner *http* beinhaltet die Scripte und Templates für das optische Warnsystem. Um die Templates nutzen zu können ist das Paket HTML::Template notwendig, welches über CPAN zu beziehen ist. Desweiteren müssen

```
?:> ./configure --prefix=/usr/monitor/nagios \
      --sysconfdir=/etc/nagios \
      --localstatedir=/var/nagios \
      --with-command-group=nagcmd \
?:> make all
?:> make install
?:> make install-init
?:> make install-commandmode
?:> ./configure --prefix=/usr/monitor/nagios \
      --sysconfdir=/etc/nagios \
      --localstatedir=/var/nagios \
```

Abbildung A.2: Kompilierungsschritte der Nagiosplugins

```
01: drwxrwx---  7 root   nagcmd  424 Sep 18 11:27 .
02: drwxrwxr-x  3 nagios nagios  192 Sep 19 09:17 ..
03: -rwxrwxr-x  1 nagios nagcmd 3789 Sep  1 14:26 checkcommands.cfg
04: -rwxrwxr-x  1 nagios nagcmd 1659 Sep  2 09:12 contactgroups.cfg
05: -rwxrwxr-x  1 nagios nagcmd 6391 Sep  2 11:25 contacts.cfg
06: drwxrwxr-x  2 nagios nagcmd 4984 Sep 12 11:25 devices
07: drwxr-xr-x  2 nagios nagcmd   80 Aug 31 15:30 externhosts
08: drwxrwxr-x  2 nagios nagcmd 6040 Sep  1 22:20 extinfo
09: -rwxrwxr-x  1 nagios nagcmd 3006 Sep  1 17:55 genericHosts.cfg
10: -rwxrwxr-x  1 nagios nagcmd 2679 Sep 18 11:27 hostgroups.cfg
11: drwxrwxr-x  2 nagios nagcmd   48 Sep 20 11:00 hosts
12: -rwxrwxr-x  1 nagios nagcmd  836 Sep 14 14:27 servicegroups.cfg
13: drwxrwxr-x  2 nagios nagcmd  328 Sep 18 11:27 services
14: -rwxrwxr-x  1 nagios nagcmd 1924 Sep  2 11:23 timeperiods.cfg
```

Abbildung A.3: Konfigurationenorder: myconfigs

```
01: -rwxrwxr-x   1 nagios nagios  1035 Aug 29 18:12 DeviceComponent.pm
02: -rwxrwxr-x   1 nagios nagios  1336 Jun  8 14:42 HostComponent.pm
03: -rwxrwxr-x   1 nagios nagios  1552 Jun  8 14:42 NetComponent.pm
04: -rwxrwxr-x   1 nagios nagios  2745 Jul  6 13:52 addHostTable.pl
05: drwxrwxr-x   3 nagios nagios   232 Jun 28 15:03 cgi
06: -rwxrwxr-x   1 nagios nagios  1738 Jul 19 17:17 checkMRTG.pl
07: -rwxrwxr-x   1 nagios nagios  4650 Aug 29 16:56 doNagiosObjects.pl
08: -rwxrwxr-x   1 nagios nagios  3303 Jun  8 16:12 drawGraph.pl
09: -rwxrwxr-x   1 nagios nagios  6416 Jun 14 14:28 findNext.pl
10: drwxrwxr-x   6 nagios nagios   144 Aug 17 16:40 http
11: -rwxrwxr-x   1 nagios nagios  2570 Jun  8 14:42 imapdata.pl
12: -rwxrwxr-x   1 nagios nagios 18259 Aug  1 17:39 libJoghurt.pm
13: -rwxrwxr-x   1 nagios nagios   368 Aug 12 12:13 nagObj.sh
14: -rwxrwxr-x   1 nagios nagios 13701 Aug 16 15:43 netdiscoSTP.pl
15: -rwxrwxr-x   1 nagios nagios   209 Jun  8 14:42 runmrtg2.sh
```

Abbildung A.4: Order der eigenen Implementationen

die Apachekonfigurationen A.5 für Nagios selbst und das Warnsystem angelegt werden.

```
ScriptAliasMatch /alertmap/(.*\.cgi) /usr/monitor/joghurt/http/alertmap/$1
Alias /alertmap /usr/monitor/joghurt/http/alertmap
<Directory "/usr/monitor/joghurt/http/alertmap">
        Options All
        AllowOverride AuthConfig
        Order allow,deny
        Allow from all
</Directory>

#################################################################

ScriptAlias /nagios/cgi-bin /usr/monitor/nagios/sbin/
<Directory "/usr/monitor/nagios/sbin/">
AllowOverride All
        Options None
        SetHandler cgi-script
        Order deny,allow
        Allow from all
        AuthName "Nagios Identifikation"
        AuthType Basic
        AuthLDAPUrl ldap://ldap.uni-koblenz.de:389/ou=People,
                    o=Uni-Koblenz,c=de?uid?sub?(objectClass=*)
        require valid-user
</Directory>
Alias /nagios /usr/monitor/nagios/share/
<Directory "/usr/monitor/nagios/share/">
        Options None FollowSymLinks
        AllowOverride AuthConfig
        Order allow,deny
        Allow from all
</Directory>
```

Abbildung A.5: Apachealiase für Nagios und Warnsystem

B Konfigurationsdateien

B.0.11 contacts.cfg

```
##########################################################
# CONTACTS
##########################################################
define contact{
        contact_name                    nagiosadmin
        alias                           Nagios Admin
        service_notification_period     0x7
        host_notification_period        0x7
        service_notification_options    n
        host_notification_options       n
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           nagios@localhost
        }
define contact{
        contact_name                    imap_deamon
        alias                           Insert info into table
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,u,r,f
        service_notification_commands   service_imapdata
        host_notification_commands      host_imapdata
        email                           nagios@localhost
        }
define contact{
        contact_name                    arndt
        alias                           Uwe Arndt
        service_notification_period     workingday
        host_notification_period        workingday
        service_notification_options    n
        host_notification_options       n
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           arndt@uni-koblenz.de
```

```
        }
define contact{
        contact_name                    cremer
        alias                           Gaby Cremer
        service_notification_period     workingday
        host_notification_period        workingday
        service_notification_options    w,u,c,r
        host_notification_options       d,u,r,f
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           cremer@uni-koblenz.de
        }
define contact{
        contact_name                    litauer
        alias                           Christoph Litauer
        service_notification_period     workingday
        host_notification_period        workingday
        service_notification_options    w,u,c,r
        host_notification_options       d,u,r,f
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           litauer@uni-koblenz.de
        }
define contact{
        contact_name                    mitnacht
        alias                           Sebastian Mitnacht
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,u,r,f
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           mitnacht@uni-koblenz.de
        }
```

B.0.12 contactgroups.cfg

```
###############################################################
# CONTACT GROUPS
###############################################################
define contactgroup{
        contactgroup_name       nagiosadmins
        alias                   Nagios Administrators
        members                 nagiosadmin,root
        }
define contactgroup{
        contactgroup_name       switchadmins
        alias                   Switch Adminsitratoren
        members                 nagiosadmin,root,mitnacht,
                                vieweg,litauer,arndt
        }
define contactgroup{
        contactgroup_name       serveradmins
        alias                   Server Administratoren
        members                 nagiosadmin,root,mitnacht,pidde
        }
define contactgroup{
        contactgroup_name       printeradmins
        alias                   Drucker Administratoren
        members                 nagiosadmin,root,haussmann,
                                cremer,mitnacht
        }
define contactgroup{
        contactgroup_name       hostadmins
        alias                   Host Administrators
        members                 nagiosadmin,root,robert,mitnacht,
                                arndt,litauer,krienke,pidde
        }
define contactgroup{
        contactgroup_name       imapwriter
        alias                   Tabellenschreiber fuer Fehler
        members                 imap_deamon
        }
```

B.0.13 timeperiods.cfg

```
##############################################################
# TIME PERIODS
##############################################################
define timeperiod{
        timeperiod_name 24x7
        alias           24 Hours A Day, 7 Days A Week
        sunday          00:00-24:00
        monday          00:00-24:00
        tuesday         00:00-24:00
        wednesday       00:00-24:00
        thursday        00:00-24:00
        friday          00:00-24:00
        saturday        00:00-24:00
        }
define timeperiod{
        timeperiod_name 0x7
        alias           0 Hours A Day, 7 Days A Week
        sunday          00:00-00:00
        monday          00:00-00:00
        tuesday         00:00-00:00
        wednesday       00:00-00:00
        thursday        00:00-00:00
        friday          00:00-00:00
        saturday        00:00-00:00
        }
define timeperiod{
        timeperiod_name workingday
        alias           Arbeitszeitraum RZ in der Woche
        sunday          00:00-00:00
        monday          08:00-17:00
        tuesday         08:00-17:00
        wednesday       08:00-17:00
        thursday        08:00-17:00
        friday          08:00-14:00
        saturday        00:00-00:00
        }
define timeperiod{
        timeperiod_name workingweek
        alias           Werktage
        sunday          00:00-00:00
        monday          07:00-18:00
        tuesday         07:00-18:00
```

```
wednesday       07:00-18:00
thursday        07:00-18:00
friday          07:00-18:00
saturday        00:00-00:00
}
```

B.0.14 hostgroups.cfg

```
##############################################################
# HOST GROUPS
##############################################################
define hostgroup{
        hostgroup_name  switches
        alias           Switches Campus Koblenz
        }
define hostgroup{
        hostgroup_name  wlan-ap
        alias           Accesspoints WLAN
        }
define hostgroup{
        hostgroup_name  sambaserver
        alias           Samba Fileserver
        members         san1,san2,san3,printhost
}
define hostgroup{
        hostgroup_name  webserver
        alias           Webserver
        members         www,wwww,management,userp,bscw
}
define hostgroup{
        hostgroup_name  imapmailserver
        alias           Mail Server IMAP
        members         mailhost
}
define hostgroup{
        hostgroup_name  smtpmailserver
        alias           Mail Server SMTP
        members         mailhost,deliver
}
define hostgroup{
        hostgroup_name  dnsserver
        alias           DNS Server
        members         cache16,unikodc,rzw2k,mailhost
}
define hostgroup{
        hostgroup_name  poolprinter
        alias           Drucker in Rechnerpools
        members         a003,a015,a024,a230,c207,f112,presse
}
define hostgroup{
```

```
        hostgroup_name  A-Gebaeude
        alias           Workstations_A
        }
define hostgroup{
        hostgroup_name  B-Gebaeude
        alias           Workstations_B
        }
define hostgroup{
        hostgroup_name  C-Gebaeude
        alias           Workstations_C
        }
define hostgroup{
        hostgroup_name  BIB-Gebaeude
        alias           Workstations_BIB
        }
define hostgroup{
        hostgroup_name  D-Gebaeude
        alias           Workstations_D
        }
define hostgroup{
        hostgroup_name  E-Gebaeude
        alias           Workstations_E
        }
define hostgroup{
        hostgroup_name  F-Gebaeude
        alias           Workstations_F
        }
define hostgroup{
        hostgroup_name  G-Gebaeude
        alias           Workstations_G
        }
define hostgroup{
        hostgroup_name  H-Gebaeude
        alias           Workstations_H
        }
define hostgroup{
        hostgroup_name  I-Gebaeude
        alias           Workstations_I
        }
define hostgroup{
        hostgroup_name  K-Gebaeude
        alias           Workstations_K
        }
```

B.0.15 servicegroups.cfg

```
##############################################################
# SERVICE GROUPS
##############################################################
define servicegroup{
        servicegroup_name  RzSrvNoService
        alias              RZ Servers ohne Dienstecheck
        members            bolide,checkalive,termserv,checkalive,
                           baculahost,checkalive,anaconda,checkalive,
                           bigear,checkalive,faxserver,checkalive,
                           mysqlhost,checkalive,pptp,checkalive,
                           news,checkalive
        }
define servicegroup{
        servicegroup_name  IWMServer
        alias              IWM Server
        members            herakles,checkalive,atlas,checkalive,
                           citrix,checkalive
        }
define servicegroup{
        servicegroup_name  Verwaltung
        alias              Server Uni-Verwaltung
        members            dbverw1,checkalive,dbverw2,checkalive
        }
define servicegroup{
        servicegroup_name  PoolDrucker
        alias              Drucker Poolräume
        }
define servicegroup{
        servicegroup_name  RzServer
        alias              RZ Server
        }
define servicegroup{
        servicegroup_name  NoRzServer
        alias              Server nicht im RZ
        members            hopfen,checkalive,malz,checkalive
        }
```

B.0.16 services.cfg

```
define service{
        name                            generic-service
        active_checks_enabled           1
        passive_checks_enabled          1
        parallelize_check               1
        obsess_over_service             1
        check_freshness                 1
        event_handler_enabled           1
        flap_detection_enabled          1
        failure_prediction_enabled      1
        process_perf_data               1
        retain_status_information       1
        retain_nonstatus_information    1
        register                        0
        }
define service{
        use                             generic-service
        hostgroup_name                  dnsserver
        servicegroups                   RzServer
        service_description             Test DNS
        is_volatile                     0
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  serveradmins
        notifications_enabled           1
        notification_options            w,u,c,r
        notification_interval           30
        notification_period             24x7
        check_command                   check_dns!www.google.de!10
}
define service{
        use                             generic-service
        hostgroup_name                  poolprinter
        service_description             Test Druckerstatus in Pools
        servicegroups                   PoolDrucker
        notifications_enabled           1
        is_volatile                     1
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
```

```
        retry_check_interval            1
        contact_groups                  printeradmins
        notification_options            w,c,r
        notification_interval           120
        notification_period             workingweek
        check_command                   check_poolprinter!public
}
define service{
        use                             generic-service
        hostgroup_name                  switches
        service_description             PING-Switch
        notifications_enabled           1
        is_volatile                     1
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  switchadmins,imapwriter
        notification_options            w,u,c,r
        notification_interval           10
        notification_period             24x7
        check_command                   check_ping!100.0,30%!500.0,70%
        }
define service{
        use                             generic-service
        hostgroup_name                  wlan-ap
        service_description             PING-wlanap
        notifications_enabled           1
        is_volatile                     1
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
        retry_check_interval            2
        contact_groups                  switchadmins,imapwriter
        notification_options            w,u,c,r
        notification_interval           10
        notification_period             24x7
        check_command                   check_ping!100.0,20%!500.0,60%
        }
define service{
        use                             generic-service
        hostgroup_name                  webserver
        servicegroups                   RzServer
        service_description             Test auf Apache
```

```
        notifications_enabled           1
        is_volatile                     1
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  serveradmins
        notification_options            w,u,c,r
        notification_interval           30
        notification_period             24x7
        check_command                   check_http
}
define service{
        use                             generic-service
        hostgroup_name                  sambaserver
        servicegroups                   RzServer
        service_description             Test ob Samba erreichbar
        notifications_enabled           1
        is_volatile                     1
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  serveradmins
        notification_options            w,u,c,r
        notification_interval           120
        notification_period             24x7
        check_command                   check_smb
}
```

B.0.17 Automatisch erzeugte Host und Device Konfigurationen

```
define host{
        name                            generic-host
        active_checks_enabled           1
        passive_checks_enabled          0
        notifications_enabled           0
        event_handler_enabled           0
        flap_detection_enabled          1
        failure_prediction_enabled      0
        process_perf_data               0
        retain_status_information       1
        retain_nonstatus_information    1
        register                        0
        }

define host{
        name                            generic-device
        active_checks_enabled           1
        passive_checks_enabled          1
        notifications_enabled           1
        event_handler_enabled           1
        flap_detection_enabled          1
        failure_prediction_enabled      1
        process_perf_data               1
        retain_status_information       1
        retain_nonstatus_information    1
        register                        0
        }
define host{
        use                     generic-host
        host_name               gandalf
        address                 141.26.64.80
        check_command           check-host-alive
        hostgroups              A-Gebaeude
        parents                 cat35-015.uni-koblenz.de,
        max_check_attempts      8
        notification_interval   120
        notification_period     24x7
        notification_options    d,u,r,f
        contact_groups          hostadmins
}
define host{
        use                     generic-host
```

```
        host_name               san1
        address                 141.26.64.130
        check_command           check-host-alive
        hostgroups              A-Gebaeude
        parents                 cisco3550,
        max_check_attempts      8
        notification_interval   120
        notification_period     24x7
        notification_options    d,u,r,f
        contact_groups          hostadmins
}
define host{
        use                     generic-device
        host_name               cat35-015.uni-koblenz.de
        hostgroups              switches
        max_check_attempts      5
        alias                   cat35-015.uni-koblenz.de (UniKo-A-021--)
        address                 141.26.128.115
        check_command           check-host-alive
        parents                 cat6506-002,
        notification_interval   30
        notification_period     24x7
        notification_options    d,u,r,f
        contact_groups          switchadmins,imapwriter
}
define host{
        use                     generic-device
        host_name               cat6506-002
        hostgroups              switches
        max_check_attempts      5
        alias                   cat65-a-k02.uni-koblenz.de (UniKo-A-K18--)
        address                 141.26.128.81
        check_command           check-host-alive
        parents                 cat6506-001,
        notification_interval   30
        notification_period     24x7
        notification_options    d,u,r,f
        contact_groups          switchadmins,imapwriter
}
```

Literaturverzeichnis

[Barth05] Barth Wolfgang, Nagios * System- und Netzwerk-Monitoring, open-source PRESS, 2005

[Barthel05] Barthel Alexander, Analysis, Implementation and Enhancement of Vendor dependent and independent Layer-2 Network Topology Discovery. Diplomarbeit, Chemnitz, April 15 2005

[Bohdan06] Bohdanowicz Frank, SNMP in VNUML Simulationen, Studienarbeit an der Universität Koblenz, 2006

[CiscoSNMS06] Webseite zu Cisco SNMS 1.5, 22.08.2006, http://www.cisco.com/en/US/products/sw/cscowork/ps2408/prod_brochure09186a00801c0a43.htm

[ct0306] c't magazin für Computer, Ausgabe 03/2006 vom 23.1.2006

[Felser06] Felser Max, Link Layer Discovery, 2006

[Harnisch02] Harnisch Carsten, Routing und Switching, bhv 2002

[HeinKoehler02] Hein Mathias und Köhler Peter T., Der IT Reader - Netzwerksicherheit, Fossil 2002

[HPNNM06] Webseite zu HP-Openview, 21.08.2006, http://h40047.www4.hp.com/software/openview/netzwerk-management.php

[HPPerform06] Webseite zu HP-OPenview, 21.08.2006, http://h20229.www2.hp.com/products/ovperf/index.html

[IBMMonitoring06] Webseite zu IBM- Tivoli Monitoring, 21.08.2006 http://www-306.ibm.com/software/info/ecatalog/de_DE/products/U106183E34344S82.html

[IBMAnalyser06] Webseite zu IBM- Tivoli Monitoring, 21.08.2006 http://www-306.ibm.com/software/info/ecatalog/de_DE/products/E106132S89722P00.html

[IBMNetview06] Webseite zu IBM- Tivoli Monitoring, 21.08.2006 http://www-306.ibm.com/software/info/ecatalog/de_DE/products/D106003R74677D75.html

[InTeHa01] Internetwork Technologie Handbook - dt. Übersetzung Uwe Ring, Markt+Technik , 2001

[Klug99] Klug Boris, Diplomarbeit am Fachbereich Informatik, Universität Koblenz, 1999

[LinMag0306] Schwartzkopff Michael, Fernsicht - Das Simple Network Management Protocol, Linux-Magazin 03/06, S. 54

[Meyers] Meyers Großes Taschenlexikon in 24 Bänden, B.I. Taschenbuch Verlag, 1992

[Netdisco] Webseite zu Netdisco, http://www.netdisco.org/install.html, 10.06.06

[PetDav2003] Peterson Larry L. - Davie Bruce S., Computer Netzwerke, dpunkt.lehrbuch 2003

[winkel03] Winkelmann O., Systemüberwachung mit MOM, inforum 3/2003

[Zenk99] Zenk Andreas , Lokale Netze - Die Technik des 21. Jahrhunderts, Addision-Wesly, 1999

Abbildungsverzeichnis