Konzeption und Entwicklung einer Web-Oberfläche zur entfernten Überwachung eines lokalen Netzwerkes mit Hilfe von SNMP-Linux-Tools

Diplomarbeit
im Fach Informationsnetze, Kommunikationstechnik und Netzwerkmanagement
Studiengang Informationswirtschaft
der
Fachhochschule Stuttgart –
Hochschule der Medien

Marco Faisst

Erstprüfer:	Prof. Dr. Wolf-Fritz Riekert
Zweitprüfer:	Prof. Askan Blum

Bearbeitungszeitraum: 1. August bis 1. Dezember 2003

Stuttgart, November 2003

Kurzfassung

Die vorliegende Diplomarbeit ist im Bereich „Netzwerkmanagement“ und beschreibt das Design und die Entwicklung einer Web-Oberfläche, die Auskunft über den Status eines kleineren Netzwerkes gibt.

Anhand grafischer Verlaufsdiagramme besteht die Möglichkeit, sich über Details wie Menge der bisher übertragenen Bytes oder z.B. CPU-Auslastung von Arbeitsplatz-PCs zu informieren. Dabei kommt das Internet-Protokoll SNMP (Simple Network Management Protocol) zum Einsatz, auf das in einem theoretischen Teil näher eingegangen wird.

Zunächst wird behandelt, welche Schritte notwendig sind, um gewöhnliche Windows-2000-PCs zur Überwachung durch den PC mit der Web-Oberfläche vorzubereiten.

Die Umsetzung erfolgt unter dem Betriebssystem Linux, da hier die Auswahl an geeigneten und freien Software-Tools sehr umfassend ist.

Der Zugriff auf die Web-Oberfläche kann durch den Einsatz eines Web-Servers passwortgeschützt sowohl aus dem lokalen Netzwerk als auch extern über das Internet erfolgen.

Schlagwörter: SNMP, Netzwerkmanagement, Netzwerke, Überwachung, Konfiguration, Verwaltung, Web-Oberfläche, Linux, Windows 2000, MRTG, RRDTool, 14all.cgi, Apache, Scripte

Abstract

The topic of this diploma thesis is about network management and describes the design and development of an web interface, which shows the status of a small network.

On the basis of graphic diagrams, it’s possible to be informed about the amount of sent and received data or, for instance, the utilisation of a CPU of a Workstation PC. For this purpose SNMP (Simple Network Management Protocol) is used. This protocol is explained in detail in a theoretical chapter.

At first, it's described which steps are necessary to prepare common Windows 2000 PC's for monitoring by the PC with the web interface.

The implementation takes place under the free operating system Linux, because the range of appropriate software tools is very comprehensive.

By using an web server, the access to the web interface can be made from the local area network or external from the internet and restricted with password protection.

Keywords: SNMP, Network Management, Networks, Monitoring, Configuration, Administration, Web Interface, Linux, MRTG, Windows 2000, RRDTool, 14all.cgi, Apache, Scripts

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

CGI	Common Gateway Interface
DNS	Domain Name Server
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
MIB	Management Information Base
MRTG	Multi Router Traffic Grapher
OID	Object ID
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RFC	Request for Comments
SNMP	Simple Network Management Protocol
SSL	Secure Socket Layer
URL	Uniform Resource Locator

1 Überblick

In der heutigen Zeit sind Computernetzwerke allgegenwärtig. Ob in Firmen, in Universitäten und Fachhochschulen oder in kleinerem Ausmaß manchmal auch in Privathaushalten – überall dort wird vernetzt per Kabel, und neuerdings zunehmend auch per Funk, auf Daten zugegriffen. Als Beispiele genannt seien die Recherche im Internet, Verschicken und Empfangen von E-Mails, Datenübertragung oder gemeinsame Druckernutzung.

In einem Netzwerk existieren meist ein oder mehrere wichtige zentrale Computer, sog. Server, die für spezielle Aufgaben im Netzwerk zuständig sind. Z.B. ein Drucker-Server zur Abwicklung von Druckvorgängen, ein File-Server, der Dateien zum Abruf bereit hält oder grundlegende Server, die bspw. für die Namensauflösung einer Internet-Adresse in eine IP-Adresse zuständig sind (DNS)[1] oder als Proxy[2] fungieren.

Dabei ist natürlich entscheidend, dass gerade diese wichtigen Server, die in der Regel permanent in Betrieb sind, nicht plötzlich aus technischen Gründen ausfallen oder in der Leistung eingeschränkt werden. Falls dies doch einmal passieren sollte, muss die Ursache der Störung möglichst schnell erkannt werden. Bei größeren Netzwerken gibt es häufig mehrere dieser entscheidenden Rechnersysteme.

Bei der im Rahmen dieser Diplomarbeit vorgestellten Web-Oberfläche wird der Online-Status von Rechnern angezeigt sowie anhand von Diagrammen die Anzahl der übertragenen Bytes im Netzwerk, die CPU-Auslastung, der freie Arbeitsspeicher usw.

Die Web-Oberfläche zeichnet sich durch ein übersichtliches Erscheinungsbild aus und beschränkt sich auf wesentliche Funktionen, die für eine Netzwerk-Überwachung von Bedeutung sind. Die Konfiguration erfordert keine besonderen Vorkenntnisse.

Die Umsetzung wird komplett mit dem Betriebssystem Red Hat Linux durchgeführt und sollte (evt. mit kleinen Änderungen bei Verzeichnisangaben o.ä.) für alle gängigen Linux-Distributionen[3] anwendbar sein.

Das Betriebssystem der zu überwachenden Geräte/Rechner ist beliebig, diese müssen lediglich SNMP beherrschen.

---

[1] Domain Name Server
[2] Externer Zwischenspeicher
[3] Linux-Komplettpakete verschiedener Anbieter

Nach der Definition der Zielsetzung und Analyse der Ist-Situation wird zunächst eine Einführung in die Grundlagen des Netzwerkmanagements mit SNMP gegeben.

Danach erfolgt die Konzeption der Web-Oberfläche und in dem Kapitel „Umsetzung“ werden die einzelnen Arbeitsschritte dokumentiert. Die fertige Oberfläche wird schließlich mit Erläuterungen und Screenshots[4] in Kapitel 7 vorgestellt.

Für Quellcode und Kommandozeilenbefehle wird die Schriftart **`Courier`** verwendet. Die Anführungszeichen **„“** sind bei Befehlen stets nicht einzugeben und dienen, falls verwendet, nur der besseren Abgrenzung vom eigentlichen Text.

[4] Bildschirmfotos

2 Zielsetzung

Ziel dieser Arbeit ist es, eine auf HTML basierende Web-Oberfläche zu entwickeln, die die jeweiligen wichtigen Schlüsselsysteme in einem Netzwerk überwacht.
Jeder Rechner ist genauso wie bspw. Hardware-Router[5] und spezielle Switches mit einer eindeutigen, festen IP-Adresse erreichbar.

- Die Web-Oberfläche soll zunächst die Erreichbarkeit der Geräte auf IP-Ebene anzeigen. Der Zustand (d.h. Online-Status) der zu überwachenden Geräte, ist daher grafisch als eine Art Ampelsignal darzustellen; also eine rote Grafik signalisiert einen Fehler und grün steht für keine Probleme. Zusätzlich ist an eine akustische Warnung gedacht.
  Zur schnelleren Identifikation soll neben der IP-Adresse auch eine Beschreibung, Standortangabe o.ä. einsehbar sein sowie ein Fehlerprotokoll mit dem festgehaltenen Zeitpunkt, an dem der Fehler auftrat.
- In einer ausführlicheren Darstellung werden Detailinformationen wie die Anzahl an Bytes, die über einen Router übertragen wurden, die CPU-Auslastung eines Windows-2000-Systems, der freie Arbeitsspeicher und weitere Werte angezeigt. Hierzu kommt das Protokoll SNMP (Simple Network Management Protocol) zum Einsatz sowie Tools, die per SNMP die Geräte abfragen können. Die Anzeige erfolgt in Form grafischer Diagramme mit zusätzlichen Textinformationen.
  Die jeweils tagesaktuellen Diagramme sind auf einer Übersichtsseite zu präsentieren. Durch Anklicken eines der Diagramme öffnet sich eine neue Seite mit allen bisherigen gesammelten Statistiken nach Tag, Woche, Monat und Jahr.
- Es wird Wert gelegt auf eine übersichtliche und einfache Navigation.
- Erweiterung und Konfiguration der Web-Oberfläche ist per Scripte vorgesehen.

Die Web-Oberfläche richtet sich in erster Linie an den/die Administrator(en) eines Netzwerkes. Endanwender oder sonstige nicht zuständige Personen sollen keinen Zugriff auf die Seite erhalten. Um dies zu gewährleisten, erfordert der autorisierte Zugriff einen Benutzernamen und ein Kennwort. Optional ist auch eine verschlüsselte SSL[6]-Verbindung möglich.

---

[5] Zur Weiterleitung von Daten zwischen verschiedenen Netzwerken oder dem Internet und mehreren Rechnern

[6] Secure Socket Layer

3 Stand der Technik

Im professionellen und semi-professionellen Bereich existieren bereits einige kommerzielle Netzwerkmanagementsysteme als Komplettlösung. Hierzu zählen OpenView von Hewlett Packard, IBM Tivoli NetView, SunConnect SunNet Manager, D-View von D-Link oder WhatsUpGold von Ipswitch.

Am bekanntesten dürften OpenView und NetView sein. Hewlett-Packard bietet für OpenView ein eigenes WebSite-Portal an. Unter www.openview.hp.com kann man sich Informationen beschaffen, Demoversionen downloaden, sich für Trainingsprogramme anmelden und vieles mehr. OpenView ist modular aufgebaut, d.h. für verschiedene Einsatzzwecke/Funktionen existieren separate Programmteile.

Ähnlich ist IBM Tivoli NetView[7], das für die Betriebssysteme AIX (eine Unix-Version von IBM), Linux, Sun Solaris, Windows NT und Windows 2000 erhältlich ist.

Die beiden nachfolgenden Abbildungen zeigen Programmoberflächen von NetView:

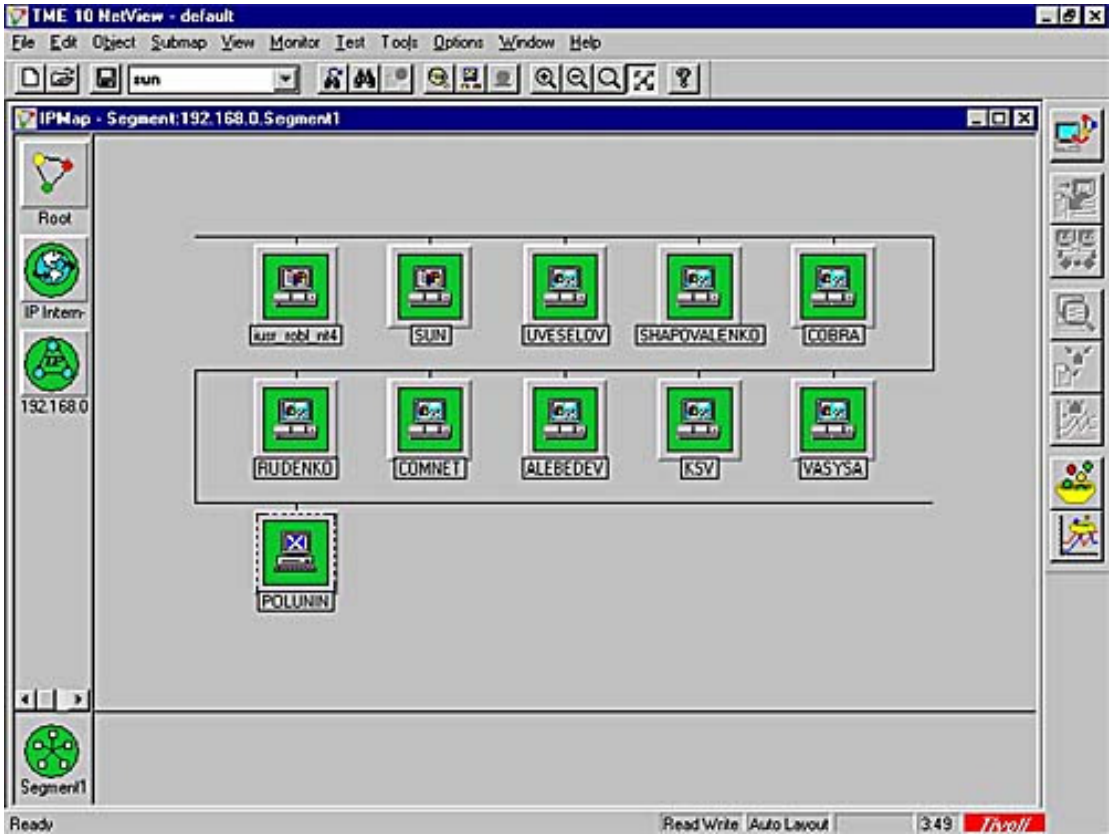

Abbildung 1: Rechner-Segment in NetView

[7] http://www-3.ibm.com/software/tivoli/products/netview/

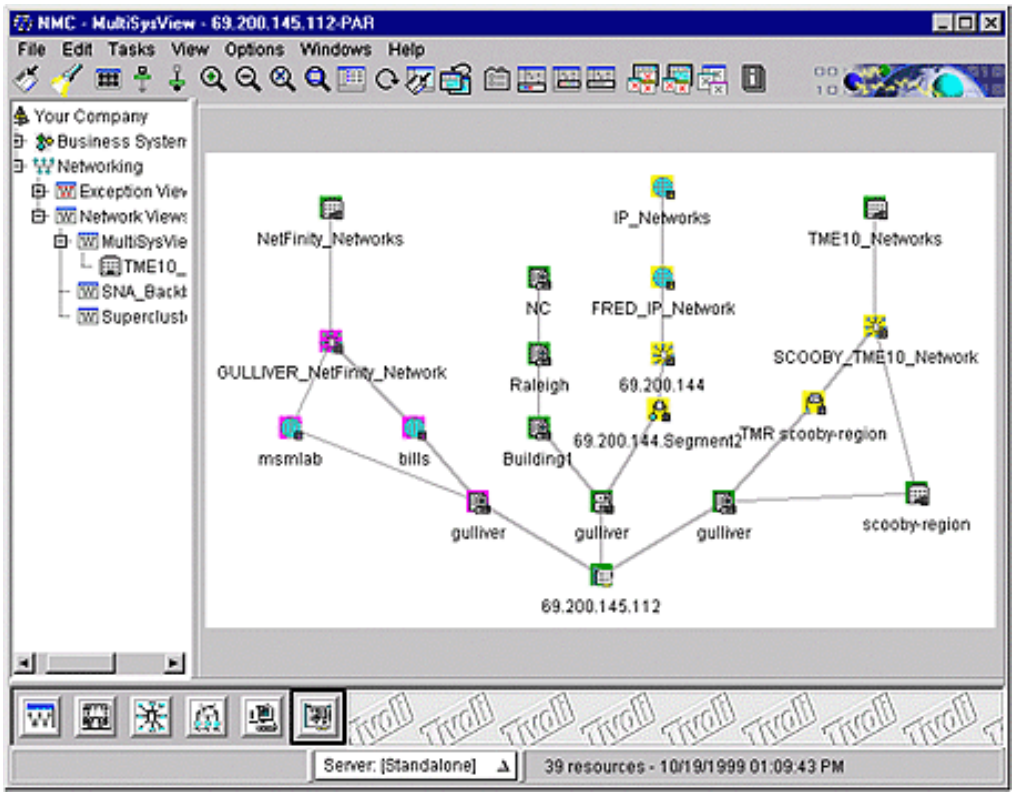

Abbildung 2: NetView-Verknüpfungsdarstellung

Einzelne Rechner oder Netzwerk-Hardware werden als Symbole dargestellt. Linien verbinden die Geräte und sorgen so für Nachvollziehbarkeit der Vernetzung. Schlüsselpunkte wie Router werden mit speziellen Symbolen dargestellt.

Verschiedene Netzwerkabschnitte können zur besseren Übersicht in Segmente unterteilt werden, z.B. nach Stockwerke eines Bürogebäudes.

Fällt ein Gerät aus, wird dies auf der Karte sofort grafisch dargestellt, z.B. durch Aufblinken und mit einem Warnsymbol. So kann das fehlerverursachende Gerät leicht identifiziert werden.

Die Lizenzen und auch Einarbeitungskosten für die genannten Programme erscheinen allerdings möglicherweise im Verhältnis zu einem kleinen Einsatzgebiet recht hoch und häufig wird ein Großteil der Funktionen gar nicht benötigt.

Der OpenSource[8]-Markt bietet hier einige Alternativen, von denen die meisten ursprünglich für das freie Betriebssystem Linux von Linus Torvalds[9] geschrieben wurden.

[8] kostenlose, beliebig veränderbare Software mit Quellcode
[9] entwickelte Anfang der 90er-Jahre als finnischer Student im Rahmen eines Projektes den Unix-Klon „Linux“

Das Angebot reicht von vielen kleinen, mehr oder weniger schwer zu konfigurierenden, SNMP-Kommandozeilentools bis hin zum vollständigen Netzwerkmanagement-System, das über Plugins sogar als Web-Oberfläche genutzt werden kann (Nagios[10], früher Netsaint). Allerdings ist auch bei letzterem die Konfiguration sehr zeitaufwendig und nicht gerade einfach. Außerdem sind viele Funktionen und Möglichkeiten implementiert, die für eine simple Überwachung nicht unbedingt nötig sind.

Die vorliegende Arbeit möchte aber genau dies ermöglichen: eine überschaubare und zugleich funktionale Lösung.

[10] http://www.nagios.org

4 SNMP

SNMP steht wie zu Beginn schon erwähnt für *Simple Network Management Protocol.* Im Rahmen dieses Grundlagenkapitels wird näher auf dieses Protokoll eingegangen.

4.1 Definition

SNMP ist ein Internet-Protokoll und wird im Bereich Netzwerkmanagement eingesetzt. Mit SNMP kann anhand eines zentralen Rechners ein komplettes Netzwerk auf einzelne Werte hin überwacht und verwaltet werden. Bedingung ist, dass die jeweiligen Netzwerkkomponenten SNMP verstehen.

Bei SNMP handelt es sich um ein Protokoll der *Anwendungsschicht* (Schicht 7) im sog. OSI[11]-Schichtenmodell, wie die Abbildung auf der nächsten Seite verdeutlicht.

> *Das OSI-Schichtenmodell legt die Behandlung und Verwaltung der Datenübertragung in einem Netzwerk fest. Das Netzwerk wird dabei in unterschiedliche Ebenen, die sog. Schichten oder Layer gegliedert.* [12]

SNMP benötigt Dienste der *Darstellungsschicht* und macht von dem Protokoll UDP[13] Gebrauch (Schicht 4 - *Transportschicht*).

---

[11] Open Systems Interconnection, Arbeitsgruppe zur Schaffung allgemeiner Standards für offene Systeme, gehört zur ISO (International Standardization Organisation)

[12] Vgl. Markt&Technik-Computerlexikon, 1998

[13] User Datagram Protocol, nicht verbindungsorientiertes Übertragungsprotkoll

Das OSI-Schichtenmodell

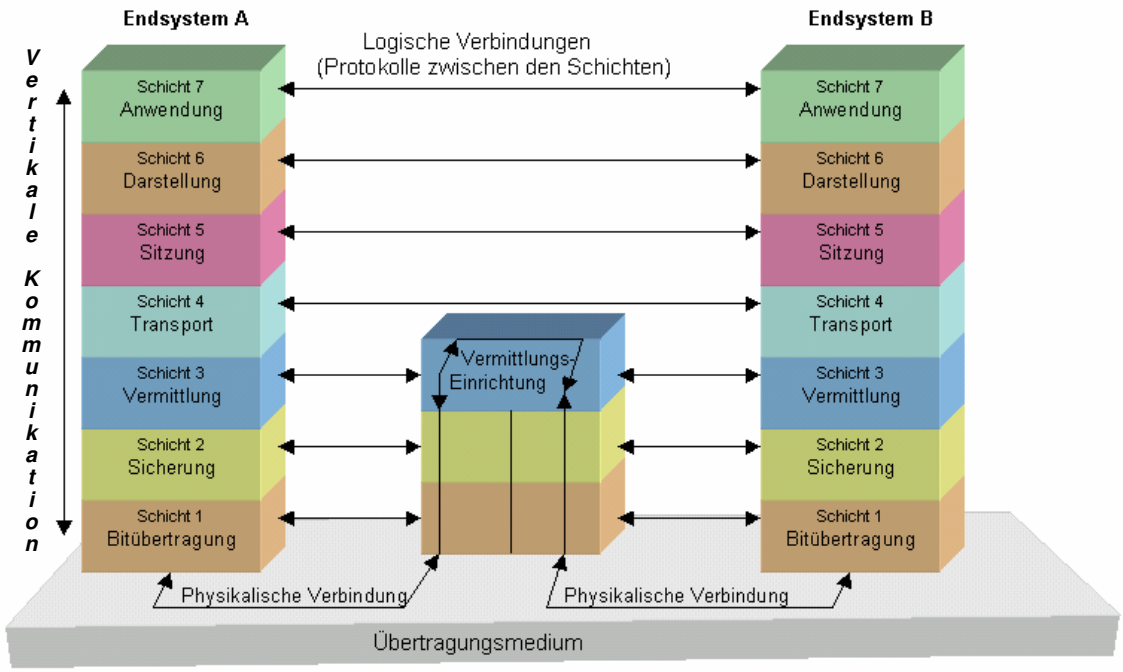

Abbildung 3: OSI-Schichtenmodell [14]

4.2 Entstehung

Entwickelt wurde SNMP 1988 als Nachfolger von SGMP (*Simple Gateway Monitoring Protocol*). Es ist als Teil des „Internet Network Management Framework“ (NMF) in den RFCs (*Request for Comments*, Internet-Textdokumente) definiert. 1990 wurde der SNMP-Standard als Standard im Internet erklärt.

Es existieren drei SNMP-Versionen, wobei die jüngste, SNMPv3, noch nicht so sehr verbreitet ist. Vor allem im Bereich der verbesserten Sicherheit unterscheiden sich SNMPv3 und v2 von SNMPv1.

SNMPv1 ist in den RFCs 1155, 1157 und 1212 definiert, SNMPv2 in den RFCs 1441 bis 1452. Die meisten Netzwerk-Geräte (einer der großen Hersteller ist Cisco Systems) unterstützen sowohl SNMPv1 als auch SNMPv2. [15]

[14] Vgl. http://www.bbeutel.de/osi1.htm
[15] Vgl. http://www.cisco.com/warp/public/535/3.html

4.3 Funktionsweise

Die folgenden Unterkapitel widmen sich dem Ablauf von SNMP-Anfragen und SNMP-Antworten.

4.3.1 SNMP-Agent

Entsprechende SNMP-fähige Geräte wie z.B. Router besitzen eine kleine Software, den sog. SNMP-*Agenten* (auf UDP Port[16] 161), der mit speziellen Befehlen von Netzwerkmanagement-Programmen angesprochen wird.

4.3.1.1 Polling

Durch manuelles oder automatisiertes Abfragen des Agenten mit einem Netzwerkmanagement-System, erhält man Auskunft über z.B. die Bezeichnung, den Betriebszustand und viele weitere feste, aber auch variable, sich verändernde Werte des Geräts. Dieser Vorgang des Einholens von Informationen wird „*Polling*“ genannt.

4.3.1.2 Traps

Der Agent kann bei vorgegebenen Ereignissen außerdem selbstständig (also nicht erst auf Anfrage) sog. „*Traps“*, d.h. Ereignismeldungen, senden, welche durch das Netzwerkmanagement-Programm auf dem UDP Port 162 empfangen werden.

Als Benachrichtigungsform für diese Traps ist z.B. eine Bildschirm-Mitteilung oder das Versenden einer E-Mail möglich, je nach verwendetem Netzwerkmanagement-Programm und Einstellung.

4.3.2 SNMP-Proxy und SNMP-Hard-/Software

Geräte, die nicht SNMP-fähig sind, können durch Verwendung eines SNMP-Proxy dennoch mit SNMP zusammenarbeiten.

Es existieren neben Hardware (Router, Switch, UPS[17], ...) außerdem Software-Programme, die ebenfalls SNMP unterstützen, z.B. professionelle Virenscanner oder das Backup-Programm „BackupExec“.

---

[16] hier: engl. Bez. für eine bestimmte Stelle eines Computersystems an der Daten eines Protokolls übertragen werden, z.B. Port 80 - http, Port 21 - ftp

[17] Uninterruptable Power Supply, Gerät zur unterbrechungsfreien Stromversorgung

Auch das Betriebssystem selbst kann per SNMP auf verschiedene Werte hin abgefragt werden, wenn der SNMP-Dienst[18] installiert und entsprechend eingerichtet ist (siehe Kap. 6.1.2).

4.3.3 Kommunikation zwischen Agent und Managementsystemen

Mit den SNMP-Agenten kommuniziert die Netzwerkmanagement-Software, auch kurz „NMS“ genannt, für „Network Management System“ bzw. „Netzwerkmanagement-system“.

Zur Verdeutlichung der Informationen in den letzten Unterkapiteln hier nun zwei grafische Darstellungen der Kommunikation ohne und mit zusätzlichem Proxy:

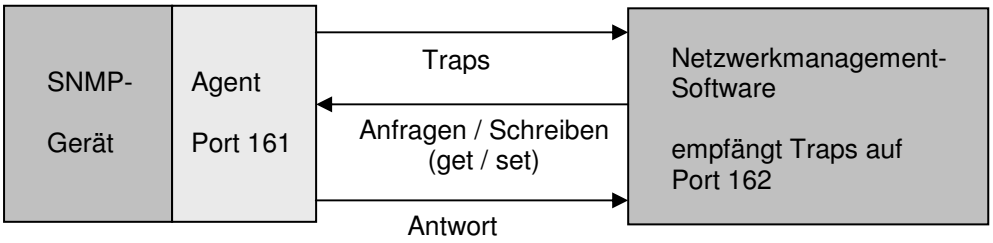

Abbildung 4: Ablauf bei SNMP-fähigen Geräten

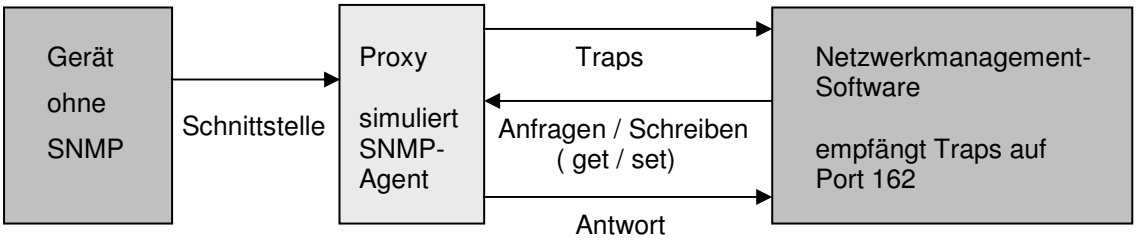

Abbildung 5: Ablauf bei nicht-SNMP-fähigen Geräten

Bei SNMPv2 gibt es für die Kommunikation die folgende sechs Befehle:

GET	Polling, d.h. Einholen von Informationen von dem Agenten
GET-NEXT	Die nächste im Agenten vorkommende Instanz wird zurückgeliefert
GET-BULK	Ganze Tabellen werden in einem Paket abgeholt
SET	Schreiben von Werten (Möglichkeit/Berechtigung vorausgesetzt)
TRAP	Der Agent sendet bei wichtigen Ereignissen *selbstständig* einen Trap
INFORM	Netzwerkmanagement-Systeme können untereinander Traps versenden

[18] Dienste sind im Hintergrund ablaufende Programme

4.4 Technischer Hintergrund

4.4.1 Aufbau eines Datenpakets

Ein SNMPv1-Datenpaket besteht aus zwei Teilen:

Der erste Teil beinhaltet die *SNMP-Version* und einen *Community-Namen*. Eine *Community* stellt eine Art gemeinsames Passwort für SNMP-Agenten und Empfänger dar. Für den Lese-Zugriff wird standardmäßig die Community „*public*“ und für den Schreib-Zugriff „*private*“ verwendet. Aus Sicherheitsgründen sollten diese Einstellungen bei den Geräten in der Praxis geändert werden.

Der zweite Teil beinhaltet die „*Protocol Data Unit*“ (PDU), die aus einen Befehl (z.B. get) sowie dem betreffende Objekt, das mit dem Befehl angesprochen wird, besteht.

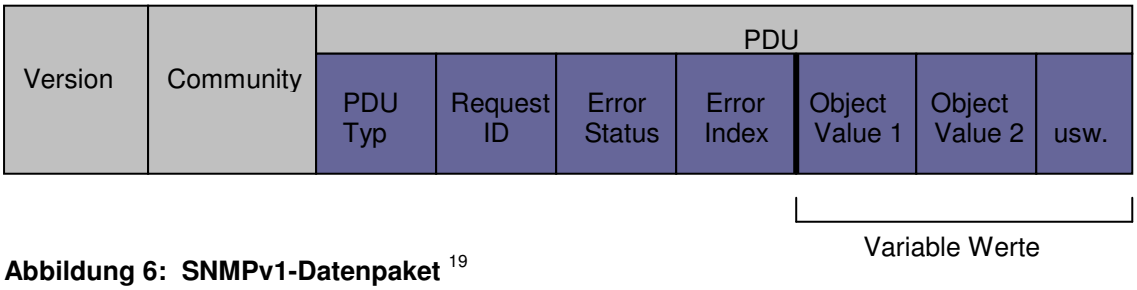

Abbildung 6: SNMPv1-Datenpaket [19]

Objekte stellen die verschiedenen einzelnen Möglichkeiten der abfragbaren und sich ändernden Informationen dar und sind in einer Art Datenbank, genannt MIB (*Management Information Base*), enthalten.

[19] Vgl. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm#xtocid13

4.4.2 Die „Management Information Base“ (MIB)

4.4.2.1 Definition

Hierbei handelt es sich nicht um eine Datenbank im klassischen Sinn, sondern vielmehr um in Baumform hierarchisch verzweigte Objekte.

Die Identifikation der einzelnen Objekte erfolgt über Ziffernketten, die ähnlich wie bei einer IP-Adresse mit Punkten in mehrere Abschnitte aufgeteilt sind. Für die MIB bedeutet dies eine Gliederung in zusammengehörige Kategorien.

Die MIB wurde in ihrer ersten Form 1990 als „MIB I“ standardisiert. 1991 wurde MIB I mit in die erweiterte MIB II übernommen. Die einzelnen MIB-Objekte (sog. *Managed Objects*) sind in einem plattform-unabhängigen Format, welches durch die OSI (*Open System Interconnection*) in der sog. *Abstract Syntax Notation 1* (ASN.1) definiert ist.

4.4.2.2 Aufbau und Bestandteile

Alle Objekte müssen einen *Namen*, eine *Syntax* und eine *Codierung* besitzen.

Mit „Name“ ist eine sog. OID (*Object ID*) gemeint, die das jeweilige Objekt eindeutig identifiziert. „Syntax“ bezeichnet die Art des Datentyps, z.B. Integer oder String (Ganzzahl / Zeichenkette). Die „Codierung“ gibt an, wie die Objekt-Informationen angeordnet sind.[20]

[20] Vgl. http://www.cisco.com/warp/public/535/3.html

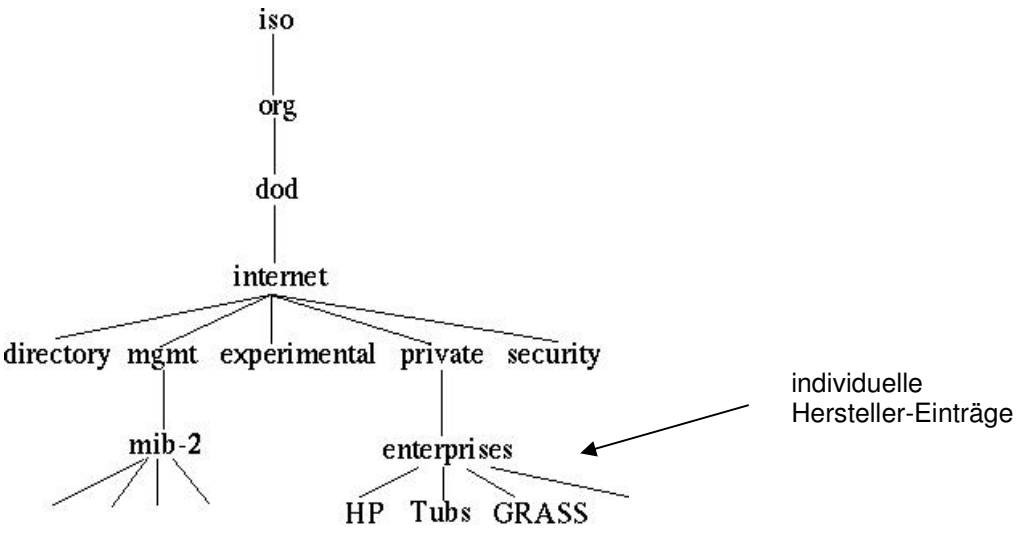

Abbildung 7: MIB-Baumstruktur (vereinfacht)

Das Objekt „sysDescr“ (mit der Object ID 1.3.6.1.2.1.1.1) setzt sich bspw. folgendermaßen zusammen:

iso	org	dod	internet	mgmt	mib	system	sysDescr
1	3	6	1	2	1	1	1

Der Internet-Standard MIB-II ist in RFC 1213 festgelegt und enthält die folgenden 10 Zweige:

System	Informationen über das Betriebssystem des Geräts
Interfaces	Informationen der Netzwerkschnittstellen
At	Informationen der Adress-Übersetzung
ip	Informationen des IP-Protokolls
icmp	Informationen des ICMP-Protokolls
tcp	Informationen des TCP-Protokolls
udp	Informationen des UDP-Protokolls
egp	Informationen des „Exterior Gateway“-Protokolls
transmission	Informationen zur Übertragung
snmp	Informationen über SNMP

4.4.2.3 Zugriff auf MIB-Objekte und Instanzen

Übersicht einiger Standard-MIB-Einträge:

1.3.6.1.2.1.system(1).

SysDescr(1):	Beschreibung des Geräts
SysObjectID(2):	Bezeichnung für die Software des Agenten
SysUpTime(3):	wie lange läuft das System bereits?
SysContact(4):	Name einer Kontaktperson
SysName(5):	Name des Geräts
SysLocation(6):	Der Ort, wo sich das Gerät befindet

1.3.6.1.2.1.interfaces(2).

IfIndex(1):	Schnittstellennummer
IfDescr(2):	Beschreibung der Schnittstelle
IfType(3):	Art der Schnittstelle
IMtu(4):	Größe der MTU
IfSpeed(5):	Übertragungsrate in bps
IfphysAddress(6):	Media-Adresse
IfAdminStatus(7):	gewünschter Zustand der Schnittstelle
IfOperStatus(8):	tatsächlicher momentaner Zustand der Schnittstelle
IfLastChange(9):	vor welcher Zeit hat die Schnittstelle ihren Zustand geändert

. . . .

Besitzt ein Objekt mehrere Instanzen, so muss immer der Wert der gewünschten Instanz angegeben werden.

Existiert nur eine Instanz in einem Objekt, dann ist die Instanz immer 0.

Im Zweig „interfaces“ existiert bspw. ein Objekt „ifNumber“, das über die Instanz **`ifNumber.0`** die Zahl der verfügbaren Netzwerkschnittstellen eines Gerätes darstellt.

Beispiel: `.iso.org.dod.internet.mgmt.mib-2.interfaces.`**`ifnumber`**
numerisch: `.1.3.6.1.2.1.2.1`

Instanz: `.iso.org.dod.internet.mgmt.mib-2.interfaces.`**`ifnumber.0`**
numerisch: `.1.3.6.1.2.1.2.1.0`

4.5 Weitere Eigenschaften von SNMP

SNMP eignet sich nicht nur zur Statusanzeige und zum Auslesen von Werten sowie zur Überwachung, sondern durch Schreibmöglichkeiten auch zur Fernwartung, d.h. Einstellungen der Geräte können über dieses Protokoll auch geändert bzw. korrigiert werden.

Das hat den Vorteil, dass der zuständige Techniker zur Konfiguration oder Fehlerbehebung im günstigsten Fall nicht vor Ort sein muss, sondern vom entfernten Rechner aus arbeitet.

Der MIB-Baum kann um weitere, herstellerspezifische MIB-Informationen erweitert werden, die die jeweiligen Hersteller in Dateiform für ihre SNMP-Hard-/Software anbieten.

5 Konzeption und Anforderungen

Kapitel 5 nennt die verwendeten Entwicklungswerkzeuge sowie die im Einsatz benötigte Hard- und Software. Außerdem wird der Zusammenhang der einzelnen verwendeten Software-Komponenten bildhaft gemacht.

5.1 Allgemeine Konzeption der Web-Oberfläche

Die Abläufe und Stellung der Web-Oberfläche innerhalb des Gesamtkonzepts verdeutlicht die nachfolgende Darstellung:

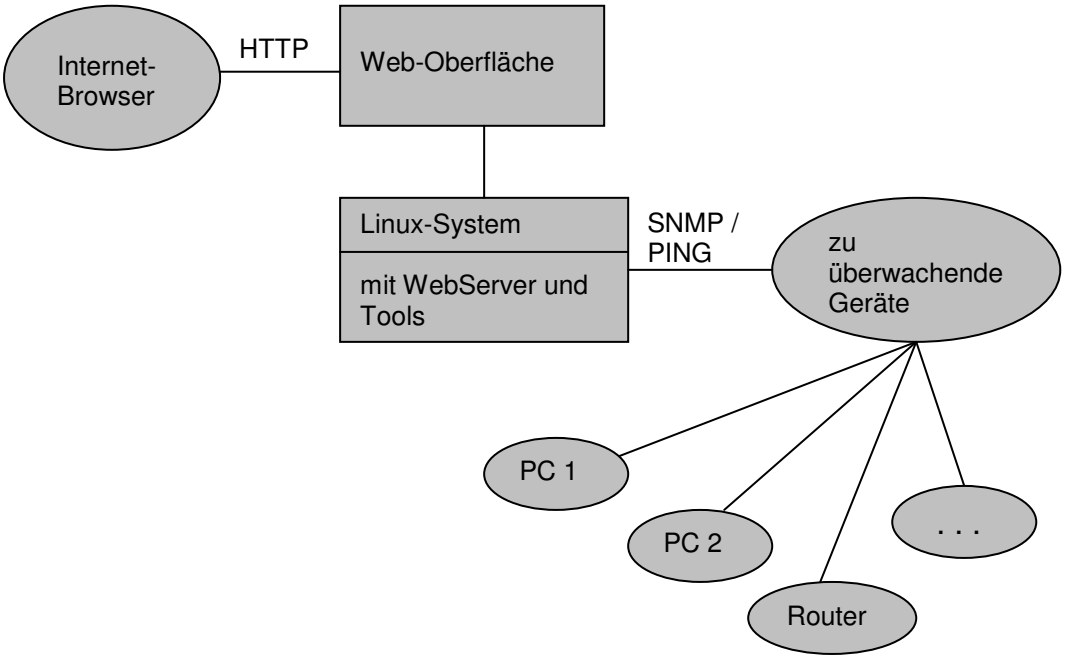

Abbildung 8: Konzept Web-Oberfläche

Den Ausgangspunkt stellt die Web-Oberfläche dar. Sie bildet eine Brücke zwischen dem Benutzer und den im Hintergrund ablaufenden Tools, die im Betrieb nicht sichtbar sind. Die Geräte werden durch diese Tools automatisch in regelmäßigen Abständen abgefragt. Sobald der Benutzer die Web-Oberfläche besucht, werden aus den gesammelten Daten Diagramme erstellt und es wird angezeigt, ob die Geräte noch funktionieren, d.h. online sind.

Konfiguration und Erweiterbarkeit mit zusätzlichen Rechnern wird durch hierfür geschriebene Kommandozeilen-Scripte gegeben.

5.2 Anforderungen

Es folgt nun eine kurze Beschreibung der Anforderungen, die die Grundlage für die Umsetzung bilden.

5.2.1 Betriebssystem und Web-Server

Als Betriebssystem für den Rechner, auf dem die Web-Oberfläche laufen soll, kommt das freie Betriebssystem Linux als Distribution Red Hat 8.0 zum Einsatz.

Red Hat ist neben SuSE eine der meistverbreiteten Distributionen. Nachfolgeversionen sollten ebenso problemlos geeignet sein wie andere Distributionen. Für SuSE Linux bspw. sind höchstens kleinere Anpassungen nötig, wie unterschiedliche Verzeichnispfade oder andere Bezeichnungen.

Der hier zum Einsatz kommende Apache-Webserver, der bei RedHat mitgeliefert wird und „httpd“ genannt wird, heißt bei SuSE „apache“.

5.2.2 Web-Oberfläche

Zur Gestaltung der HTML-Seiten der Web-Oberfläche kommen der integrierte Composer[21] des Web-Browsers[22] „Mozilla“[23] und ein gewöhnlicher Texteditor zum Einsatz.

Auf aufwändiges Grafikdesign, sog. „Cascading Style Sheets“ (CSS), Java, Flash-Animationen usw. wird zugunsten der Funktionalität, Kompatibilität und schnelleren Ladezeiten bewusst verzichtet. Lediglich ein bis zwei kleine JavaScript-Befehle werden Verwendung finden, daher sollte der Browser JavaScript aktiviert haben.

[21] Grafischer HTML-Editor
[22] Programm zum Navigieren im World Wide Web
[23] http://www.mozilla.org

5.2.3 SNMP-Tools, Scripte und sonstige Hilfsmittel

Zum Testen der Erreichbarkeit der Netzwerk-Geräte kommt ein Linux-Shell[24]-Script[25] zum Einsatz.

Das Abfragen dieser SNMP-Geräte und Sammeln von Daten wird das Tool *MRTG*[26] von Tobias Oetiker und Dave Rand in Kombination mit „*RRDTool*“[27] übernehmen.

Durch zwei weitere zu entwerfende Scripte soll die Ergänzung der Web-Oberfläche mit zu überwachenden Rechnern und die Einrichtung von Diagrammen erleichtert werden. Die Diagramme werden dann später mit Hilfe des CGI[28]-Scripts „*14all.cgi*“[29] in die Web-Oberfläche eingebunden und in Echtzeit erstellt.

Damit nicht veraltete Informationen aus dem Browser-Cache[30] dargestellt werden, ist dieser vorher zu leeren und die Seite alle 5 Minuten neu zu laden, sofern sie gerade geöffnet ist.

Zu den grundlegenden Anforderungen zählt noch ein installierter *Perl-Interpreter*[31] (bei RedHat mitgeliefert), da MRTG in der Sprache Perl geschrieben ist.

5.2.4 Zu überwachende Geräte

Als Testsysteme für die Überwachungsdiagramme werden folgende Geräte verwendet:

- ein SNMP-fähiger Home-Office-Router von Zyxel, Modell Prestige
- ein PC mit Microsoft Windows 2000 Professional und installiertem SNMP-Dienst sowie vorhandener SNMP-Unterstützung.

---

[24] Kommandozeile
[25] auf Kommandozeilenebene ausführbare Programme mit Befehlsanweisungen
[26] „Multi Router Traffic Grapher“ http://www.mrtg.org
[27] „Round Robin Database“ http://rrdtool.eu.org
[28] Common Gateway Interface, u.a. zur dynamischen Informationserstellung bei WebSites
[29] http://my14all.sourceforge.net
[30] Lokaler Zwischenspeicher für häufig aufgerufene Seiten
[31] Zur Übersetzung und Ausführung von Dateien mit Perl-Quellcode

6 Umsetzung

In diesem Kapitel werden ausführlich die Schritte beschrieben, die zur Entwicklung der Web-Oberfläche und deren Funktion notwendig waren.
Die Vorbereitungsphase behandelt Kapitel 6.1, die Umsetzung der Web-Oberfläche findet in Kapitel 6.2 statt.

6.1 Vorbereitung der Geräte

Dieses Unterkapitel dokumentiert die Einstellungen an Soft- und Hardware, die erforderlich sind, damit die Geräte später von dem Überwachungs-System abgefragt werden können.

6.1.1 Router

Der hier verwendete Router ist ein Zyxel Prestige. Die Konfiguration kann direkt über das Netzwerk erfolgen. Hierzu sind folgende Schritte durchzuführen:

1.) Per Telnet[32] wird das SNMP-Menü aufgerufen und sichergestellt, dass die Read/Get-Community (aus Gründen der Einfachheit) die Standard-Bezeichnung „*public*“ trägt. Die Write/Set-Community und Traps sind nicht von Bedeutung, da keine Werte geschrieben werden sollen und auch Traps nicht benötigt werden, sondern später nur einfaches Abfragen (Polling) durchgeführt wird.

Der Befehl zum Aufruf des Menüs lautet sowohl unter Windows als auch Linux:

```
telnet [Router-IP]   hier: telnet 192.168.0.1
```

Danach öffnet sich das auf der nächsten Seite als Abbildung 9 gezeigte Menü:

[32] Zur textorientierten Fernkonfiguration von Systemen

```
          Menu 22 - SNMP Configuration

SNMP:
  Get Community= public
  Set Community= public
  Trusted Host= 0.0.0.0
  Trap:
    Community= public
    Destination= 0.0.0.0




    Press ENTER to Confirm or ESC to Cancel:
```

Abbildung 9: SNMP-Konfiguration - Router-Telnet-Menü

2.) Als nächster Schritt wird die Web-Oberfläche der Routers durch die Eingabe http://192.168.0.1 in einem Web-Browser geöffnet. Im Register „*SNMP*“ sollte der Menüpunkt „*Management*“ wie folgt aussehen:

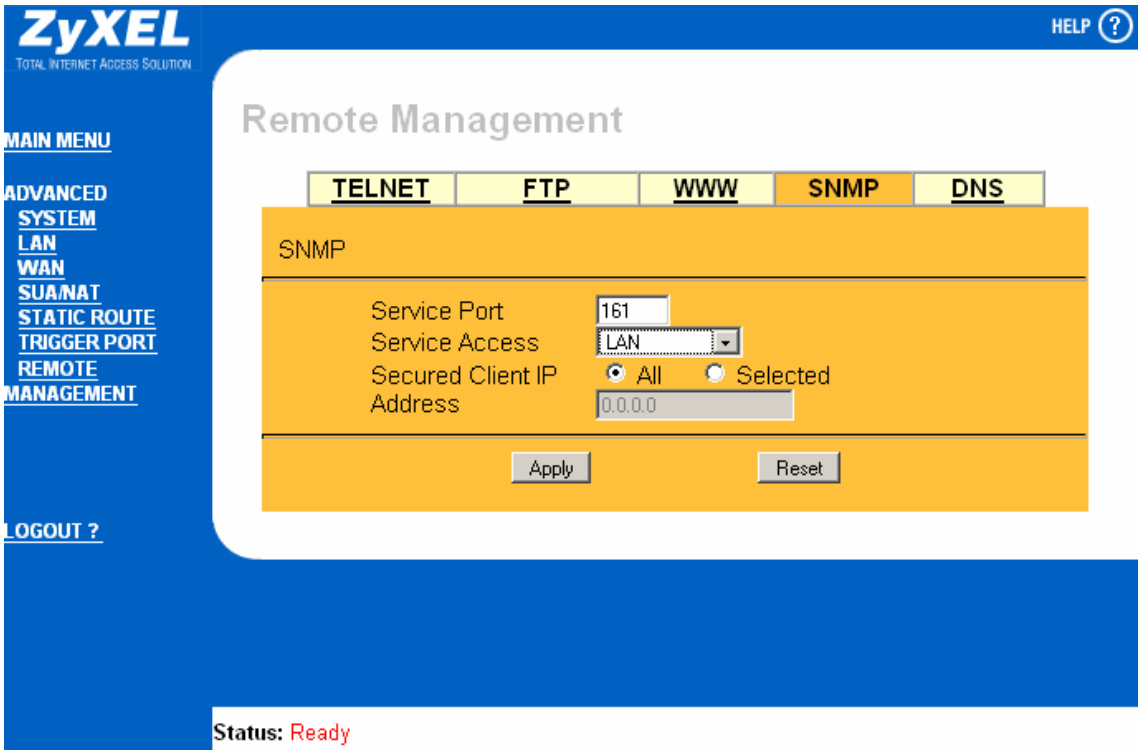

Abbildung 10: SNMP-Konfiguration - Router-Web-Oberfläche

Wichtig ist hier der Service-Port 161, den die Managementprogramme erwarten, damit eine Kommunikation möglich ist.

6.1.2 Windows 2000-Rechner

Damit der Windows 2000-PC vollständig per SNMP abgefragt werden kann, z.B. für den freien Arbeitsspeicher, sind zunächst folgende Schritte an diesem PC notwendig:

1.) Über „*Systemsteuerung / Software / Windows-Komponenten hinzufügen*“ wird „*Verwaltungs- und Überwachungsprogramme / **SNMP***“ installiert. Dieser Dienst wird standardmäßig nicht von Windows mitinstalliert.

Zur Verdeutlichung ist den Abbildungen zu folgen:

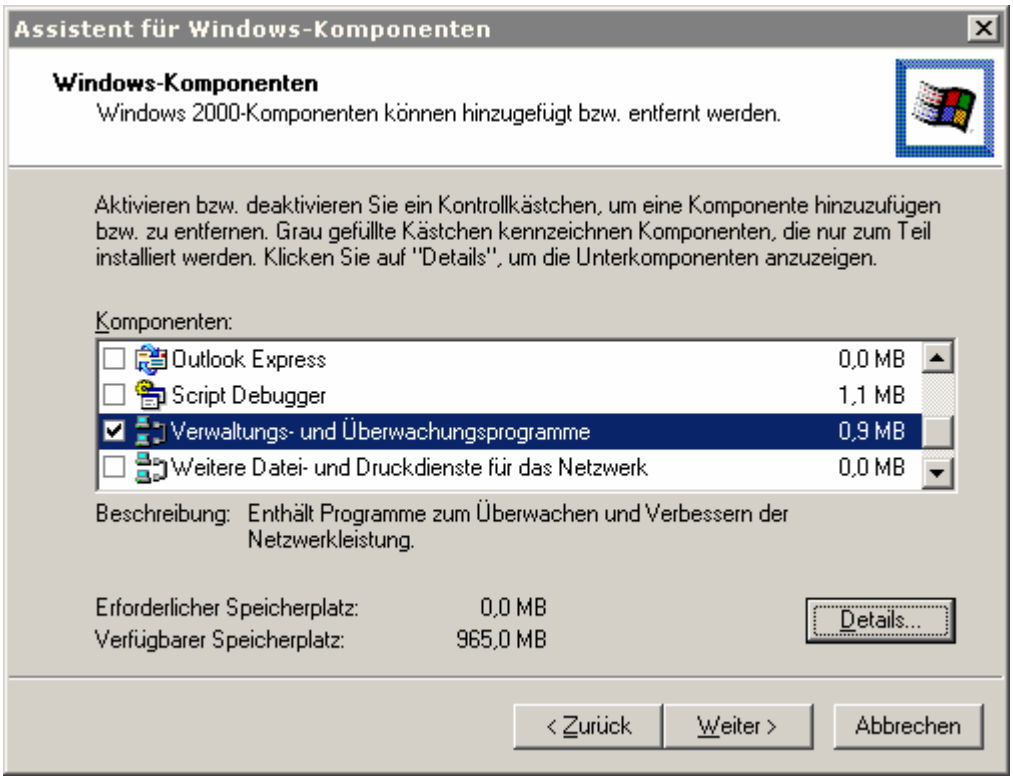

Abbildung 11: Assistent für Windows-Komponenten

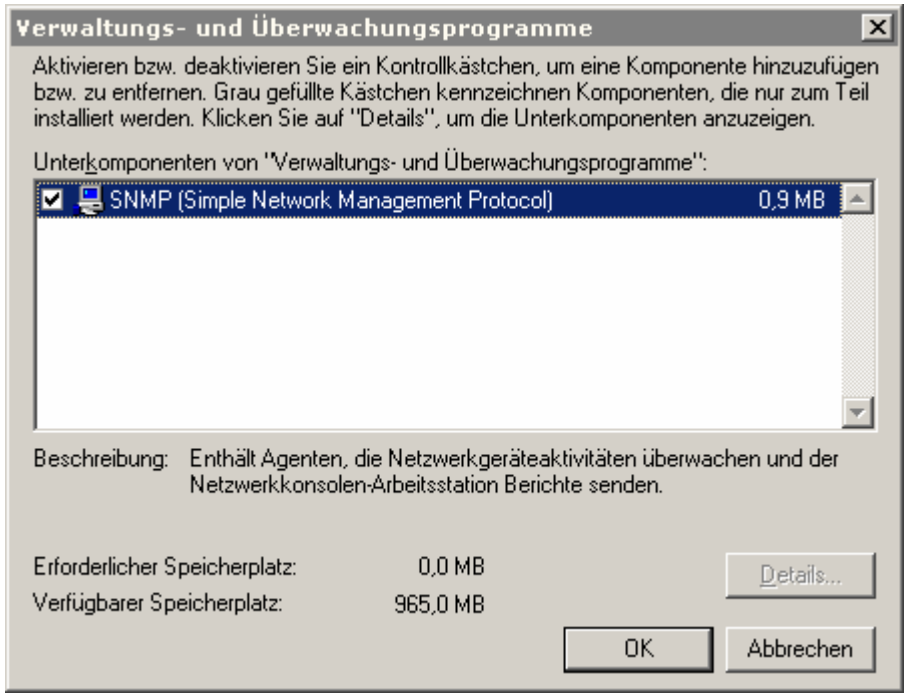

Abbildung 12: Untermenü „Verwaltungs- und Überwachungsprogramme“

2.) Danach wird über „*Systemsteuerung / Verwaltung / Dienst*“ überprüft, ob der „SNMP-Dienst“ gestartet ist. Über *Eigenschaften* wird im Register „*Sicherheit*“ der Name der Lese-Community eingetragen. Standardmäßig ist dies bereits „*public*“, was auch hier wieder beispielhaft verwendet wird, wobei diese im Praxiseinsatz aus Sicherheitsgründen besser anders genannt werden sollte. (Dies muss dann natürlich durchgehend auch in allen anderen Einstellungen übereinstimmen).

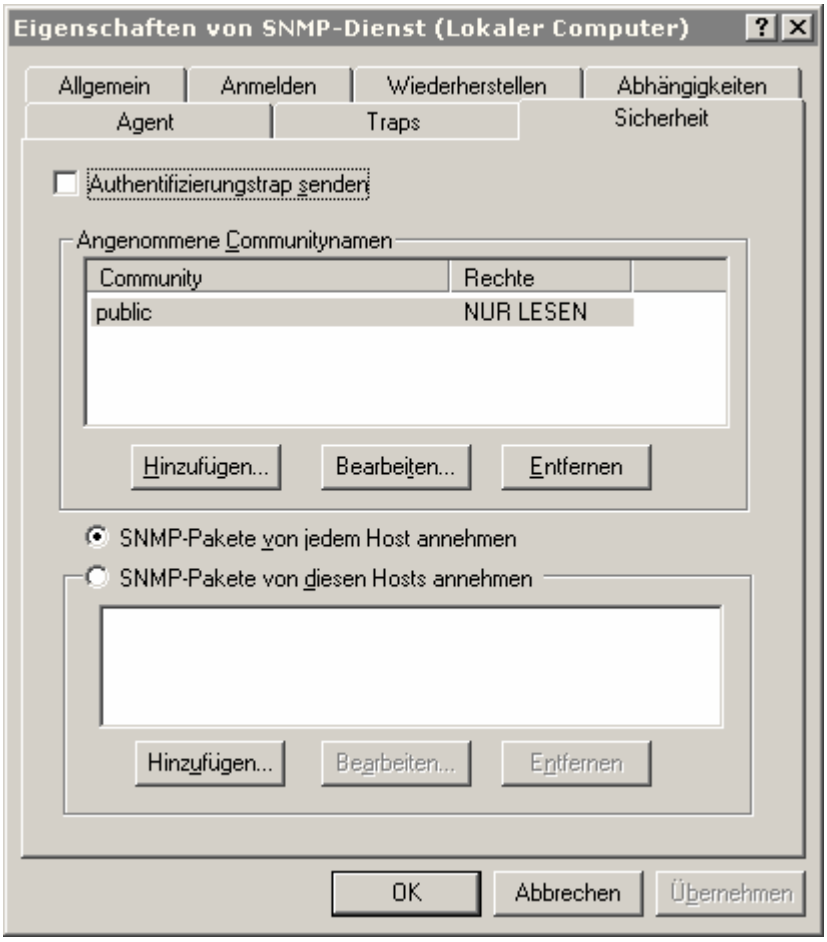

Abbildung 13: Eigenschaften SNMP-Dienst, Register „Sicherheit“

Das System ist nun grundsätzlich für SNMP vorbereitet, allerdings funktioniert bislang nur das Abfragen von Standard-MIB-II-Werten wie bspw. unter dem „Interfaces“-Zweig die maximal mögliche Geschwindigkeit der installierten Netzwerkkarte (100 Mbit).

Werte wie CPU-Auslastung, freier Speicher usw. sind standardmäßig nicht freigeschaltet. Hierzu ist normalerweise etwas Aufwand auf Kommandozeilenebene erforderlich und man benötigt das kostenpflichtige *Windows 2000 Resource Kit*[33] von Microsoft.

Eine einfachere und kostenfreie Lösung ist jedoch die folgende:

Unter der Adresse http://www.wtcs.org/snmp4tpc kann man sich das Tool *SNMP4W2K* herunterladen. Dieses beinhaltet bereits die notwendigen MIB-Einträge und integriert diese nach einem Aufruf der Installationsdatei „snmp4w2k-std.exe“ und einigen Bestätigungen im System.

[33] Vgl. http://www.microsoft.com/windows2000/techinfo/reskit/default.asp

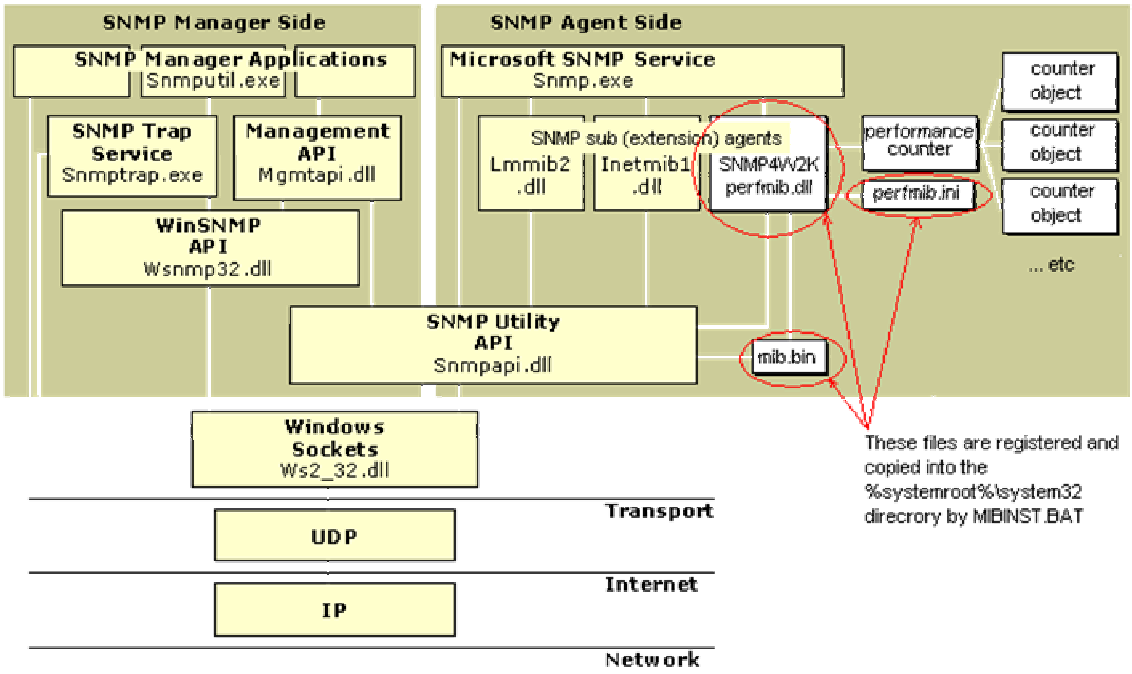

Abbildung 14: Integration und Funktionsweise von SNMP4W2K [34]

Ein letzter Schritt ist noch notwendig, wenn man später den freien Festplattenspeicher auslesen möchte. Dies erfordert die Eingabe des Windows-Befehls

```
diskperf -yv
```

Nach einem Systemneustart sind dann die sog. logischen Leistungsindikatoren aktiviert. (Der Befehl `diskperf -nv` macht das Ganze wieder rückgängig.)

Die Konfiguration des Windows2000-Systems - und damit die Möglichkeit zur Überwachung - ist an dieser Stelle nun abgeschlossen.

[34] Vgl. http://www.wtcs.org/snmp4tpc/snmp4w2k.htm

6.2 Überwachungs-System

Nach erfolgter Installation eines RedHat-Linux-Systems, bei dem während der Paketinstallation der (Apache-)Web-Server (httpd) und die Programmiersprache „Perl“ mitinstalliert wurde, war nun als nächster Schritt das Design und Grundgerüst für die Web-Oberfläche zu entwerfen.

6.2.1 Entwurf und Design der Web-Oberfläche

Die Web-Oberfläche besteht aus drei einfachen Frames[35], die ohne sichtbare Grenzen zusammengeführt wurden:

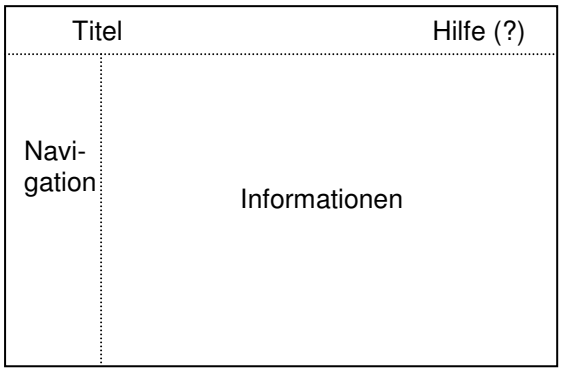

Abbildung 15: Skizze Web-Oberfläche

Der obere Frame (banner.html) ist für die Darstellung eines Titels und eines Hilfe-Buttons zuständig. Durch einen Mausklick auf das Hilfe-Symbol (?) öffnet sich per JavaScript ein kleines Hilfe-Fenster mit einer Kurzbeschreibung der Seite.

[35] eigenständige Bildschirmbereiche für verschiedene HTML-Dateien

Die benötigten JavaScript-Befehle zum Öffnen und Schließen des Hilfe-Fensters lauten:

```
<script language="JavaScript">
var hilfedatei="hilfe.html";
function showHelp()
{
 window.open(hilfedatei,'HELP','width=500,height=350,
             scrollbars=yes, resizable=no');
}
</script>

<a href="#" onClick="showHelp()">
  <img src="images/helpicon.gif" width="28" height="24" border="0">
</a>
```

```
<form>
  <input type="button" value="Schließen" onClick="window.close()">
</form>
```

Am linken Bildschirmrand der Web-Oberfläche befindet sich die Navigationsmöglichkeit (navigation.html) für die beiden Seiten „Online-Status“ und „Details“, die im Frame des Hauptbereichs dargestellt werden.

Die Seite „Online-Status“ wiederum besitzt einen Link „Fehlerprotokoll“ zur Anzeige des Datums und der Uhrzeit während der das jeweilige System nicht reagiert hat und bei der Seite „Details“ öffnen sich durch Anklicken der Diagramme ausführlichere Statistiken.

Durch Aufrufen der Datei index.html wird als Startpunkt der Online-Status (status.html) angezeigt. (Für Screenshots siehe Kapitel 7.)

HTML-Code der Datei „index.html“ zur Erstellung der Frames:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="Author" content="Marco Faisst">
  <meta name="description" content="Netzwerk-Fernueberwachung">
  <title>Netzwerk-Fern&uuml;berwachung</title>
</head>

<frameset rows="10%,90%" border=0 frameborder=0 framespacing=0>
  <frame src="banner.html" name="frame_oben" noresize scrolling=no>
  <frameset cols="15%,85%" border=0 frameborder=0 framespacing=0>
    <frame src="navigation.html" name="frame_links" noresize scrolling=no>
    <frame src="status.html" name="frame_rechts" noresize scrolling=yes>
  </frameset>
</frameset>

<body>
  <noframes>
    Der verwendete Browser kann keine Frames darstellen!
    Bitte einen anderen bzw. aktuelleren Browser verwenden.
  </noframes>
</body>

</html>
```

Die eigentliche Festlegung der Frames erfolgt innerhalb der ersten Anweisung `<frameset … >` und der letzten Anweisung `</fameset>` .

HTML-Code der Datei „status.html“ (Auszug):

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
   <meta HTTP-EQUIV="pragmas" CONTENT="no-cache">
   <meta http-equiv=refresh content="300; url=status.html">
 <title>Netzwerk-Fern&uuml;berwachung</title>
</head>
<body text="#FFFFFF" bgcolor="#003366" link="#99CCFF" vlink="#99CCFF"
alink="#99CCFF">

<table width="100%" border="0">

.....

<td><div align="center"><font face="Arial, Helvetica, sans-serif">
<img src="images/192.168.0.2_status.gif" alt="rot=Fehler, gr&uuml;n=alles
 ok" width="17" height="17"></font></div></td>
  <td><font face="Arial, Helvetica, sans-serif">192.168.0.2</font></td>
  <td><font face="Arial, Helvetica, sans-serif"> </font></td>
  <td><i><font face="Arial, Helvetica, sans-serif">Windows 2000 Prof.
   Arbeitsplatz #1</font></i></td>
  <td><font face="Arial, Helvetica, sans-serif">
  [ <a href="192.168.0.2_errorlog.txt" target="_blank">Fehlerprotokoll</a> ]
     </font></td>

.....
```

Zu Beginn wird durch `<meta http-equiv=refresh content="300; url=status.html">` die Seite status.html alle 5 Minuten aktualisiert.

Es wird der Text „Windows 2000 Prof. Arbeitsplatz #1“ ausgegeben mit einem Statuslämpchen als Grafik (`images/192.168.0.2_status.gif`) sowie die eingetragene IP-Adresse.

Ein Link „Fehlerprotokoll“ verweist auf die Datei `192.168.0.2_errorlog.txt`.
In Kapitel 6.2.5 wird im Rahmen der Integration eines CGI-Scripts nochmals kurz auf den HTML-Code der Web-Oberfläche eingegangen. Dort finden auch die ersten Zeilen wieder Verwendung.

Die vollständigen HTML-Dateien befinden sich auf dem beigelegten Datenträger zur Einsicht.

6.2.2 Scripte für das Menü „Online-Status“

Auf der Seite „Online-Status“ wurden vier Funktionen implementiert:

- die Anzeige eines grünen oder roten Lämpchen
- eine akustische Warnmeldung, wenn mindestens 1 Lämpchen rot ist [36]
- ein Eintrag im Fehlerprotokoll, wenn das betreffende Lämpchen rot ist

Folgendes Shell-Script erledigt diese Aufgaben:

```
#!/bin/csh

ln --force /home/snmp/test/up.gif /var/www/html/images/$1_status.gif
ping $1 -w 4 -q | grep " 0% loss"
if ($status == 1 ) then
  date >> /var/www/html/$1_errorlog.txt
  echo $1 antwortet nicht - Verbindung pruefen! >>
           [ gehört zur obiger Zeile ]          /var/www/html/$1_errorlog.txt
  echo ------------------------------------------------- >>
           [ gehört zur obiger Zeile ]          /var/www/html/$1_errorlog.txt
  ln --force /home/snmp/test/down.gif /var/www/html/images/$1_status.gif
  ln --force /home/snmp/test/alert.html /var/www/html/status.html
endif
```

Script: ipcheck

Unter Linux existieren verschiedene Shell-Sprachen, z.B. ist die bekannteste die „BASH Shell“ (*Bourne Again Shell*), eine freie Version der „Bourne Shell“ mit vielen zusätzlichen Funktionen. Die Bourne Shell war die erste UNIX Shell.

Für dieses Script wurde jedoch die „C Shell“ gewählt. Ein Vorteil ist, dass die Syntax der Programmiersprache C angelehnt ist.[37]

Die erste Zeile im Script (`#!/bin/csh`) gibt an, welche Shell verwendet wird. Das „csh“ steht für C Shell.

---

[36] Diese Funktion erfordert aus Kompatibilitätsgründen den Aufruf der Oberfläche mit dem Microsoft Internet Explorer.

[37] Vgl. http://www.linux-magazin.de/Artikel/ausgabe/1999/05/DrLinux/drlinux2.html

Das Script erwartet einen Parameter, welcher im Quelltext durch „`$1`“ Verwendung findet. Da IP-Adressen geprüft werden sollen, muss der Parameter beim Aufruf der Scripts also eine IP-Adresse, z.B. 192.168.0.1, sein.

Die zweite Zeile mit Anweisungen erstellt zunächst einen sog. „festen Link“ (Hard Link) auf eine Grafikdatei mit einem grünen Lämpchen (für „alles ok“) von einer GIF-Datei, dessen Name sich aus der IP-Adresse als Parameter sowie dem Anhang „`_status`“ zusammensetzt. Ist die Datei bereits vorhanden, wird sie (absichtlich) überschrieben. Die erstelle Linkdatei hat immer den gleichen Inhalt wie die Original-Zieldatei, stellt aber nur einen zusätzlichen Eintrag im Dateisystem dar und benötigt keinen zusätzlichen Speicherplatz.

Die nächste Zeile führt einen Ping[38]-Befehl auf die IP-Adresse aus und filtert dabei durch Verkettung mit dem Grep[39]-Befehl die Zeichenkette „ `0 % loss`“ (0% Verlust). Diese wird vom Ping-Befehl ausgegeben, wenn alle Ping-Versuche erfolgreich waren, sprich das Gerät noch erreichbar ist. Grep liefert einen Statuswert zurück. Er ist 0, wenn die Zeichenkette gefunden wurde und 1, wenn sie nicht gefunden wurde. Dieser Wert wird durch den `if`-Befehl verglichen. Falls der Status 1 ist, dann wird der Programmteil bis zum Befehl „`endif`“ ausgeführt. Falls er nicht 1 ist, wird dieser Teil ignoriert und direkt zu `endif` gesprungen. Da nach `endif` keine weiteren Anweisungen mehr folgen, wäre das Script beendet und die eingangs erstellte Grafik mit dem grünen Lämpchen korrekt, da die IP-Adresse ja erreichbar war.

Für den Fall dass der Status 1 ist, also die Zeichenkette irgendetwas anderes als „ 0 % loss“ ist, z.B. „ 25 % loss“, dann ist das betreffende Gerät nicht oder nicht mehr richtig erreichbar und somit wird der Teil zwischen `if` und `endif` ausgeführt.

Zunächst wird mit dem Befehl `date`, das aktuelle Datum und die Uhrzeit in eine Textdatei geschrieben um den Zeitpunkt der Nicht-Erreichbarkeit festzuhalten.

Dem wird eine Warnmeldung mit der IP-Adresse sowie eine Trennlinie zur Übersicht bei weiteren Einträgen angehängt.

Nun muss noch die eingangs grüne Lämpchengrafik durch eine warnende rote ersetzt werden. Dies erfolgt wieder mit dem ln[40]-Befehl und der Datei down.gif anstatt up.gif.

Der zweite ln-Befehl verlinkt noch eine spezielle Statusseite, die sich durch einen Signalton von der normalen Statusseite unterscheidet.

---

[38] Packet Internet Groper. Benutzt das Protokoll ICMP und sendet ein Paket an die IP-Adresse eines Rechners, um zu überprüfen ob dieser darauf reagiert.

[39] Zum Suchen und Filtern von beliebigen Zeichen(-folgen).

[40] ln = Link

Als Zielverzeichnis wurde /var/www/html/ gewählt, da sich dort standardmäßig der Apache-WebServer befindet (siehe nächstes Kapitel) und man somit die Möglichkeit hat, auf diese Informationen per Web zuzugreifen.

Das automatisierte Abfragen von IP-Adressen erledigt ein zweites Script:

```
#!/bin/csh

ln --force /home/snmp/test/noalert.html /var/www/html/status.html
/home/snmp/test/ipcheck 192.168.0.1
/home/snmp/test/ipcheck 192.168.0.2
/home/snmp/test/ipcheck 192.168.0.3
/home/snmp/test/ipcheck 192.168.0.9
/home/snmp/test/ipcheck 192.168.0.11
/home/snmp/test/ipcheck 192.168.0.123
```

Script: ipcheck_all

In dem ersten Script wurde beschrieben, dass eine IP-Adresse als Parameter erwartet wird. Das Script muss also so oft ausgeführt werden, wie zu prüfende Geräte vorhanden sind und natürlich muss hierzu auch die jeweils gültige IP-Adresse angegeben werden. Dazu wurden im zweiten Script alle vorhandenen IP-Adressen als Parameter für das erste Script eingetragen.

Beim Aufruf von `ipcheck_all` startet das Script `ipcheck` also insgesamt sechs mal.

Zu Beginn wird ähnlich dem ln-Befehl im ersten Script zunächst die Statusseite auf den Normalzustand gebracht, d.h. ohne akustische Warnmeldung. Diese wird dann ggf. durch die vorletzte Zeile des Scripts `ipcheck` aktiviert.

Zu beachten wäre, dass die beiden Scripte bislang nur gewöhnliche, nicht ausführbare Textdateien sind. Sie mussten daher also erst für den jeweiligen Benutzer (hier: snmp) ausführbar gemacht werden. Dies erreicht man mit dem Kommando:

```
chmod u+x snmp [Dateiname]
```

Dadurch werden die Rechte für den Benutzer (u = User) auf ausführbar (x = eXecutable) gesetzt.

Nachdem die Scripte ausführbar gemacht wurden, kann man mit dem Befehl „`./ipcheck_all`“ im aktuellen Verzeichnis bzw. hier absolut angegeben mit

„`/home/snmp/test/ipcheck_all`“ die eingetragenen IP-Adressen auf Erreichbarkeit überprüfen und die Ergebnisse später per Web-Oberfläche betrachten. Natürlich wäre es ziemlich sinnlos, dies von Zeit zu Zeit manuell durchzuführen. Das Script sollte also automatisch z.B. alle 5 Minuten die IP-Adressen testen.

Zum automatischen Ausführen von Programmen/Befehlen gibt es unter Linux sog. *Cronjobs*; das sind Aufgaben, die zu bestimmten Zeitpunkten von dem Dienst „*crond*“ (für cron daemon) abgearbeitet werden. Standardmäßig ist dieser Dienst bereits geladen (manuell als root[41]-Benutzer mit: „`service crond start`“) und muss nur noch konfiguriert werden.

Zunächst wurde eine Textdatei mit dem Namen *onlinestatus* angelegt, in der eine Zeile mit wie folgt formatiertem Inhalt existiert:

Minute	Stunde	Monat	Tag	Wochentag	Benutzer	Befehl

In vorliegenden Fall soll das Script nicht zu bestimmten Uhrzeiten, sondern alle 5 Minuten ausgeführt werden. Das bedeutet auf die Uhrzeit übertragen um 00:05 Uhr, um 00:10 Uhr, um 00:15 Uhr usw. In nicht zu berücksichtigende Felder wird ein * eingetragen. Die eingetragene Zeile muss wie folgt aussehen:

```
0 5 10 15 20 25 30 35 40 45 50 55 * * * * * snmp /home/snmp/test/ipcheck_all
```

Dies lässt sich jedoch auch noch weiter vereinfachen:

```
*/5 * * * * * snmp /home/snmp/test/ipcheck_all
```

Nun wurde als Benutzer *snmp* die Datei mit dem Befehl „`crontab onlinestatus`“ zu den Aufgaben hinzugefügt. Eventuell werden auf manchen Systemen root-Rechte benötigt, d.h. man muss als Benutzer *root* angemeldet sein und dann folgenden Befehl verwenden: „`crontab -u snmp onlinestatus`“. In beiden Fällen wird angenommen, dass sich die Datei *onlinestatus* im aktuellen Verzeichnis befindet.

Nach dem Ausführen von crontab, sollte sich im Verzeichnis /var/spool/cron eine neue Datei mit dem obigen Inhalt befinden.

[41] Der Benutzer unter Linux, der systemweit Schreib-/Leserechte hat.
Analog hierzu: Administrator-Rechte unter Microsoft Windows

Man kann die neue Datei übrigens auch direkt mit dem crontab-Befehl erstellen, jedoch sollte man hierzu möglichst Grundkenntnisse in *vi* besitzen; das ist der Editor, der durch den Befehl „ `crontab -e` “ automatisch gestartet wird. Um etwas zu schreiben muss man zunächst mit der Taste „ i “ in den Insert/Einfügen-Modus wechseln. Zum Sichern der Datei und zum Beenden ist die Tastenfolge „ Esc : x “ nötig.

6.2.3 Konfiguration des Apache-Webservers

Dieses Kapitel beschreibt die grundlegenden Einstellungen, die für den Betrieb des Apache-Webservers und damit der Web-Oberfläche erforderlich sind und behandelt zusätzlich auch einige Sicherheitsaspekte.

Aus Gründen der Übersichtlichkeit wurde die vollständige Konfigurationsdatei httpd.conf, die sich im Verzeichnis /etc/httpd/conf befindet, auf die beigelegte CD-ROM ausgelagert und hier werden nur die Passagen aufgeführt, die sich von der vorhandenen Standardkonfigurationsdatei unterscheiden oder hinzugefügt wurden.

6.2.3.1 Basiskonfiguration

Es wird angenommen, dass der Apache-Webserver, wie bereits in Kapitel 5 angedeutet, installiert ist. Ist dies nicht der Fall, kann bei Red Hat Linux das fehlende Paket „ httpd “
über den sog. Paket-Manager bzw. das grafische Setup-Programm zum Hinzufügen/Löschen von Software nachinstalliert werden. Bei ersterem muss man die Datei mit einer Bezeichnung ähnlich „ httpd*.rpm[42] “ auf den Installations-CDs suchen und mit dem Befehl „`rpm -i`“ installieren. Alternativ kann man sich den Webserver auch direkt von http://httpd.apache.org in der neusten Version downloaden. Allerdings wird hierauf nicht extra eingegangen, da es zu Unterschieden bei den Dateinamen und Verzeichnissen kommen kann.

Zunächst wurde die Datei /etc/httpd/conf/httpd.conf in einen Editor geladen. Ist eine grafische Benutzeroberfläche wie KDE[43] oder Gnome geladen, kann dies mit den dort vorhandenen Texteditoren erfolgen. In einem Terminal-Fenster, d.h. Kommandozeile/Shell eignet sich der Editor „pico“ ganz gut.

---

[42] rpm = Dateiformat für den [R]ed Hat [P]acket [M]anager
[43] K Desktop Environment

Ziemlich am Anfang der Datei httpd.conf befindet sich folgender Abschnitt:

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
Listen 8080
```

Hier wird die Port-Nr. für den Webserver angegeben. Standardmäßig ist dies für http immer der *Port 80*. Es sollte im Normalfall unterhalb der Kommentarzeilen (alle mit # am Anfang) also ein „Listen 80“ stehen. Eine Beschränkung auf bestimmte IP-Adressen wird nicht benötigt.

Falls Port 80 wie hier nicht zur Verfügung steht, weil er bspw. bereits durch ein anderes Programm mit Web-Oberfläche belegt ist, muss für eine einwandfreie Funktion ein anderer Port gewählt werden. Häufig verwendet man dann den leicht zu merkenden Port 8080. Im Web-Browser ist der Apache-Webserver dann später mit http:/192.168.0.9:8080 erreichbar. Diese IP-Adresse steht hier wieder stellvertretend für die tatsächliche IP-Adresse des Linux-Systems. Bei Port 80 kann die Angabe „:8080“ im Browser entfallen.

```
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your host, server-generated
# redirections will not work.  See also the UseCanonicalName directive.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
ServerName 192.168.0.9:8080
```

Im Abschnitt *ServerName* ist ein DNS-Name oder in diesem Fall die feste IP-Adresse des Linux-Systems und der Port anzugeben.

Wichtig ist weiterhin, dass die Standard-Ausgabeverzeichnisse immer auf „/var/www/html“ verweisen.

Angaben wie E-Mail-Adresse des Administrators, Standard-Sprache der Fehlermeldungen, etc. sind für den Betrieb nicht zwingend erforderlich, können zur Vervollständigung aber ebenfalls eingetragen werden.

6.2.3.2 Passwortgeschützter Zugriff

Damit auf die Web-Oberfläche keine unbefugten Personen zugreifen können, wird beim Aufruf der URL[44] (http://192.168.0.9:8080) ein Benutzername und Passwort abgefragt. Um diese Funktion zu implementieren, wurde zunächst wieder die Konfigurationsdatei httpd.conf bearbeitet.

Am Ende der Datei waren die folgenden Zeilen anzufügen:

```
<Directory /var/www/html>
 AllowOverride AuthConfig
 order allow,deny
 allow from all
</Directory>
```

Die erste Zeile gibt an, für welche Verzeichnisse der Zugriffsschutz erfolgen soll, in diesem Fall für „`/var/www/html/`“, also praktisch das komplette Verzeichnis mit den HTML-Dateien usw., auf das als erstes zugegriffen wird, da nach Aufruf der Adresse http://192.168.0.9:8080 der Webserver nach einer Datei namens index.html sucht, welche sich in diesem Verzeichnis befindet.

Als zweiter Schritt wurde in obigem Verzeichnis eine Datei mit dem Namen „**`.htaccess`**“ angelegt. (Den Punkt vor htaccess beachten!)

[44] Uniform Resource Locator, Protocol und Internet-Adresse

Inhalt der Datei .htaccess :

```
AuthName "Zugriff"
AuthType Basic
AuthUserFile /usr/local/apache/htpasswd.users
require valid-user
```

In der ersten Zeile wird zunächst eine beliebige Bezeichnung festgelegt, die später im Dialogfeld zur Authentifizierung erscheint, wenn http://192.168.0.9:8080 aufgerufen wird.

Zeile drei gibt ein Verzeichnis an, in dem die Passwörter und Benutzernamen für den Webserver gespeichert werden sollen.

Die letzte Zeile besagt schließlich, dass nur gültige, d.h. die in der Datei gespeicherten Benutzer später Zugriff erhalten.

Der oder die Benutzer müssen hierfür natürlich erst einmal angelegt werden. Dies wird mit dem Befehl „htpasswd“ erreicht:

```
htpasswd -c /usr/local/apache/htpasswd.users admin
```

Der Pfad muss mit dem in der Datei .htaccess übereinstimmen. Als Benutzername wurde „admin“ verwendet. Der Parameter „-c“ legt eine neue Datei an. Nach dem Ausführen des Befehls wird man aufgefordert, ein Passwort zu wählen.

Möchte man weitere autorisierte Benutzer hinzufügen, so ist der Parameter „-c“ wegzulassen und ein anderer Benutzername anzugeben.

6.2.3.3 Unterstützung für CGI-Scripte

Da in Kapitel 6.2.5 ein CGI-Script verwendet wird, muss der Webserver noch für das Ausführen von CGI-Scripts konfiguriert werden.

Dies ermöglicht folgender Eintrag in der Datei httpd.conf:

```
<Directory /var/www/cgi-bin>
 AllowOverride AuthConfig
 order allow,deny
 allow from all
 Options ExecCGI
</Directory>
```

6.2.3.4 Abschließende Konfiguration

An diesem Punkt angelegt, ist der Apache-Webserver mit den HTML-Dateien für die Web-Oberfläche im Verzeichnis /var/www/html fertig konfiguriert und es werden nur noch Daten für die Detail-Seite mit den Diagrammen benötigt.

Der Webserver sollte auf keinen Fall mit angemeldetem root-User laufen - dies würde nicht nur einen Sicherheitsmangel darstellen, sondern funktioniert in den meisten Fällen auch gar nicht erst. Standardmäßig wird der Benutzer und die Gruppe „apache“ verwendet. Falls das Linuxsystem mit dem Webserver nicht noch anderweitig zum Arbeiten verwendet wird, sollte kein Benutzer angemeldet sein und auch eine grafische Oberfläche ist nicht notwendig, denn die Dienste funktionieren auch so.

Zu berücksichtigen wäre noch, dass der Benutzer snmp im Verzeichnis /var/www/html Schreib- und Leserechte hat und die bereits vorhandenen Dateien ebenfalls für diesen Benutzer schreib- und lesbar sind. Falls das nicht der Fall ist, z.B. weil die Scripte schon einmal als root-User ausgeführt wurden, muss man die vorhandenen Dateien entweder vorher löschen oder mit dem Befehl „ `chown`[45] `snmp:snmp [Dateiname]` “ dem Benutzer und der Gruppe „snmp“ übergeben. Ggf. sind die Dateirechte noch mit „`chmod u+rw`“ anzupassen.

[45] chown = change owner, Dateieigentümer wechseln

Gestartet wird der Webserver mit dem Befehl

```
/sbin/service httpd start
```

Im Normalfall wird der Apache-Webserver (genauer: der Dienst httpd) bei jedem Systemstart automatisch geladen.

Andernfalls kann er im Verzeichnis /etc/xinetd.d als neuer Dienst hinzugefügt werden.

Erreichbar ist die Web-Oberfläche wie bereits mehrfach erwähnt unter

http://192.168.0.9:8080

Außerdem ist, falls gewünscht, auch eine sichere Verbindung (SSL) möglich:

https://192.168.0.9 (hierbei wird automatisch der Port 443 für SSL verwendet)

Damit sie auch außerhalb des Intranets über das Internet erreichbar ist, muss anstelle der obigen lokalen IP-Adresse die externe IP-Adresse angegeben werden, mit der die Rechner im Internet angebunden sind.

Der Systemadministrator muss hierzu noch im Hauptnetzwerkrouter ein sog. Portforwarding[46] für den verwendeten http-Port (hier 8080, standardmäßig 80 bzw. 443 für SSL) auf die lokale IP-Adresse 192.168.0.9 einrichten, damit die externen Anfragen den richtigen Rechner mit der Web-Oberfläche, d.h. mit dem Webserver, erreichen.

[46] zur Weiterleitung von Portanfragen an eine bestimmte IP-Adresse

6.2.4 MRTG und RRDTool

MRTG bedeutet ausgeschrieben „Multi Router Traffic Grapher“ und stammt von Tobias Oetiker und Dave Rand. Seine Hauptaufgabe ist es den Traffic[47] von Routern oder Switches grafisch darzustellen. Jedoch ist das Tool längst nicht nur darauf beschränkt, denn dadurch dass es von SNMP Gebrauch macht, kann nahezu jeder beliebige über SNMP abfragbare Wert in grafischer Form aufbereitet werden.

Hierfür sammelt MRTG in regelmäßigen Abständen Daten von den SNMP-Geräten aus denen dann mit dem Hilfstool rateup Grafiken im Format *.PNG[48] erstellt werden. Die Grafiken werden außerdem in ebenfalls automatisch generierte HTML-Seiten eingebunden.

MRTG und seine beiden Konfigurationsscripte sind in der Sprache Perl geschrieben. Rateup ist aus Geschwindigkeitsgründen eine in C kompilierte Binärdatei.

Da die Grafiken in einem festgelegten Intervall ständig erzeugt werden, auch wenn man sie nur selten ansehen möchte, wurde rateup in dieser Arbeit durch rrdtool (ebenfalls von Tobias Oetiker) das lediglich die reinen, nicht grafisch aufbereiteten Daten sammelt und in Dateien schreibt.

Das Erstellen der HTML-Seiten wurde einem CGI-Script überlassen, daher ist hier die einzige Aufgabe von MRTG, nur in regelmäßigen Abständen Daten zu sammeln, während die Aufbereitung dieser Daten dann in den nächsten Kapiteln behandelt wird. Das Format der Konfigurationsdatei von MRTG ist dort größtenteils wiederverwendbar.

Von www.mrtg.org wurde eine aktuelle Version von MRTG heruntergeladen. Der Dateiname für die Linux-Variante lautete zum Stand dieser Arbeit `mrtg-2.10.0.tar.gz`[49] und die Versionsnummer erhöht sich entsprechend bei aktuelleren Versionen. Mit dem Kommando „`tar xzf mrtg-2.10.0.tar.gz`“ wurde das Archiv in ein vorher ausgewähltes Verzeichnis entpackt. Danach wurde in dem nun vorhandenen Verzeichnis der Befehl

```
./configure --prefix=/usr/local/mrtg-2
```

ausgeführt, gefolgt von „`make`“ und „`make install`“. Dadurch wird MRTG im System installiert.

---

47 Umgangsprachlicher, engl. Begriff für „Netzwerkverkehr“
48 Portable Network Graphic, plattformunabhängiges, lizenzfreies Grafikformat
49 Das Standard-Archivformat von Unix/Linux. Kompression mit GNU Zip.

Anschließend sollten sich in einem Unterverzeichnis „bin“ drei Scripte befinden, nämlich mrtg, cfgmaker und indexmaker. Mit cfgmaker kann eine Konfigurationsdatei als „Grundgerüst“ für die spätere Bearbeitung erstellt werden.

Dies erreicht man mit dem Befehl

```
./cfgmaker public@192.168.0.1 > mrtg.cfg
```

wobei public die SNMP-Community und 192.168.0.1 die IP-Adresse eines SNMP-Geräts ist. Durch das Zeichen „>“ wird die automatisch erstellte Konfiguration in eine neue Datei namens mrtg.cfg geschrieben. Der Befehl kann um weitere IP-Adressen/Geräte und Kommandos erweitert werden.

Standardmäßig werden die Geräte auf Netzwerktraffic überwacht, vorausgesetzt es handelt sich um Geräte, die diese SNMP-Funktion auch bereitstellen, also z.B. Router, Switche, Netzwerkkarten.

Die nun vorhandene Datei „mrtg.cfg“ wurde in einen Texteditor geladen.

Am Anfang der Datei wurde

```
Workdir: /var/www/html/mrtg
```

gewählt sowie die folgenden Einträge hinzugefügt:

```
RunAsDaemon: yes
Interval: 5
Logformat: rrdtool
PathAdd: /usr/local/rrdtool-1.0.45/bin/
LibAdd: /usr/local/rrdtool-1.0.45/lib/perl/
```

Die ersten beiden Einträge bewirken, dass MRTG später auch wirklich permanent im Arbeitsspeicher verbleibt und im Hintergrund alle 5 Minuten Daten sammelt.
Der Eintrag „`Logformat: rrdtool`“ ersetzt das Tool rateup, das normalerweise die Grafiken erzeugt durch rrdtool, was zur Folge hat, das nun keine Grafiken mehr erzeugt werden und auch keinerlei HTML-Seiten mehr erstellt werden.

RRDTool ist auch über www.mrtg.org erhältlich und liegt als Archiv namens „rrdtool-1.0.45.tar.gz“ vor. Wie in obigem Kasten ersichtlich ist, wurde als Installationsverzeichnis /usr/local/rrdtool-1.0.45 gewählt.

Die durch das cfgmaker-Script automatisch erzeugten Einträge beginnend mit „### Interface 1 ...“ müssen normalerweise nicht mehr geändert werden, man kann sie jedoch noch etwas anpassen.

Die folgenden beiden Zeilen

```
Options[192.168.0.1_1]: nobanner
Background[192.168.0.1_1]: #336699
```

bspw. bewirken, dass kein MRTG-Logo angezeigt werden soll (nobanner) und die Hintergrundfarbe für die dann erzeugten HTML-Seiten auf das gleiche Dunkelblau gesetzt wird wie bei der Web-Oberfläche.

Da wie bereits erwähnt, aber keine Grafiken und HTML-Dateien von MRTG erzeugt werden, sind diese Angaben ebenso wie die nächste erst später für das CGI-Script von Bedeutung, das die selbe Datei mrtg.cfg verwendet.

Die Zeile

```
Target[192.168.0.1_1]: 1:public@192.168.0.1
```

bleibt unverändert, sei aber wegen ihrer Wichtigkeit kurz erwähnt. „public“ und „192.168.0.1“ wurden automatisch durch den Aufruf des cfgmaker-Befehls eingetragen. Die Ziffer 1 bedeutet, dass in diesem Abschnitt der eingehende und ausgehende Netzwerkverkehr für das erste Interface (den ersten RJ45-Port[50]) gemessen werden soll und ist eine Abkürzung für die erste Instanz der Object ID’s, die in Textform „ifInOctets“ und „ifOutOctets“ lauten würden.

Die Abfrage der CPU-Auslastung etc. des Windows2000-Rechners muss manuell in die Konfigurationsdatei mrtg.cfg eingetragen werden. (Nachfolgend wird als Beispiel die CPU-Auslastung verwendet. Bei den anderen Werten ist der Ablauf ähnlich.)

[50] hier: Standard-Ethernetbuchse

Zunächst muss man wissen, welche Object ID die SNMP-Variable für die CPU-Auslastung hat, da MRTG selbst nur die für den Traffic kennt.

Entweder durchsucht man die MIB-Zweige manuell nach den relevanten OID's. Hierzu eignet sich unter Linux der Befehl `snmpwalk` und unter Windows das Tool „GetIf"[51] :

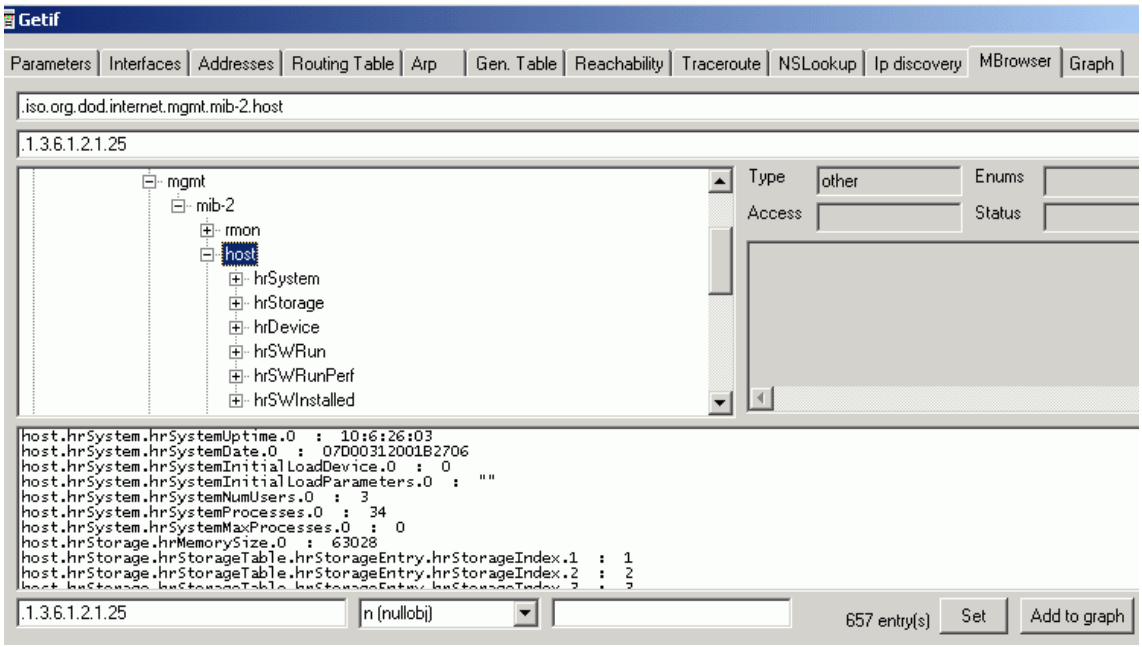

Abbildung 16: GetIf

Oder man sucht nach den Einträgen im Internet, in der Hoffnung, dass sie irgendwo beschrieben sind.

Wenn man die zweite Methode wählt, hat sich als gute Anlaufstelle hierfür die WebSite www.wtcs.org/snmp4tpc erwiesen. Unter dem Menüpunkt *Management / MRTG* finden sich im Windows2000-Abschnitt schon einige vorgefertigte Konfigurationsdateien (für ein auf MRTG basierendes kostenpflichtiges Überwachungssystem namens „Net Sonar" / hier nicht weiter von Bedeutung), denen man die OID entnehmen kann.

Die erste Datei WIN2KFS1_CPU.CFG enthält die benötigten Informationen zwischen

„`Target[WIN2KFS1_PRIV_USER]`" und „`LegendO[WIN2KFS1_PRIV_USER]`".

Der Rest ist nicht von Interesse. Dieser Abschnitt wurde also zunächst der Datei mrtg.cfg hinzugefügt. Die zuständige Object ID (die eigentliche, wichtige Angabe) ist in der Zeile beginnend mit „Target" eingetragen. Die Angaben in den eckigen Klammern sind immer eine beliebige Zeichenkette zur Identifikation der jeweiligen Abfrage (hier

[51] http://www.wtcs.org/snmp4tpc/getif.htm

CPU-Auslastung) und müssen für verschiedene Abfragen auch unterschiedlich benannt sein. Sie können frei gewählt werden.

Am Ende der Target-Zeile wurde noch die Community und IP-Adresse des Windows2000-Rechners angepasst. Nach einigen weiteren kleinen optischen Veränderung, sieht der relevante Abschnitt wie folgt aus:

```
#################################################################
# CPU-Auslastung Windows 2000
#################################################################

PageTop[192.168.0.2_CPU]: <font face="Arial, Helvetica, sans-serif"><h1>aktuelle CPU-
 Auslastung</h1></font>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Windows 2000 Professional</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.0.2</TD></TR>
 </TABLE>
Background[192.168.0.2_CPU]: #336699
Target[192.168.0.2_CPU]: 1.3.6.1.4.1.311.1.1.3.1.1.2.1.5.1.48&
  [Fortsetzung von obiger Zeile]  .1.3.6.1.4.1.311.1.1.3.1.1.2.1.4.1.48:public@192.168.0.2
MaxBytes[192.168.0.2_CPU]: 50
AbsMax[192.168.0.2_CPU]: 100
Title[192.168.0.2_CPU]: Durchschnittliche CPU-Auslastung in % (System -- Anwendungen)
Options[192.168.0.2_CPU]: gauge, nopercent, nobanner
YLegend[192.168.0.2_CPU]: CPU-Auslastung in %
ShortLegend[192.168.0.2_CPU]: %
Legend1[192.168.0.2_CPU]: CPU-Auslastung -System- in Prozent
Legend2[192.168.0.2_CPU]: CPU-Auslastung -Anwendungen- in Prozent
LegendI[192.168.0.2_CPU]: Sys.CPU:
LegendO[192.168.0.2_CPU]: Anw.CPU:
```

Die meisten Angaben sind für die Beschriftung der Diagramme und HTML-Ausgaben zuständig.

```
AbsMax[192.168.0.2_CPU]: 100
```

Dies bedeutet, dass der absolute, d.h. höchste Wert, den das Diagramm anzeigen kann, 100 ist, da die CPU-Auslastung in Prozent angeben wird und bei 100% das System unter Voll-Last laufen würde.

Es werden zwei Werte gemessen und angezeigt: die Auslastung, die das Windows-System selbst verursacht und die Auslastung durch Programme, die Benutzer gestartet

haben. Daher sind in der Target-Zeile auch zwei Object ID's eingetragen, die durch das Zeichen „&“ voneinander getrennt sind. Soll nur ein Zustand gemessen werden, muss man zwei mal die selbe Object ID eintragen. MRTG benötigt immer genau zwei Werte zur Verarbeitung, nicht mehr und nicht weniger.

```
Options[192.168.0.2_CPU]: gauge, nopercent
```

Hiermit wird festgelegt, dass nur der aktuelle Wert und keine Durchschnittswerte gemessen werden und die Legende keine prozentualen Angaben zeigen soll.

Nachdem mit den restlichen gewünschten Object ID's für den freien Arbeitsspeicher usw. ähnlich verfahren wurde, kann MRTG bereits gestartet werden und Daten einholen. Der Start erfolgt mit durch folgendes Kommando (ggf. mit Pfadangaben zu ergänzen):

```
./mrtg mrtg.cfg
```

Möglicherweise erhält man eine Fehlermeldung, dass das der verwendete Zeichensatz der Kommandozeile Probleme verursachen könnte. In diesem Fall muss der Aufruf lauten:

```
env LANG=C ./mrtg mrtg.cfg
```

Der Prozess wird automatisch in den Hintergrund übergeben und man kann gleich wieder Befehle eintippen.

Es empfiehlt sich, MRTG automatisch beim Systemstart zu laden, auch wenn das System zur Überwachung sowieso permanent laufen muss. Hierzu kann ein Startscript im Verzeichnis /etc/xinetd.d angelegt werden.

6.2.5 Integration des CGI-Scripts „14all.cgi“

Beim Aufruf der Detail-Seite der Web-Oberfläche müssen aus den gesammelten Daten noch aktuelle Diagramme erstellt werden und die HTML-Seiten darauf vorbereitet werden.

Die Erstellung der Grafiken und HTML-Statusseiten übernimmt ein auf MRTG abgestimmtes CGI-Script. Es existieren mehrere Lösungen; ein Script das diese Aufgabe einfach und zuverlässig erledigt, trägt die Bezeichnung „ *14all.cgi* “ und ist unter der Adresse http://my14all.sourceforge.net zu beziehen.

Nach dem Download der Datei 14all-1.1.txt wurde diese in 14all.cgi umbenannt und ins CGI-Verzeichnis des Webservers (/var/www/cgi-bin) kopiert.

Danach wurde diese ebenfalls in einem Editor bearbeitet. Gleich am Anfang sollte die Zeile

```
use lib qw(/usr/local/mrtg-2/lib/mrtg2);
```

aktiv sein (kein #-Kommentarzeichen davor). Sie gibt an, wo sich die MRTG-Libaries (Hilfsdateien) befinden. Nach der Installation von MRTG sollte es das oben angegebene Verzeichnis sein.

Wichtig ist noch die Angabe, wo sich die im letzten Kapitel erstelle Konfigurationsdatei „mrtg.cfg“ befindet. Dies wird durch

```
$cfgfile = '/usr/local/mrtg-2/mrtg.cfg';
```

festgelegt.

In diesem Fall wurde sie in /usr/local/mrtg-2 abgelegt.

Beim Aufruf der Detail-Seite der Web-Oberfläche soll zunächst eine Übersichtseite mit den Tagesdiagrammen angezeigt werden. Hierzu wurde eine Seite namens details.html erstellt, die Grafik-Tags[52] des folgenden Formats besitzt:

```
<img src="cgi-bin/14all.cgi?log=192.168.0.2_memory&png=daily.s&small=1">
```

[52] HTML-Befehlsmarken: <...> </...>

Wenn der Web-Browser die Grafiken anzeigen will, wird durch den Verweis auf das CGI-Script dieses gestartet und die Grafik in Echtzeit erstellt.

Den Grafiken wurde auch ein spezieller Hyperlink[53] hinterlegt:

```
<a href="cgi-bin/14all.cgi?log=192.168.0.2_memory">
```

Durch einen Mausklick auf eine Grafik erzeugt das Script ebenfalls in Echtzeit eine HTML-Seite wie sie in der Datei mrtg.cfg definiert wurde, mit Statistiken und Diagrammen für Tag, Woche, Monat, Jahr.

Sollte die Übersichtsseite details.html längere Zeit geöffnet sein, werden ihre Grafiken normalerweise nicht automatisch aktualisiert. Damit dies geschieht, wurden im Header[54] der HTML-Datei noch folgende Meta-Tags gesetzt, die bewirken, dass die Seite alle 300 Sekunden, also alle 5 Minuten, aktualisiert wird und kein Cache verwendet wird, der möglicherweise veraltete Seiten anzeigen würde:

```
<meta HTTP-EQUIV="Refresh" CONTENT="300; url=details.html">
<meta HTTP-EQUIV="Cache-Control" content="no-cache">
<meta HTTP-EQUIV="Pragma" CONTENT="no-cache">
```

[53] Verknüpfung auf externe Informationen
[54] Kopfzeile / Anfang einer Datei

6.2.6 Das Konfigurationsscript „statcfg“

Mit dem Script „statcfg“ kann die Online-Status-Seite (status.html) der Web-Oberfläche automatisch mit weiteren Einträgen ergänzt werden, ohne dass diese manuell mit einem Editor bearbeiten werden muss.

Das Layout kann später natürlich trotzdem noch mit einem HTML- bzw. Text-Editor individuell optimiert und nachbearbeitet werden, falls erforderlich.

Das Script erwartet zwei Parameter: eine IP-Adresse und eine Kurzbeschreibung.
Wurde kein Parameter angegeben, wird folgende Ausgabe angezeigt:

```
Konfigurationsprogramm fuer Onlinestatus-Seite - Netzwerkueberwachung
-----------------------------------------------------------------------


Aufruf:   statcfg [IP-Adresse] [Beschreibung_ohne_Leerzeichen]

Beispiel: statcfg 192.168.1.2 Switch_3_Erdgeschoss
```

Wurden die erforderlichen Parameter angegeben, so wird dem Script „ipcheck_all“ ein Eintrag hinzugefügt und eine Datei status_neu.html erzeugt bzw. aktualisiert. Eine bereits vorhandene Datei status.html kann dann gesichert werden und die Datei status_neu.html nach Umbenennen in status.html für die Web-Oberfläche verwendet werden.

Im Gegensatz zu dem Script „ipcheck“ sind „statcfg“ und das nachfolgende Script „graphcfg“ für die Bash-Shell (#!/bin/bash) geschrieben. Sie befinden sich ebenfalls auf der beigelegten CD-ROM.

6.2.7 Das Konfigurationsscript „graphcfg“

Damit bei späteren Diagramm-Ergänzungen die Datei mrtg.cfg (siehe Kap. 6.2.4) nicht immer von Hand editiert werden muss, wurde für die wichtigsten Werte ein weiteres Script geschrieben, welches dies automatisiert.

Dabei werden außerdem die dazugehörigen HTML-Einträge für die Details-Übersichtsseite der Web-Oberfläche in einer Datei details_neu.html abgelegt, welche nach Umbenennen in details.html ähnlich wie bei dem vorherigen Script direkt verwendbar ist. Ggf. kann sie auch noch manuell angepasst werden, falls gewünscht.

Nachfolgend die Hilfe-Ausgabe des Scripts, die beim Aufruf ohne oder mit falschem Parameter erscheint:

```
Konfigurationsprogramm fuer Detailseite - Netzwerkueberwachung
---------------------------------------------------------------


Aufruf:  graphcfg [IP-Adresse] [Community] [Ueberwachungsfunktion]


folgende Funktionen stehen zur Verfuegung:
traffic, cpu, hd, ram

Beispiel:  graphcfg 192.168.0.1 public cpu
```

Das Script muss zuvor wieder über den Befehl `chmod u+x snmp graphcfg` für den Benutzer „snmp“ ausführbar gemacht werden und wird mit `./graphcfg` gestartet.
(*Analog verhält es sich für das vorherige Script statcfg.*)

Um bspw. die Datei mrtg.cfg mit den Konfigurationsanweisungen für den freien Arbeitsspeicher eines System zu ergänzen (bzw. zu erstellen, falls noch nicht vorhanden), ist folgender Aufruf notwendig:

```
./graphcfg 192.168.0.23 public ram
```

Als IP-Adresse des neuen Systems wird im obigen Beispiel 192.168.0.23 verwendet und als Community die bekannte Standard-Community „public“.

`ram` steht für den Arbeitsspeicher, `hd` für die Festplatte/Laufwerk C: (Hard Disk), `cpu` für die CPU-Auslastung und `traffic` für den Netzwerk-Traffic.

7 Beschreibung und Bedienung der Web-Oberfläche

Nachdem die Installation wie in den vorherigen Kapiteln beschrieben durchgeführt wurde, ist die Web-Oberfläche nun mit einem Web-Browser wie z.B. Microsoft Internet Explorer, Netscape, Mozilla usw. erreichbar.

Hierzu wird in das Adressfeld die IP-Adresse des Überwachungssystems und eine evt. Portnummer eingetragen:

http://192.168.0.9:8080
im lokalen Netzwerk (Intranet) und

http://[vom Provider[55] vergebene IP-Adresse]:8080
im Internet von entfernten Rechnern außerhalb des lokalen Netzwerkes

Nach dem Aufruf erscheint zunächst ein Dialogfeld mit der Aufforderung, den Benutzername und das Passwort einzugeben:

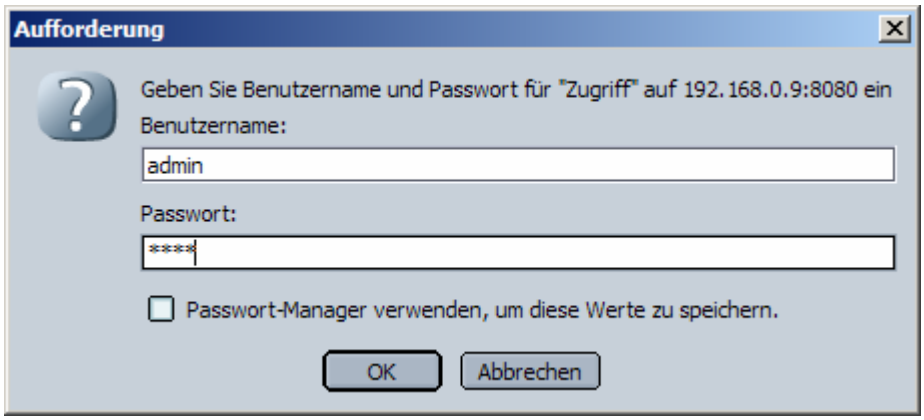

Abbildung 17: Passwortabfrage

Existiert der eingegebene Benutzername und ist das Passwort korrekt, so erhält man Zugriff auf die Web-Oberfläche und als Startseite erscheint der Online-Status der Geräte.

[55] Anbieter eines Internet-Zugangs, z.B. T-Online

Abbildung 18: Hauptseite / Online-Status

Ein grünes Lämpchen bedeutet, dass mit dem Gerät der aufgeführten IP-Adresse alles in Ordnung ist. Rot dagegen bedeutet, dass es im Netzwerk nicht erreichbar ist, weil z.B. ein Netzwerkkabel entfernt wurde oder das System abgestürzt ist.

Durch einen Mausklick auf „Fehlerprotokoll“ kann das Datum und der Zeitpunkt eingesehen werden, zu dem das Problem aufgetreten ist; hier als Beispiel für die IP-Adresse 192.168.0.11 :

```
Die Sep  9 13:42:15 CEST 2003
192.168.0.11 antwortet nicht - Verbindung pruefen!
-------------------------------------------------
Die Sep  9 13:42:39 CEST 2003
192.168.0.11 antwortet nicht - Verbindung pruefen!
-------------------------------------------------
Die Sep  9 13:44:54 CEST 2003
192.168.0.11 antwortet nicht - Verbindung pruefen!
-------------------------------------------------
Die Sep  9 13:49:52 CEST 2003
192.168.0.11 antwortet nicht - Verbindung pruefen!
-------------------------------------------------
```

Abbildung 19: Fehlerprotokoll

Das Hilfe-Symbol (?) am rechten oberen Bildschirmrand öffnet ein kleines Hilfe-Fenster mit einer Kurzbeschreibung der Seite:

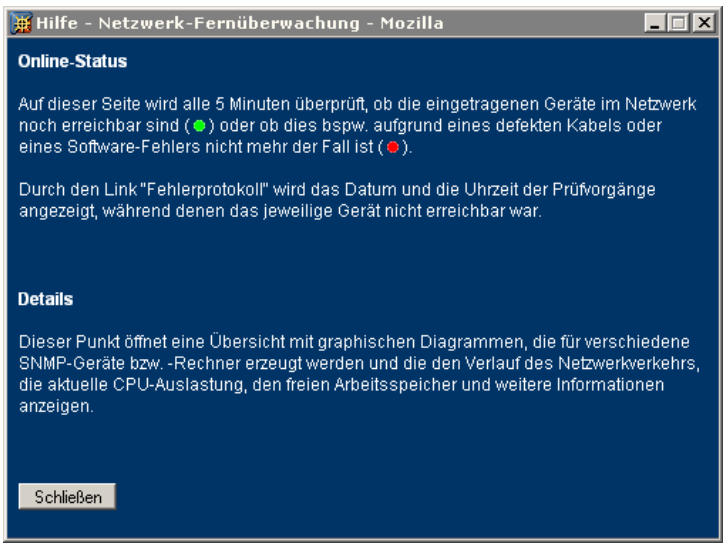

Abbildung 20: Hilfe-Fenster

Durch die Auswahl „Details“ im Ansichten-Navigationsfeld wird die Seite mit Diagrammen für Netzwerk-Traffic, CPU-Auslastung, usw. geöffnet und die Diagramme in Echtzeit aus den gesammelten Daten erstellt:

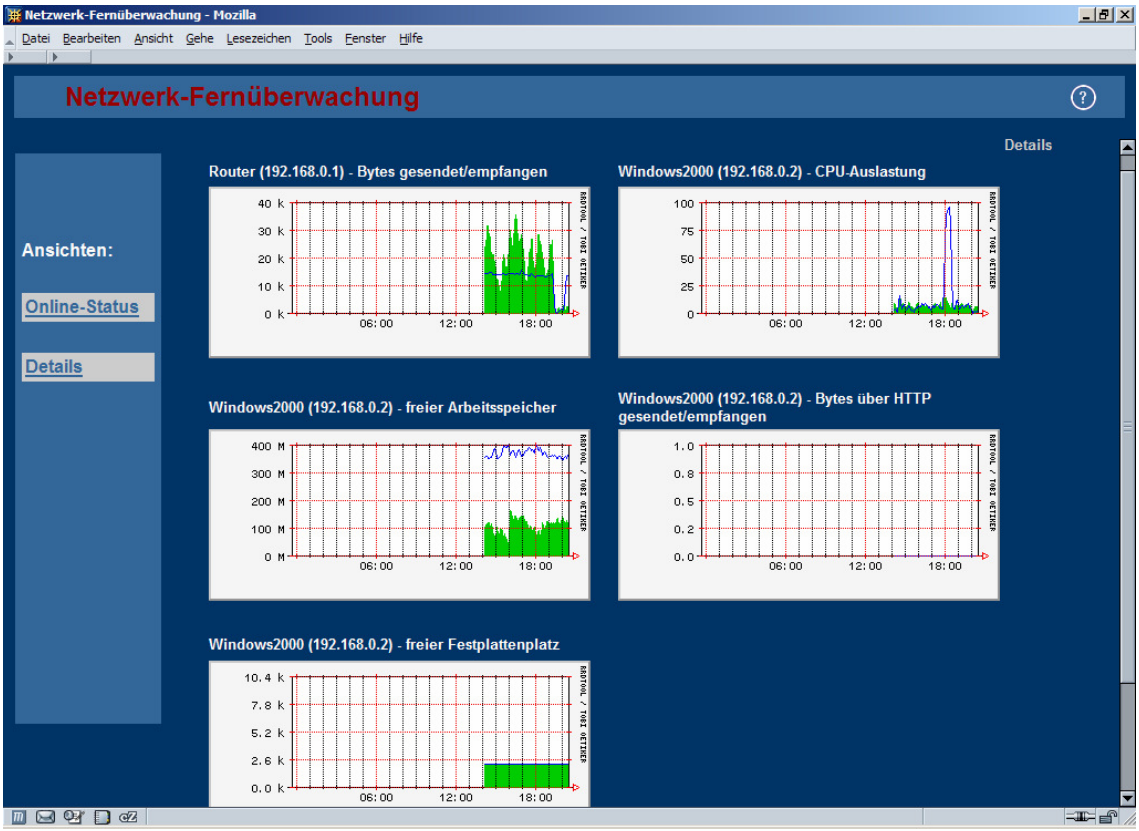

Abbildung 21: Detailansicht

Wenn man jetzt wiederum auf eines dieser Diagramme klickt (hier als Beispiel das erste), so öffnet sich für das jeweilige Diagramm eine ausführlichere Ansicht, die neben zusätzlichen beschreibenden Informationen und dem schon bekannten Tagesdiagramm weitere Diagramme anzeigt.

Dies sind die Diagramme für Woche, Monat und Jahr. Es kann also auch noch über einen längeren Zeitraum hinweg kontrolliert werden, wie sich die Diagramme entwickelt haben. Außerdem werden in den Diagrammgrafiken auch Durchschnittswerte berechnet, z.B. der durchschnittliche tägliche Netzwerktraffic:

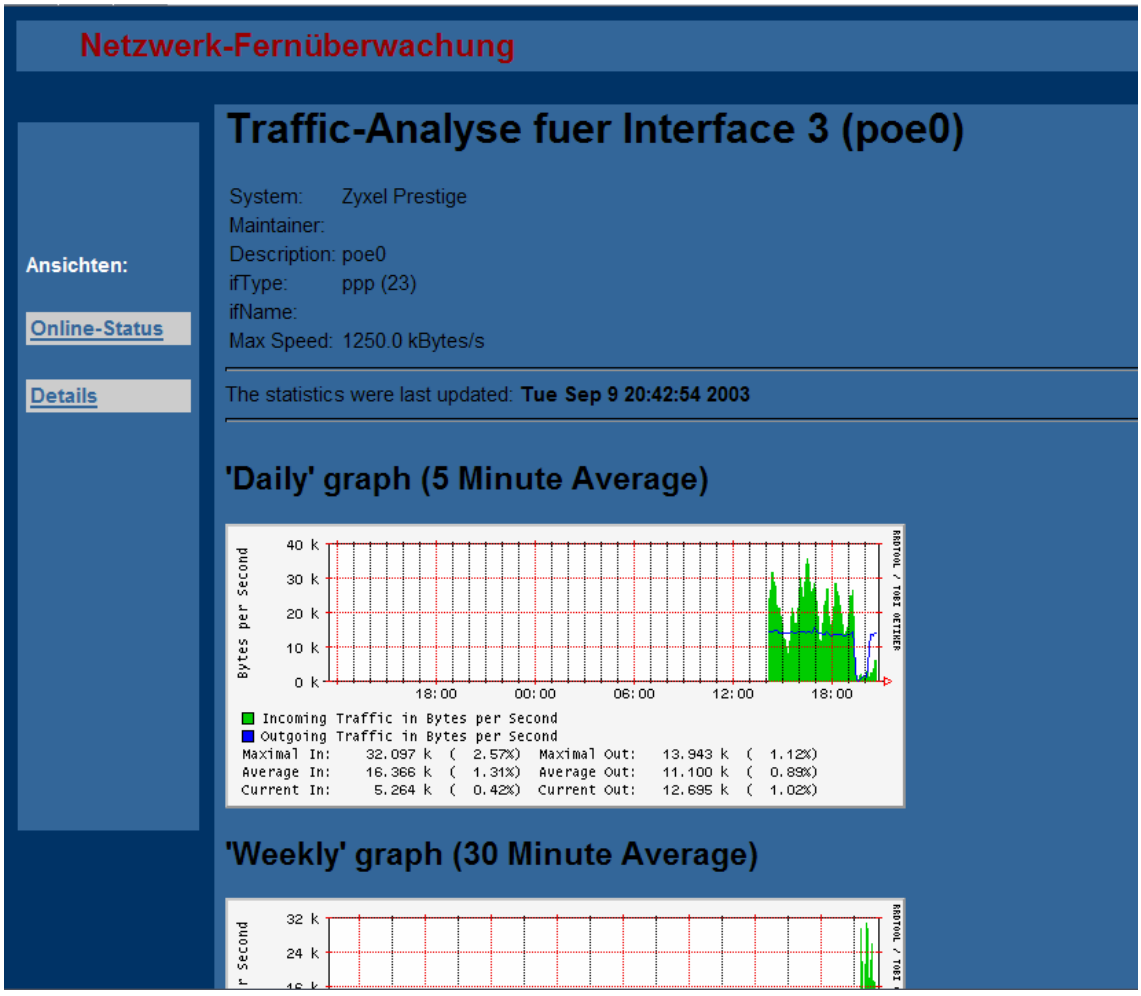

Abbildung 22: Detailansicht, Statistiken für ein ausgewähltes Diagramm (Ausschnitt)

Möchte man die Seite „Online-Status“ um weitere IP-Adressen ergänzen, so müssen der Datei „ipcheck_all“ beliebig viele weitere Zeilen in der Form „ipcheck 192.168.xxx.xxx“ angehängt und die Adressen/Beschreibungen in der Datei „status.html“ ergänzt werden, was sich mit Hilfe des Scripts „statcfg“ (Kap. 6.2.6) erledigen lässt.

Um die Diagramme der Ansicht „Details“ zu erweitern, ist wie folgt vorzugehen:

Entweder wird die Datei „mrtg.cfg“ direkt manuell editiert oder es werden weitere Diagramme mit Hilfe des entwickelten Scripts „graphcfg“ (Kap. 6.2.7) definiert. Falls diese Diagramme nicht nur per direkt eingegebenen Link erreicht, sondern auch in der Übersichtsseite angezeigt werden sollen, muss natürlich zusätzlich die Datei „details.html“ angepasst werden.

Hierfür kann ebenfalls einfach die durch „graphcfg“ automatisch erzeugte Datei details_neu.html verwendet werden, um sich manuelle Anpassungen zu ersparen.

Falls die Datei mrtg.cfg bspw. durch eine Abfrage der CPU-Auslastung von einem System mit der IP-Adresse 192.168.0.99 ergänzt wurde (`graphcfg 192.168.0.99 public cpu`), sieht das dazugehörige Tagesdiagramm mit Link auf die kompletten Diagramme als HTML-Eintrag für die Details-Übersichtsseite ungefähr folgendermaßen aus:

```
<a href="cgi-bin/14all.cgi?log=192.168.0.99_cpu">
<img src="cgi-bin/14all.cgi?log=192.168.0.2_cpu&png=daily.s&small=1.jpg">
</a>
```

Siehe hierzu auch die Dateien dem beigefügten Datenträger.

8 Zusammenfassung

Ziel dieser Diplomarbeit war es, eine übersichtliche Web-Oberfläche zur Fernüberwachung eines kleineren Netzwerkes zu entwickeln.

Dabei wurde ein breites Aufgabenspektrum abgedeckt und durch ein SNMP-Grundlagenkapitel ergänzt.

Es wurde versucht, die durchgeführten Aufgabenschritte so zu beschreiben, dass diese auch für Personen, die weniger mit dem Thema Netzwerkmanagement vertraut sind, relativ leicht nachvollziehbar sind.

Ein positiver Nebeneffekt durch die Verwendung von Linux ist, dass für die Umsetzung dieser Lösung keine Lizenzkosten entstehen und die Web-Oberfläche leicht erweiterbar ist. Zudem gilt Linux als ein ziemlich sicheres und stabiles, d.h. zuverlässiges Betriebssystem.

Auch wenn keine SNMP-Hardware zur Verfügung steht, können zumindest vorhandene Linux- oder Windows-Rechner mit wenigen Schritten per SNMP abgefragt werden, da die Betriebssysteme selbst SNMP-fähig sind.

Die Web-Oberfläche zeichnet sich aus durch:

- Onlinestatus-Seite
- Diagramme und Statistiken auf Abruf
- Passwortgeschützter Zugang
- Erweiterungsmöglichkeit durch Konfigurationsscripte
- Einfache Bedienung

Änderungen, Ergänzungen oder Verbesserungen sind durch den modularen und flexiblen Aufbau mit relativ wenig Aufwand möglich.

Anhang

Anhang A: Konfigurationsdatei „mrtg.cfg“ für MRTG und 14all.cgi

```
### Allgemeine Konfiguration
# Ausgabeverzeichnis
# hier: Verzeichnis im Zugriff des Webservers
WorkDir: /var/www/html/mrtg
# RRDTool verwenden:
LogFormat: rrdtool
# Installationsverzeichnisse von RRDTool:
PathAdd: /usr/local/rrdtool-1.0.45/bin/
LibAdd: /usr/local/rrdtool-1.0.45/lib/perl/
# Alle 5 Min. Daten einholen:
Interval: 5
# Permanent im Hintergrund aktiv bleiben:
RunAsDaemon: Yes
######################################################################
# System:       Zyxel Prestige
# Description:  Router
# Contact:
# Location:
######################################################################

### Interface 1 >> Descr: 'enet0' | Name: '' | Ip: '0.0.0.0' | Eth: '00-30-ab-11-1e-8c'

Target[192.168.0.1_1]: 1:public@192.168.0.1:
SetEnv[192.168.0.1_1]: MRTG_INT_IP="0.0.0.0" MRTG_INT_DESCR="enet0"
MaxBytes[192.168.0.1_1]: 12500000
Title[192.168.0.1_1]: Traffic Analyse-fuer Interface 1 (enet0)
PageTop[192.168.0.1_1]: <font face="Arial, Helvetica, sans-serif"><H1>Traffic-Analyse
f&uuml;r Interface 1 (enet0)</H1></font>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Zyxel Prestige</TD></TR>
   <TR><TD>Maintainer:</TD> <TD></TD></TR>
   <TR><TD>Description:</TD><TD>enet0  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
   <TR><TD>ifName:</TD>     <TD></TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>12.5 MBytes/s</TD></TR>
 </TABLE>
YLegend[192.168.0.1_1]: Bytes/s
Options[192.168.0.1_1]: nobanner
Background[192.168.0.1_1]: #336699
```

```
### Interface 2 >> Descr: 'enet1' | Name: '' | Ip: '' | Eth: '00-00-e2-8b-40-19' ###

Target[192.168.0.1_2]: 2:public@192.168.0.1:
SetEnv[192.168.0.1_2]: MRTG_INT_IP="" MRTG_INT_DESCR="enet1"
MaxBytes[192.168.0.1_2]: 1250000
Title[192.168.0.1_2]: Traffic-Analyse fuer Interface 2 (enet1)
PageTop[192.168.0.1_2]: <font face="Arial, Helvetica, sans-serif"><H1>Traffic-Analyse
f&uuml;r Interface 2 (enet1)</H1></font>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Zyxel Prestige</TD></TR>
   <TR><TD>Maintainer:</TD> <TD></TD></TR>
   <TR><TD>Description:</TD><TD>enet1  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
   <TR><TD>ifName:</TD>     <TD></TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>1250.0 kBytes/s</TD></TR>
 </TABLE>
Options[192.168.0.1_2]: nobanner
Background[192.168.0.1_2]: #336699

### Interface 3 >> Descr: 'poe0' | Name: '' | Ip: '80.131.24.98' | Eth: '' ###

Target[192.168.0.1_3]: 3:public@192.168.0.1:
SetEnv[192.168.0.1_3]: MRTG_INT_IP="80.131.24.98" MRTG_INT_DESCR="poe0"
MaxBytes[192.168.0.1_3]: 1250000
Title[192.168.0.1_3]: Traffic-Analyse fuer Interface 3 (poe0)
PageTop[192.168.0.1_3]: <font face="Arial, Helvetica, sans-serif"><H1>Traffic-Analyse
f&uuml;r Interface 3 (poe0)</H1></font>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Zyxel Prestige</TD></TR>
   <TR><TD>Maintainer:</TD> <TD></TD></TR>
   <TR><TD>Description:</TD><TD>poe0  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ppp (23)</TD></TR>
   <TR><TD>ifName:</TD>     <TD></TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>1250.0 kBytes/s</TD></TR>
 </TABLE>
Options[192.168.0.1_3]: nobanner
Background[192.168.0.1_3]: #336699

######################################################################
# Arbeitsspeicher - Windows 2000
######################################################################

PageTop[192.168.0.2_MEMORY]: <font face="Arial, Helvetica, sans-serif"><h1>Freier Ar-
beitsspeicher</h1></font>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Windows 2000 Professional</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.0.2</TD></TR>
 </TABLE>
Background[192.168.0.2_MEMORY]: #336699
```

```
Target[192.168.0.2_MEMORY]:
1.3.6.1.4.1.311.1.1.3.1.1.1.2.0&1.3.6.1.4.1.311.1.1.3.1.1.1.3.0:public@192.168.0.2

# bei 512 MB RAM installiert:
MaxBytes[192.168.0.2_MEMORY]: 536870912
AbsMax[192.168.0.2_MEMORY]: 1073741824
Title[192.168.0.2_MEMORY]: Bytes frei -- committed
Options[192.168.0.2_MEMORY]: gauge, nobanner
YLegend[192.168.0.2_MEMORY]: Bytes
ShortLegend[192.168.0.2_MEMORY]: Bytes
Legend1[192.168.0.2_MEMORY]: freier Speicher in Bytes
Legend2[192.168.0.2_MEMORY]: "Committed" Speicher in Bytes
LegendI[192.168.0.2_MEMORY]: Frei:
LegendO[192.168.0.2_MEMORY]: Commit.:
# "Committed" (Bezeichnung der Object ID) = Windows zugeteilter Gesamtarbeitsspeicher


######################################################################
# CPU-Auslastung - Windows 2000
######################################################################

PageTop[192.168.0.2_CPU]: <font face="Arial, Helvetica, sans-serif"><h1>aktuelle CPU-
Auslastung</h1></font>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Windows 2000 Professional</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.0.2</TD></TR>
 </TABLE>
Background[192.168.0.2_CPU]: #336699

Target[192.168.0.2_CPU]:
1.3.6.1.4.1.311.1.1.3.1.1.2.1.5.1.48&.1.3.6.1.4.1.311.1.1.3.1.1.2.1.4.1.48:public@192.16
8.0.2

MaxBytes[192.168.0.2_CPU]: 50
AbsMax[192.168.0.2_CPU]: 100
Title[192.168.0.2_CPU]: Durchschnittliche CPU-Auslastung in % (System -- Anwendungen)
Options[192.168.0.2_CPU]: gauge, nopercent, nobanner
YLegend[192.168.0.2_CPU]: CPU-Auslastung in %
ShortLegend[192.168.0.2_CPU]: %
Legend1[192.168.0.2_CPU]: CPU-Auslastung -System- in Prozent
Legend2[192.168.0.2_CPU]: CPU-Auslastung -Anwendungen- in Prozent
LegendI[192.168.0.2_CPU]: Sys.CPU:
LegendO[192.168.0.2_CPU]: Anw.CPU:


######################################################################
# Bytes ueber HTTP gesendet - Windows 2000
######################################################################

PageTop[192.168.0.2_HTTP]: <font face="Arial, Helvetica, sans-serif"><h1>Bytes &uuml;ber
HTTP gesendet/empfangen</h1></font>
```

```
 <TABLE>
   <TR><TD>System:</TD>     <TD>Windows 2000 Professional</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.0.2</TD></TR>
 </TABLE>
Background[192.168.0.2_HTTP]: #336699

Target[192.168.0.2_HTTP]:
1.3.6.1.4.1.311.1.7.3.1.2.0&1.3.6.1.4.1.311.1.7.3.1.4.0:public@192.168.0.2

MaxBytes[192.168.0.2_HTTP]: 1000000000
Title[192.168.0.2_HTTP]: HTTP Bytes gesendet - empfangen
Options[192.168.0.2_HTTP]: nopercent, nobanner
YLegend[192.168.0.2_HTTP]: Bytes
ShortLegend[192.168.0.2_HTTP]:
Legend1[192.168.0.2_HTTP]: Anzahl Bytes ueber HTTP gesendet
Legend2[192.168.0.2_HTTP]: Anzahl Bytes ueber HTTP empfangen
Legend3[192.168.0.2_HTTP]:
Legend4[192.168.0.2_HTTP]:
LegendI[192.168.0.2_HTTP]: Bytes gesend.:
LegendO[192.168.0.2_HTTP]: Bytes empf.:


######################################################################
# freier Festplattenplatz LW C: - Windows 2000
# (Aktivierung per "diskperf.exe -yv" erforderlich!)
######################################################################

PageTop[192.168.0.2_HD]: <font face="Arial, Helvetica, sans-serif"><h1>Freier Festplat-
tenspeicher auf Laufwerk C:</h1></font>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Windows 2000 Professional</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.0.2</TD></TR>
 </TABLE>
Background[192.168.0.2_HD]: #336699

Target[192.168.0.2_HD]:
1.3.6.1.4.1.311.1.1.3.1.1.5.1.4.2.67.58&1.3.6.1.4.1.311.1.1.3.1.1.5.1.4.2.67.58:public@1
92.168.0.2

# bei Festplatte mit 10 GB Gesamtkapazitaet:
MaxBytes[192.168.0.2_HD]: 10000
# Angaben in MByte und GByte:
kMG[192.168.0.2_HD]: M,G
Title[192.168.0.2_HD]: frei auf LW C:
Options[192.168.0.2_HD]: gauge, nopercent, nobanner
Unscaled[192.168.0.2_HD]: dwmy
YLegend[192.168.0.2_HD]: Bytes
ShortLegend[192.168.0.2_HD] : Bytes
Legend1[192.168.0.2_HD]: Verfuegbarer Festplattenspeicher in Bytes
Legend2[192.168.0.2_HD]: "
LegendI[192.168.0.2_HD]: Frei:
```

Anhang B: Datenträger

Dieser Diplomarbeit ist eine CD-ROM beigelegt, welche die HTML-Dateien für die Web-Oberfläche, die Scripte, Tools und Konfigurationsdateien enthält. Anmerkungen zur Verwendung befinden sich in der Datei *Hinweis.txt*.

Inhaltsübersicht:

```
/Hinweis.txt              - Hinweise zur Verwendung der CD-ROM
/Konfigurationsdateien/   - enthält mrtg.cfg, httpd.conf, .htaccess
/Scripts/                 - enthält die Scripte ipcheck, ipcheck_all,
                            statcfg, graphcfg
/Tools/                   - enthält MRTG, RRDTool, SNMP4W2K, 14all.cgi
/Web-Oberflaeche/         - enthält das Grundgerüst der Web-Oberfläche
/Web-Oberflaeche offline/ - enthält eine Version der Web-Oberfläche
                            zur Offline-Ansicht
```

Literaturverzeichnis

Banning, Jens (2002): Linux Netzwerkadministration. Addison-Wesley Verlag München

Bawidamann, Rainer (2003): 14all.cgi-Dokumentation. http://my14all.sourceforge.net
(Datum des letzten Zugriffs: 9. September 2003)

Cisco Systems (1996): Simple Network Management Protocol (SNMP).
Product Overview. http://www.cisco.com/warp/public/535/3.html
(Datum des letzten Zugriffs: 9. September 2003)

Cisco Systems (2002): Simple Network Management Protocol (SNMP).
Documentation. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm
(Datum des letzten Zugriffs: 9. September 2003)

Cohen, Yoram (1995): SNMP - Simple Network Management Protocol.
http://www2.rad.com/networks/1995/snmp/snmp.htm
(Datum des letzten Zugriffs: 9. September 2003)

Korbel, Oliver (2000): Der Batchdaemon cron. Pro-Linux
http://www.pl-forum.de/t_system/crontab.html
(Datum des letzten Zugriffs: 9. September 2003)

Oetiker, Tobias; Rand, Dave (2003): MRTG- und cfgmaker-Dokumentation.
http://people.ee.ethz.ch/~oetiker/webtools/mrtg/reference.html
http://people.ee.ethz.ch/~oetiker/webtools/mrtg/cfgmaker.html
(Datum des letzten Zugriffs: 9. September 2003)

Oetiker, Tobias (2003): RRDTool-Dokumentation.
http://people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg-rrd.html
http://people.ee.ethz.ch/~oetiker/webtools/rrdtool
(Datum des letzten Zugriffs: 9. September 2003)

Red Hat (2002): Red Hat Linux 8.0 - Das Offizielle Red Hat Linux Referenzhandbuch. http://www.europe.redhat.com/documentation/rhl8.0/rhl-rg-de-8.0 (Datum des letzten Zugriffs: 17. November 2003)

Riekert, Wolf-Fritz (2002): Vorlesungsscript Computernetze. HdM Stuttgart http://v.hdm-stuttgart.de/~riekert/lehre/cn.pdf (Datum des letzten Zugriffs: 9. September 2003)

Saxonia Systems: Die C-Shell und Tcsh. http://www.linuxfibel.de/csh.htm (Datum des letzten Zugriffs: 9. September 2003)

Internet Engineering Taskforce (IETF): Request for Comments No. 1066, 1155, 1157, 1212, 1213 http://www.ietf.org/rfc.html (Datum des letzten Zugriffs: 17. November 2003)

Williams Technology Consulting Services (2003): SNMP4TPC. http://www.wtcs.org/snmp4tpc (Datum des letzten Zugriffs: 9. September 2003)

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ort, Datum	Unterschrift