Management von Access Routern auf der Basis von Cabletron Spectrum

DIPLOMARBEIT

zur Erlangung des akademischen Grades
Diplomingenieur(FH)
an der
Fachhochschule für Technik und Wirtschaft Berlin

Fachbereich Ingenieurwissenschaften I
Studiengang Technische Informatik

Betreuer: Prof. Wolfgang Schebesta
Dr. Thomas Schmidt

Eingereicht von Thorleif Wiik

Berlin, 29.01.98

Vorwort

Das Internet ist das größte Computernetzwerk der Welt. Es ermöglicht den Menschen, ungeachtet von Ländergrenzen über Kontinente hinweg zu kommunizieren, als ob sie nur wenige Kilometer voneinander entfernt wären. Die Kommunikation findet über die verschiedensten Medien statt; noch sind e-mail und das World Wide Web die meistgenutzten Dienste, doch auch hier zeichnen sich Veränderungen ab.

Parallel zu dieser Entwicklung ist eine starke Verbreitung von Telearbeitsplätzen zu beobachten, zuerst in Amerika, jetzt auch in Europa.

Die Menschen, die diese beiden Möglichkeiten von zu Hause aus nutzen, haben eines gemeinsam:

Sie müssen sich von ihrem heimischen Rechner über das Telefonnetz – sei es per Modem oder ISDN – mit einem anderen Netz verbinden. Diese Verbindung zu einem anderen Netz kommt im allgemeinen über einen Access-Router zustande.

Da es jetzt einige Millionen Internetnutzer gibt, die sich über das Telefonnetz mit dem Internet verbinden, läßt es sich leicht vorstellen, wie viele Geräte dieser Art irgendwo installiert sein müssen.

Diese Geräte müssen natürlich von den Betreiben kontrolliert und überwacht werden, damit entstehende Probleme frühzeitig erkannt werden. Dies bezeichnet man als das Management, wobei wir jetzt auch beim Thema dieser Arbeit wären: Das Management von Access-Routern auf der Basis von Cabletron Spectrum.

Die Idee zu dieser Arbeit entstand im Frühjahr 1997, Gespräche mit Betreibern und Herstellern von Access-Lösungen zeigten, daß an dieser Problematik Interesse bestand.

So auch die Firma Cabletron Systems, bei der ich mich hiermit für die Bereitstellung des Managementsystems Spectrum inklusive Entwicklungsumgebung bedanken möchte.

Berlin, im Januar 1998 Thorleif Wiik

Inhaltsverzeichnis

Abkürzungsverzeichnis

Aufgeführt sind nur jene Abkürzungen, die mehrfach verwendet werden und in engem Zusammenhang mit dem Thema stehen.

API	Application Programmers Interface
ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notation one
CHAP	Challenge Handshake Authentication Protocol
CMIP	Common Network Information Protocol
DCM	Device Communication Manager
DMI	Desktop Management Interface
EPI	External Protocol Interface
FTP	File Transfer Protocoll
GIB	Generic Information Block
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
ICMP	Internet Control Message Protocol
ID	Identifier
IETF	Internet Engineering Task Force
IIB	Icon Information Block
IP	Internet Protocol
IPX	Internet Packet Exchange
ISDN	Integrated Services Digital Network
LAN	Local Area Network
MIB	Management Information Base
MIF	Management Information File
MM	Management Module
MT	Model Type
MTE	Model Type Editor
NMS	Netzwerkmanagementsystems
OID	Object Identifier
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First

PAP	Password Authentication Protocol
PPP	Point to Point Protocol
PDU	Protocol Data Unit
RADIUS	Remote Authentication Dial In User Service
RAS	Remote-Access-Server
RFC	Request for Comments
RIP	Routing Information Protocol
SLIP	Serial Line Internet Protocol
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
SSAPI	Spectroserver Application Programming Interface
TACACS	Terminal Access Controller Access Control System
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
WAN	Wide Area Network
WWW	World Wide Web

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

ARP Das Adress Resolution Protocol wird zum Addressmapping von IP-Adressen auf Hardware-Adressen benutzt.

ASN.1 Die Abstract Syntax Notation One ist eine OSI-Definition zur systemübergreifenden Beschreibung von Daten.

CMIP Das Common Network Information Protocol ist ein OSI Management-Protokoll.

Dial-In Bezeichnung für eine Einwählleitung.

LAN Mit Local Area Networks bezeichnet man Datennetze mit relativ begrenzter Ausdehnung.

MIB Die Management Information Base ist eine Ressource und kann eine Vielzahl von Parametern und Attributen enthalten.

OID Object Identifier dienen zum Kennzeichnen einzelner managbarer Objekte.

PPP Das Point to Point Protocol ermöglicht die Übermittlung von Datenpaketen über Wählleitungen.

RFC Die Requests for Comments sind Arbeitspapiere, Protokollspezifikationen oder Kommentare zu aktuellen Themen der Internet Community.

SLIP Das Serial Line Internet Protocol wird zur Übertragung von TCP/IP-Daten über serielle Verbindungen verwendet.

SNMP Das Simple Network Management Protocol baut auf TCP/IP auf und gilt als Standard-Managementprotokoll.

TCP/IP Standardprotokoll-Suite zur Kommunikation in Rechnernetzen.

TFTP Das Trivial File Transfer Protocol ist ein Filetransferprotokoll in TCP/IP-Umgebungen.

WAN Wide Area Networks sind Datennetze mit unbegrenzter Ausdehnung.

1. Einleitung

Die meisten Fachhochschulen und Universitäten bieten ebenso wie das Rechenzentrum der FHTW allen Studenten und Mitarbeitern die Möglichkeit an, sich von zu Hause über ihren PC mittels Modem oder ISDN mit dem Campusnetz ebenso wie mit dem Internet zu verbinden.
So stellt das Rechenzentrum der FHTW derzeit 46 Zugänge über Access-Server bereit, ein weiterer Ausbau der Zugänge ist für 1998 geplant.
Die Motivation dieser Arbeit ist es herauszufinden, welche Möglichkeiten es beim Management eines Access-Routers des Typs Ascend MAX4000 über das verfügbare Managementsystem Cabletron Spectrum gibt. Dabei liegt der Schwerpunkt der Arbeit bei der Implementierung eines Management Moduls in Spectrum selbst.
Die vorliegende Arbeit gliedert sich in acht Teile.
In einem knappen Exkurs werden im Kapitel 2 ein Überblick über Netzwerkmanagementprotokolle, im speziellen über das Simple Network Management Protocol gegeben und das Managementsystem Cabletron Spectrum und dessen Entwicklungsumgebungen vorgestellt.
In Kapitel 3 wird der Ausdruck Access-Router definiert, auf dessen Eigenheiten eingegangen und der Ascend MAX4000 vorgestellt. Daraufhin wird auf die Ansprüche an das Management eines Access-Servers eingegangen.
Kapitel 4 enthält den Entwurf eines Management Moduls für einen Ascend Access-Router in Cabletron Spectrum, im weiteren Verlauf der Arbeit wird in Kapitel 5 auf die Implementierung des Moduls und die dabei auftretenden Schwierigkeiten eingegangen. Hinweise zum Test des Management-Moduls sind schließlich in Kapitel 6 zu finden.
Das Resümee der vorangegangenen Kapitel findet sich in Kapitel 7 wieder und liefert eine Einschätzung der Managebarkeit eines Access-Routers aufgrund der gemachten Erfahrungen. Dieses geht in Kapitel 8 über, wo weitere Möglichkeiten für die Zukunft geschildert werden.

2. Managementframework

2.1. Überblick

Grundlegend wird Management als Verwaltung, Organisation und Regulierung einer Einheit verstanden. Abgebildet auf eine Firma umfaßt das Management verschiedene Bereiche: Verwaltung, Buchhaltung, Marketing, Vertrieb, Planung, Support usw.. Das Management ist ein sensibler und bedeutender Faktor innerhalb einer Organisation. Dies läßt sich ebenso auf das Management von Computernetzen übertragen.
Der Netzwerkadministrator wird bei der Erfüllung seiner Aufgaben durch das Netzwerkmanagement unterstützt. Sein Arbeitsmittel sind dabei ein oder mehrere Netzwerkmanagementsysteme (NMS), welche für ihn in erster Linie die Visualisierung der Daten des ihm unterliegenden physischen Netzwerks an seinem Arbeitsplatz ermöglichen. Ein Netzwerkmanagementsystem bietet dem Administrator die Möglichkeit, eine oder mehrere graphische Ansichten eines Netzwerks zu bilden, wobei die verschiedensten Informationen aus dem Netzwerk über ein Netzwerkmanagementprotokoll geliefert werden. Der Administrator soll durch das NMS möglichst schnell einen Überblick über das Netz und dessen Betriebszustand bekommen. So hat das Netzwerkmanagement hierbei zwei Hauptfunktionen: Überwachung und Kontrolle. Diese beiden Ziele lassen sich nach Open Systems Interconnection (OSI) durch folgende Gruppen weiter unterteilen[4]:

Fehlermanagement

Das Fehlermanagement in einem Netzwerk mit der Möglichkeit des Erkennens, Isolierens und Behebens von Störungen ist ein wichtiger Punkt, welcher eine Anzahl Fähigkeiten beeinhalten sollte. Dazu gehören das Lokalisieren von Netzwerkproblemen, das Erstellen von Fehlerstatistiken, das Entdecken der Störungsursachen, das Einleiten der korrigierenden Maßnahmen und das anschließende Überprüfen der Korrektur.

Konfigurationsmanagement

Das Konfigurationsmanagement soll die Funktionalität anbieten, Daten von den Geräten im Netzwerk ebenso einzulesen wie deren Konfiguration beeinflussen zu können. Dabei stehen weiterhin außerdem solche Merkmale im Mittelpunkt wie das automatische Entdecken der Netzwerkstruktur und dessen Inventar, das Feststellen von Änderungen, wie auch das Ausführen von Änderungen im Netzwerk.

Performancemanagement

Das Performancemanagment dient zur Kontrolle von Bandbreitennutzung, Gerätenutzung und Datenvolumen in Segmenten des Netzes. Das Performancemanagement erlaubt die Bewertung des Zustands des Netzwerks, wobei auch der Vergleich verschiedener Netzwerkkonfigurationen in Hinblick auf den Zustand miteinbezogen wird. Ebenso gehören Fähigkeiten wie das Ansammeln von Netzwerkdaten, zum Beispiel Paketraten, Kollisionen und Durchsätze, das Übersetzen der Daten in Performancekonzepte, die Bewertung des Netzwerks im Hinblick auf alternative Konfigurationen und schließlich die Einleitung sinnvollerer Konfigurationen dazu.

Accountingmanagement

Um die Kosten für die Nutzer eines Netzwerks zu berechnen, ist ein Accountingmanagement vonnöten, welches im wesentlichen dazu dienen soll, die Kostenanteile für die Nutzung bestimmter Dienste herauszufinden, Daten im Hinblick auf die Nutzung bestimmte Services zu sammeln und nicht zuletzt dem Anwender seine Nutzung in Rechnung zu stellen.

Sicherheitsmanagement

Immer mehr in den Vordergrund rückt auch das Sicherheitsmanagement, welches dazu dienen soll, Informationen gegen unerlaubte Eindringlinge zu schützen, erfolgreiche und fehlgeschlagene Angriffe zu dokumentieren, Eindringlinge zu identifizieren, ebenso wie Grundsätze für die Nutzung des Netzwerks zu erstellen, Verschlüsselungscodes einzuführen und zu warten, Aufzeichnungen über

Zugriffe zu verwalten, unerlaubten Zugriff zu verhindern und zu dokumentieren, Prozeduren bei unerlaubtem Zugriff einzuleiten und nicht zuletzt Computerviren zu erkennen und zu beseitigen.

Die Praxis

Die in dieser summarischen Übersicht genannten Punkte werden in der Praxis sicherlich nicht immer vollständig erfüllt.
Je nach Umgebung sind meist nur einige der oben genannten Punkte überhaupt in einen spezifischen Anforderungskatalog zu übernehmen.
Aber wie sieht die Praxis überhaupt aus? In der Regel befinden sich in einem Computernetzwerk eine Anzahl von Geräten wie Hosts, Hubs, Routern und weitere aktive Komponenten, die zu managen sind. Als Basis hat sich in TCP/IP-Umgebungen das Simple Netzwork Management Protocol (SNMP) etabliert, welches wir im folgenden Abschnitt genauer betrachten wollen.

2.2. Der Standard - SNMP

Während herstellerspezifische Netzwerke früher durch herstellereigene Managementsysteme ausreichend überwacht und gesteuert werden konnten, ist dieses für heute übliche heterogene Landschaften und herstellerübergreifende Netzwerke nicht mehr der Fall. Hier kam der Standard SNMP, der aus dem Simple Gateway Monitoring Protocol (SGMP) Ende der 80er Jahre hervorging. Durch das schon damalige Wachstum des Internets wurde die Anforderung nach Management - und einem Standard - immer größer. Dies endete dann in folgenden drei RFCs[1](Request for Comments)[5]:

- Structure of Management Information (SMI)[6]
- Management Information Base (MIB)[7]
- Simple Network Management Protocol (SNMP)[8]

Das Design von SNMP ermöglicht es, ohne physikalische Verbindung Netzwerkkomponenten zu managen. Ein anderer Ansatz des Managements besteht zum Beispiel darin, im Netz vorhandene Ethernetschnittstellen in den sogenannten *Promiscuous Mode* zu bringen, um dadurch Informationen über den Zustand der Netzwerksegmente zu erhalten.
Bei SNMP jedoch muß lediglich eine TCP/IP-Verbindung zu den entsprechenden Geräten bestehen, SNMP erlaubt dann den Austausch von Zustandsvariablen über diese Verbindung. Daraus entstand der Begriff des *remote management*. Die wahre Stärke von SNMP liegt aber darin, Geräte der verschiedensten Hersteller managen zu können, und dabei ein und dasselbe Interface zu haben, um an die nötigen Informationen zu kommen, ohne dabei die Kenntnis über viele herstellerspezifische Kommandos haben zu müssen. Deshalb wurde aus SNMP dann schnell der Defacto-Standard, wie die vielen Implementierungen in TCP/IP-Umgebungen zeigen.
Basierend auf der SNMP Version 1 wurden mehrere Entwürfe von SNMPv2, (SNMPv2C, SNMPv2* und SNMPv2u) entwickelt, jedoch wurden diese Varianten nicht von der Industrie angenommen, unter anderem deshalb, weil sie nicht mehr dem Grundsatz der Einfachheit entsprachen.

[1]RFCs sind die offiziellen Dokumente für die IP-Protokoll-Familie.

Derzeit ist die Internet Engineering Task Force (IETF) gerade dabei, den Entwurf für SNMPv3 fertigzustellen. Hierzu fand im August 1997 eine IETF-Konferenz in München statt, und bis April 1998 sollen alle Spezifikationen von SNMPv3 veröffentlicht sein. Wenn wir im weiteren Verlauf von SNMP reden, meinen wir aber immer noch SNMPv1.

2.2.1. Das Modell

Bei SNMP wird nicht das klassische Client/Server-Modell eingesetzt. Man unterscheidet hier zwischen Agent und Manager. Die Agenten befinden sich auf den zu managenden Geräten und stellen die Informationen für den oder die Manager bereit. Der Zugriff des Managers oder der Manager auf die Agenten kann sowohl lesend als auch schreibend erfolgen, dieser wird durch die Read/Write-Community-Strings ermöglicht. Ein Community-String ist als Passwort zu verstehen, erst darüber wird der Zugriff auf einen Agenten möglich. Viele Agenten erlauben zusätzlich die Definition von Zugriffslisten, um Anfragen nur von bestimmten Managern zuzulassen.
Manager fragen in sich wiederholenden Zyklen die Agenten ab. Die Intervallzeit sollte hierbei nicht zu niedrig ausfallen, um erhöhte Netzlast zu vermeiden, was andererseits zur Folge haben kann, daß Detailinformationen verloren gehen könnten.
Als Beispiel folgendes Szenario: Ein Manager pollt in einem Abstand von 10 Minuten alle Netzwerkschnittstellen eines Routers. In dieser Zeit gab es jedoch einen Ausfall auf einer Netzwerkschnittstelle. Das Netzwerkmanagementsystem kann diese Veränderung nicht melden, da es diese Information nie erhalten hat. Um dies zu vermeiden, verfügen die Agenten zusätzlich über die Möglichkeit, selbst Meldungen an Manager zu versenden, diese Mitteilungen werden *Traps* genannt.
Zusammenfassend lassen sich die vier Komponenten des Managementframework in Abbildung 1 wiedererkennen. Dazu gehören ein oder mehrere zu *managende Geräte*, auf denen sich jeweils ein Managementagent befindet, mindestens ein *Netzwerkmanagementsystem*, auf dem eine oder mehrere Netzwerkmanagementapplikationen laufen. Zur Kommunikation kommt ein *Netzwerkmanagementprotokoll* zum Einsatz, welches dazu dient, *Managementinforma-*

tionen auszutauschen. Hierbei ist es möglich, die zu managenden Geräte in zwei Kategorien zu unterteilen: *physische Systeme*, wie Router, Hosts oder Hubs und *logische Systeme*, wie Dienste und Applikationen.

Abbildung 1: Das Managementmodell

2.2.2. Die Management Information Base

Allgemein gesehen ist eine MIB eine geordnete Kollektion von Zustandsvariablen. Sie beschreibt Informationen, die über ein Netzwerkmanagementprotokoll abgerufen und/oder geändert werden können. Erst diese Informationen ermöglichen es, daß Systeme in einem Netzwerk mittels SNMP gemanagt werden können. So kann entsprechend dem Design der MIBs SNMP dazu genutzt

werden, um Fehler-, Konfigurations-, Performance-, Accounting- und Sicherheitsmanagement durchzuführen.
Eine MIB definiert managebare Objekte anhand von SMI. SMI beschreibt, wie die Managementinformation gruppiert und benannt ist, die erlaubten Operationen, Datentypen und Syntax. Der Zugriff auf die einzelnen Objekte erfolgt

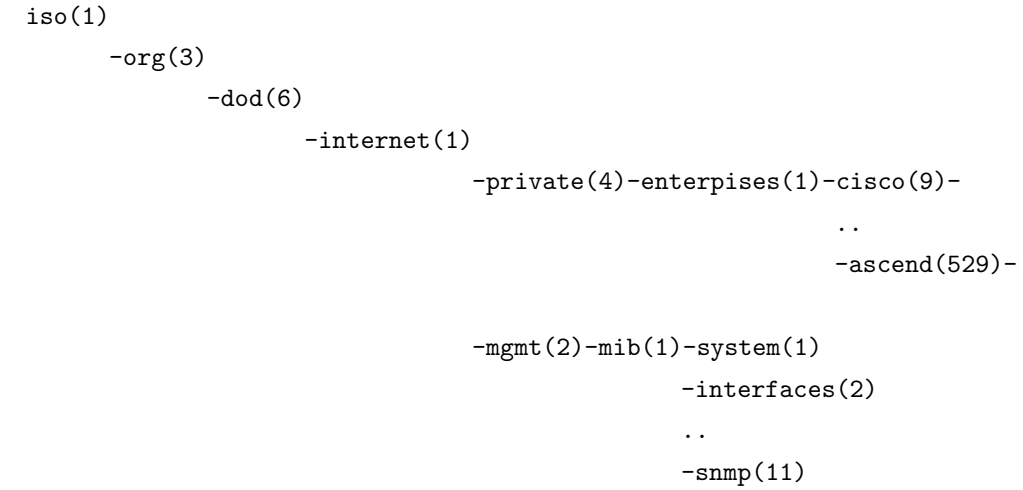

Abbildung 2: Der MIB Baum

dabei über *Object Identifiers* (OIDs). OIDs sind eine Sequenz von nichtnegativen Integern, welche hierarchisch organisiert sind. In Abbildung 2 ist ein Ausschnitt des OID-Baums für SNMP zu erkennen. OIDs werden üblicherweise in einem der Formate wie in Abbildung 3 beschrieben: Der SMI Standard

```
{ iso(1) org(3) dod(6) internet(1) mgmt(2) mib(1) system(1) sysDescr(1) }

            iso.org.dod.internet.mgmt.mib.system.sysDescr

                        1.3.6.1.2.1.1.1
```

Abbildung 3: OID Darstellungsarten

sagt aus, daß alle MIB-Variablen nach *Abstract Syntax Notation one* (ASN.1) definiert und referenziert werden müssen. ASN.1 ist eine formale Sprache, die zwei Haupteigenschaften hat: Die ASN.1 Notation wird in Dokumenten, die von Menschen gelesen werden, benutzt und dient in Kommunikationsprotokollen zur Datenrepräsentation. In Abbildung 4 ist anhand der in MIB-II vorhan-

denen Beschreibung des zu managenden Systems ein Beispiel der Objektdefinition in ASN.1-Darstellung zu sehen.

```
sysDescr OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
         "A textual description of the entity.  This value
         should include the full name and version
         identification of the system's hardware type,
         software operating-system, and networking
         software.  It is mandatory that this only contain
         printable ASCII characters."
    ::= { system 1 }
```

Abbildung 4: Objekt-Definition in ASN.1-Darstellung

SNMP-Agenten enthalten üblicherweise den vom Gerät unterstützten Anteil der *MIB-II*[2]. In dieser sind Kategorien wie *system, interfaces, ip, icmp, tcp, udp* usw. zu finden. Da diese Gruppen nicht ausreichen, um gerätespezifische Daten darzustellen, besteht die Möglichkeit, diese durch herstellerspezische MIBs zu repräsentieren, diese sind im MIB-Baum unter `1.3.6.1.4.1` wiederzufinden und werden *Enterprise MIBs* genannt.

2.2.3. Das Protokoll

SNMP setzt als Applikationsprotokoll auf dem verbindungslosen User Datagram Protocol (UDP) auf. Hier wird zur Kommunikation zwischen Manager und Agent die UDP-Portnummer 161 verwendet, Traps jedoch werden auf UDP-Port 162 übertragen. In Tabelle 1 sind die von SNMPv1 unterstützten fünf Operationen und die SNMPv2 Erweiterungen aufgelistet. Diese werden als Protocol Data Units (PDUs) übermittelt.

[2]MIB-II ist die Weiterenticklung der RFC Standard-MIB.

Tabelle 1: SNMP Operationen

Kommando	Bedeutung
`get-request`	Wert einer bestimmten Variable lesen
`get-next-request`	Abfragen des Nachfolgers in der Liste der Attribute
`get-response`	Antwort auf eine Abfrage
`set-request`	Wert einer bestimmten Variable schreiben
`trap`	Antwort, ausgelöst durch Ausnahmesituation
`get-bulk-request`	*(SNMPv2)* Kann eine Vielzahl von Variablen lesen
`inform`	*(SNMPv2)* Traps zwischen verschiedenen NMS

2.2.4. Einschränkungen

SNMP hat neben seinen vielen Vorteilen auch einige bekannte Mängel. So werden die Daten und somit auch die SNMP-Community-Strings, also die Passwörter, durch die erst der Zugriff auf die Agenten möglich wird, unverschlüsselt durch das Netz übertragen. Dies ermöglicht es durch sogenannte Netzwerksniffer, diese sensiblen Daten zu abzuhören. Mithilfe der SET-Community-Strings wäre es dem Eindringling dann möglich, ganze Netzwerksegmente abschalten zu können.
Da es keinen Befehl für die Sammelabfrage von Werten aus der MIB gibt, und die Variablen einzeln übertragen werden, entsteht ein hoher Kommunikationsaufwand und somit eine erhöhte Netzlast. Dieser Umstand wurde in SNMPv2 mit dem Befehl `get-bulk-request` behoben.
Leider bieten die Implementierungen der SNMP-Agenten verschiedener Hersteller oft nicht genügend Möglichkeiten, um ein zeitgemäßes Management durchzuführen, dies ist vor allem im Bereich der Hosts zu beobachten.

2.3. Weitere Entwicklungen

SNMP hat sich sehr gut zur Überwachung von Netzwerkkomponenten bewährt. Es gibt aber auch SNMP-Agenten zum Managen von Datenbanken wie Sybase, Oracle usw. Diese Agenten lassen neben dem Auslesen von Daten auch zu, Konfigurationsänderungen über SNMP vorzunehmen. Doch SNMP bietet nicht genügend Möglichkeiten zum Managen von Desktop-Rechnern und gan-

zen Server-Farmen. Hierzu wurde das *Desktop Management Interface* (DMI) entwickelt. Analog zur SNMP-MIB holt DMI seine Informationen aus dem *Management Information File* (MIF). Ein Blick auf den Markt zeigt, daß einige Hersteller bereits DMI-Agenten in ihrer Produktpalette haben. So liefert IBM mit seinem Netfinity-Server ebenso wie Sun mit Solaris in der Version 2.6 einen DMI-Agenten mit aus. Das OSI Managementprotokoll mit dem Namen *Common Network Information Protocol* (CMIP) hat in Computernetzwerken so gut wie keine Bedeutung, es findet sich aber im Bereich der *Telecommunications Management Networks* (TMNs) wieder.

2.4. Spectrum

2.4.1. Überblick

Cabletron Spectrum ist eine integrierte Plattform für das Netzwerk- und Systemmanagement. Es ist ein offenes System auf Client/Server-Basis. Standardmäßig unterstützt Spectrum TCP/IP und die darauf basierenden Protokolle SNMP und *Internet Control Management Protocol* (ICMP), kann aber auch durch andere Managementprotokolle ergänzt werden. Nur so kann Spectrum dazu genutzt werden, nicht nur Netzwerkmanagement durchzuführen, sondern auch Systemmanagement, zum Beispiel unter Einsatz proprietärer Protokolle oder zu erwartender Standards wie DMI.
In der Grundausstattung jedoch ist Spectrum in erster Linie ein Netzwerkmanagementsystem und besteht aus Spectroserver und Spectrograph. Daneben werden eine Vielzahl von Applikationen mitgeliefert:

AutoDiscovery dient zum Entdecken eines Netzwerks anhand von Adressbereichen mit verschiedenen Methoden wie Router-, Range-, Network Information Service (NIS) und Address Resolution Protocol (ARP)-Discovery.

Command Line Interface ist eine Schnittstelle zum Spectroserver auf Kommandozeilenebene.

Imaging Toolkit dient zur graphischen Gestaltung der Netzwerkansichten.

PathView zeigt alle vorhandenen Netzwerkschnittstellen und deren Auslastung auf dem Weg zwischen zwei Geräten an.

SpectroWATCH dient zur Berechnung und Überwachung von Variablen im System.

Data Export ist die Schnittstelle von Spectrum zu verschiedenen Datenbanksystemen wie *Sybase* und *Oracle*.

Report Generator kann u.a. neben graphischen Langzeitanalysen der Netzwerkauslastung auch Inventarlisten erzeugen.

Scheduler führt automatisch Vorgänge zu bestimmten Uhrzeiten aus, dies kann zum Beispiel ein Backup der Spectrum-Datenbank sein.

Control Panel dient zum Start und zur Konfiguration von Spectroserver und Spectrograph.

MAC Address Locator Tool ermöglicht die Suche nach MAC-Adressen im modellierten Netzwerk.

Ich möchte hier nicht weiter diese Programme beschreiben, sondern mich auf den Spectroserver und Spectrographen beschränken. Weitere Informationen sind unter anderem in [11] und [15] zu finden.

Spectroserver

Die Serverkomponente von Spectrum heißt Spectroserver und beeinhaltet, wie in Abbildung 5 zu sehen ist, als Kernstück die Spectrum-Datenbank. In dieser Datenbank befinden sich zum einen Metainformationen, wie zum Beispiel die Kenntnis von herstellerspezifischen SNMP-MIBs, die daraus entstandenen Module und deren Verhältnisse zueinander. Diese werden als *Model Types* bezeichnet.
Im laufenden Betrieb werden Informationen über das vorhandene Netzwerk in der Datenbank gesammelt, zum Beispiel Betriebsstatistiken, Gerätedaten und Konfiguration und Topologie des Netzwerks.
Jedes zu managende Gerät ist eine Instanz eines Model Types und wird als *Model* bezeichnet. Auf diese Daten wird auf der unteren Ebene über den *Device Communication Manager* (DCM) zugegriffen. Der DCM ist dafür zuständig, Anfragen an die zu managenden Geräte in das entsprechende Managementprotokoll zu übersetzen, als Beispiel sei SNMP genannt, und in Intervallen alle im System befindlichen Models zu überwachen. Der Zugriff der Spectrum-Clienten auf den Spectroserver erfolgt über das *Spectroserver Application Programming Interface* (SSAPI). Die Kommunikation zwischen Client und Server erfolgt via TCP/IP, wobei hierbei nahezu beliebig viele Clienten auf den Spectroserver zugreifen können.

Spectrograph

Der Spectrograph ist die Client-Applikation, mit der wohl am meisten gearbeitet wird. Er besteht aus einer graphischen Oberfläche, die dazu dient, die

Abbildung 5: Spectrum Architektur

im Spectroserver befindlichen Daten, sprich das Netzwerk, zu visualisieren. So können mehrere Nutzer gleichzeitig über mehrere Spectrographen auf den Spectroserver zugreifen, das Netzwerk überwachen und modellieren. Die Nutzerverwaltung von Spectrum gibt den Anwendern hierbei die entsprechenden Rechte. Diese reichen von der Modellierung der Ansicht bis zum reinen Betrachten von wenigen Subnetzen.
Der Spectrograph bietet hierbei verschiedene Perspektiven an. Zum einen gibt es den in Abbildung 6 dargestellten *Topology View*, der die physische Sicht des Netzes widerspiegelt, zum anderen den *Geographical View*, der es ermöglicht, die zu managenden Geräte in geographischer Sichtweise zu plazieren (Stadt, Gebäude, Raum). Zuletzt sei noch der *Organization Chart View* erwähnt, der eine graphische Gestaltung entsprechend der Unternehmenshierarchie zuläßt.
Der Alarmmanager zeigt alle im Spectroserver vorhandenen Alarme und deren Ursachen an. Er ermöglicht eine schnelle Einkreisung der im Netzwerk vorhan-

denen Probleme. Die verschiedenen Farben in den Icons und Views zeigen den

Abbildung 6: Spectrograph Topology View

Status der Elemente an: So steht zum Beispiel Grün für Ok, Rot für Ausgefallen und Braun für Inaktiv und Ok. Durch die sogenannte *Inductive Modeling Technologie* (IMT) wird bei einem Ausfall eines verbindenden Knotens hierbei nur für diesen ein Alarm ausgelöst und das Gerät in der Darstellung mit der Farbe Rot versehen. Alle dahinterliegenden, und somit nicht notwendigerweise ursächlich betroffenen Komponenten, werden in der Farbe Grau als nicht erreichbar dargestellt. Innerhalb der Views ist es wiederum möglich, auf Subviews zuzugreifen, meistens über Icons, die die entsprechenden Geräte – oder in Spectrum gesprochen: Models – repräsentieren.

Erweiterungen

Als typische Spectrumerweiterungen seien an erster Stelle die Management Module (MM) genannt, welche es ermöglichen, die gerätespezifischen SNMP-Erweiterungen von Netzwerkkomponenten zu managen.
Ein Managementmodul erweitert hierbei sowohl den Spectroserver, als auch den Spectrographen. Hierbei wird dem Spectroserver u.a. die Enterprise-MIB bekanntgemacht, der Spectrograph wird um neu hinzugekommen Views und Icons ergänzt. Sollten mehrere Spectroserver und Spectrographen vorhanden sein, ist es deshalb notwendig, die Erweiterung für alle Server und Clienten zu lizensieren.
Spectrumerweiterungen sind natürlich nicht auf SNMP-basierende Geräte begrenzt, sondern beinhalten eine Vielzahl von High-Level-Applikationen, zum Beispiel SpectroRX, ein Werkzeug, das auf CBR (Case Based Reasoning) basiert und somit zur automatischen Eingrenzung der verschiedensten Probleme genutzt werden kann.
Des weiteren bietet Spectrum die Integration mit einer hohen Anzahl von Applikationen von Drittherstellern, als Beispiel sei hier WinWatch der Firma Metrix genannt. Durch Einsatz dieser Erweiterung ist es mit Hilfe von Spectrum möglich, automatisch im Netzwerk verteilte Installationen von Software auf Microsoft Windows Plattformen einzuleiten.

2.4.2. Die Entwicklungsumgebungen

Neben den kommerziell erhältlichen Managementmodulen ist es möglich, mit Hilfe verschiedener Entwicklertools Spectrum entsprechend den eigenen Wünschen zu erweitern.
Hierzu bietet Cabletron folgende Produktgruppen an:

Level I Tools ermöglichen es, Icons und graphische Ansichten zu verändern, ebenso wie herstellerspezifische MIBs einzulesen und neue Managementmodule zu kreieren.

Level II Tools ermöglichen es, neue Elemente sowohl für den Spectrograph als auch für den Spectroserver zu entwickeln. Die API basiert auf C/C++.

Die Bestandteile der Level II Entwicklungsumgebung werden abgeleitet von der zu erweiternden Spectrum-Schnittstelle wie folgt unterteilt:

External Protocol API Toolkit bietet eine Schnittstelle, durch die es ermöglicht wird, Spectrum um weitere Kommunikationsprotokolle zu ergänzen.

Spectroserver API Toolkit dient z.B. zur Anbindung neuer Clienten an den Server, und um neue Elemente für die graphische Repräsentation im Spectroserver zu generieren.

Inference Handler API Toolkit läßt die Erweiterung des Inference Handlers, der sogenannten Basis der Managementintelligenz von Spectrum, zu.

In Bild 7 [13] ist zu sehen, an welchen Punkten in Spectrum die Entwicklungsumgebungen ansetzen. In unserem Fall stehen die Level I Tools zur Verfügung. Diese bestehen aus folgenden Einzelumgebungen:

Model Type Editor

Der Spectrum Model Type Editor (MTE) wird dazu verwendet, dem Spectroserver neue Elemente hinzuzufügen und bestehende zu verändern. Er ermöglicht es, neue Model Types zu kreieren, MIBs zu importieren und um Attribute und Relationen zu verändern.

Abbildung 7: Spectrum Entwicklungsmöglichkeiten

Einführend will ich einige Begriffe, die wiederholt auftauchen werden, erläutern. Zum einen wären da die *Attribute*, welche durch ihre *Attribute ID* identifiziert werden. Sie sind Bestandteil eines jeden Model Types. Sie können Container für Daten mit verschiedenen Datentypen wie Integer, Boolean, usw. sein. Man unterscheidet zwischen internen und externen Attributen, wobei externe Attribute dadurch definiert sind, daß ihr Inhalt über ein Protokoll des DCM von einem zu managenden Gerät abgefragt wird. Als nächstes sollen die *Relations* erwähnt werden. Diese werden über ein *Relation Handle* identifiziert und beschreiben vorhandene Beziehungen zwischen Modellen. Die Model Types sind hierarchisch organisiert und stehen in der *Model Type Hierarchy*, welche *De-*

rivation Points bietet, um neue Model Types davon abzuleiten und dabei auf bereits vorhandene Entwicklungen zurückgreifen zu können.
Durch den Model Type Editor wird es möglich, auf die oben genannten Objekte des Spectroservers zuzugreifen.

Generic Information Block Editor

In den Generic Information Block (GIB) Views[3] des Spectrographen werden Informationen über Netzwerkkomponenten dargestellt. Die Darstellungsarten dieser Views lassen sich mit Hilfe des GIB Editors erzeugen und verändern. Der Zugriff auf Daten erfolgt hierbei über die Attribute IDs des Spectroservers. Somit ist auch der Zugriff auf externe Daten, z.B. aus SNMP-MIBs des entsprechenden Model Types, möglich. Abhängig vom Datentyp eines Attributs ist der Zugriff auf die verschiedensten graphischen Elemente möglich.
Diese graphischen Darstellungen lassen sich jedoch nicht nur mit dem GIB Editor erzeugen und verändern, sondern mit jedem beliebigen ASCII-Editor, da alle Views in simplen Text-Dateien gespeichert sind, die der Spectrograph vor deren Darstellung parsiert.
Allerdings sind zur Zeit keine Dokumentationen vorhanden, die diese Sprache genau beschreiben.
Der GIB Editor wird im göffneten GIB View über den Menüpunkt `Edit` aktiviert, er ist voll in die Oberfläche des Spectrographen integriert. Dann können neue Elemente ausgewählt und hinzugefügt werden. Dabei stehen die verschiedensten Repräsentationsmittel zur Verfügung, zum Beispiel numerische Anzeigen, Selektionselemente, Tortendiagramm, Graphen in Abhängigkeit von der Zeit, Tabellen usw.. Bei den Tabellen muß erwähnt werden, daß diese nicht alleine mit Hilfe des GIB Editors erzeugt werden können. Hier ist es unbedingt notwendig, diese mit Hilfe eines ASCII-Editors zu erstellen. Daten lassen sich in den entsprechenden Elementen nicht nur lesen, sondern, wenn es das Attribut erlaubt, auch schreiben. Stellt das Attribut in diesem Fall eine MIB-Variable dar, werden die Daten mittels SNMP in das entsprechende Gerät zurückgeschrieben.

[3]Generic Information Block Views werden im weiteren Verlauf Information Views genannt.

Icon Information Block Editor

Alle Darstellungen des Spectrographen enthalten Icons. Diese Icons stellen die Modelle dar, wie sie im Spectroserver vorhanden sind. Icons können hierbei komplette Netzwerkkomponenten wie z.B. ein Ethernet-LAN oder einen Router darstellen, aber auch nur Charakteristiken eines Modells beinhalten. Icons lassen es in den meisten Fällen zu, über entsprechend zugeordnete Menüpunkte oder über sichtbar gekennzeichnete Bereiche neue Ansichten im Spectrographen zu öffnen. Der Icon Information Block (IIB) Editor ermöglicht das Verändern dieser Icons entsprechend den Wünschen des Entwicklers.
Generell werden jedoch nach Erzeugung eines neuen Modells automatisch Standardicons verwendet, die dem look & feel von Spectrum entsprechen und den Ansprüchen des Anwenders im allgemeinen genügen.

2.4.3. Der Entwicklungszyklus

In diesem Abschnitt soll erklärt werden, in welchem Verhältnis die verschiedenen Einzelumgebungen des Level I Toolkits verwendet werden, und wie der Entwicklungszyklus für ein Management Modul aussieht.
In diesem Fall wird die Modellierung anhand des *Generic SNMP* (GnSNMP) Toolkits aufgezeigt. GnSNMP ist sowohl ein Management Modul für auf SNMP basierende Geräte, als auch ein Toolkit zum Entwickeln neuer Module. So ist in GnSNMP bereits die SNMP Standard MIB-II enthalten. GnSNMP bietet verschiedene Ableitungspunkte, die entsprechend der gewünschten Funktionalität des neu zu erstellenden Model Types zu wählen sind[14].
In den meisten Fällen wird es von Interesse sein, neben der MIB-II die Enterprise-MIB eines Geräts in die Datenbank des Spectroservers aufzunehmen. Darauf aufbauend ist es möglich, diese Daten zu visualisieren, das heißt im Spectrographen entsprechende Views zu kreieren.
Als erstes müssen also die firmenspezifischen MIBs dem Spectroserver zugänglich gemacht werden. Hierzu dient der Model Type Editor.
Oftmals sind herstellerspezifische MIBs bereits gruppiert, manche unterteilen MIBs funktionsorientiert, andere geräteorientiert. In vielen Fällen kann es auch sinnvoll sein, diese MIBs weiter zu unterteilen oder sie umzugruppieren, je nachdem, wie die Daten später zur Verfügung stehen sollen.

Vor dem Importieren einer MIB im MTE wird das Model Type gewählt, welches als Ableitungspunkt dienen soll.
Dieses ist für den MIB-Import immer *GnSNMPMibDerPt*. Darauf basierend wird ein neuer Model Type benannt und die entsprechende MIB-Datei importiert.
Sollen diese Daten später im Application View als Major Application Icon erscheinen, muß ein Model Type, abgeleitet vom Derivation Point *GnSNMPAppDerPt* erzeugt werden. Zur besseren Unterscheidung dieser beiden Model Types sollte der Name des von GnSNMPAppDerPt abgeleiteten Model Types auf *App* enden, der andere auf *Mib*.
Dem von GnSNMPAppDerPt abgeleiteten Model Type wird muß nun das vom GnSNMPMibDerPt abgeleitete als `Base Model Type` hinzugefügt werden. Dadurch ist es später möglich, im Spectrographen über das erzeugte Icon auf die Daten der MIB zurückzugreifen.
Beim Erzeugen von Minor Application Icons[4] muß statt des Ableitungspunkts GnSNMPAppDerPt `GnSNMPSubDerPt` gewählt werden, ansonsten ist der Vorgang dem der Erzeugung eines Major Application Icons identisch, allerdings muß im MTE zusätzlich eine *PROVIDES*-Relation zwischen den Minor Applications und der Major Application geschaffen werden.
Damit später beim Entdecken eines Geräts vom Spectrographen aus die entsprechenden Modelle instanziiert werden, d.h. die erzeugten Icons sichtbar werden, ist es noch notwendig, ein Attribut mit dem Titel *default_attr* im Attribute View des MTE für jeden erzeugten Model Type mit einem Wert zu versehen. Hierzu wird aus der importierten MIB ein Wert gewählt, der nicht Bestandteil einer Tabelle und möglichst ein Integer ist. Dazu muß die Attribute ID, welche die MIB-Variable repräsentiert, notiert werden, um dann als Wert für das default_attr eingetragen werden zu können. Schließlich muß das MT noch instanziierbar gemacht werden.
Wenn die Arbeiten mit dem MTE abgeschlossen sind, muß für jedes neu erzeugte Model Type das *mmbuild*-Script auf der Kommandozeile aufgerufen werden. Diese Script fragt interaktiv nach dem Namen des neuen Model Types, nach dem Namen des Base Model Types und nach der Developer ID.

[4] Minor Application Icons können nur dann instanziiert werden, wenn auch das dazugehörige Major Application Icon instanziiert werden konnte. Abbildung 16 zeigt dieses.

Anhand des Base Model Types entscheidet mmbuild, welche Spectrograph-Support-Dateien genutzt werden sollen und erzeugt diese für den neuen Model Type. So ist der Base Model Type sowohl für ein Major- als auch für ein Minor Application Icon *GnHubApp*[14].
Jetzt können der Spectroserver und der Spectrograph gestartet werden. Es muß geprüft werden, ob die Ausführungen im MTE korrekt waren und ein Model generiert werden kann. Das heißt, das Gerät, für welches entwickelt wird, muß über das Netzwerk erreichbar und seine SNMP-Community bekannt sein. Im günstigsten Falle werden alle Model Types erkannt und durch entsprechende Icons repräsentiert. Falls es Probleme gibt, muß wieder auf den MTE zurückgegriffen werden, um entsprechende Korrekturen vorzunehmen.
Falls die Model Types korrekt erkannt und instanziiert sind, ist es möglich, mit dem Design der Information Views zu beginnen, welches das Zeitaufwendigste in der Entwicklung eines Management Moduls ist. Dabei ist zu berücksichtigen, daß der Entwickler Schreibrecht auf die Dateien im Verzeichnis `$specroot/SG-Support/CsGib/$mtname` hat, die Spectrum Nutzerverwaltung hat darauf keinen Einfluß. Hier können die entsprechenden Elemente, welche die Daten visualisieren sollen, ausgewählt und plaziert werden.
Als weitere Option besteht die Möglichkeit, mit der Hilfe des Icon Information Block Editor die Icons, welche das Model im Spectrographen repräsentieren, zu verändern.

3. Access-Router

3.1. Definition

Access-Router[5], auch *Remote-Access-Server* (RAS) genannt, ermöglichen es einer nahezu unbegrenzten Anzahl von Nutzern, sich von außerhalb in Firmen oder Universitätsnetze, ebenso wie ins Internet einzuwählen und Verbindung zu anderen Rechnern aufzunehmen, um ihre Arbeit so auszuführen, als würden sie an ihrem gewohnten Arbeitsplatz sitzen. Access-Router verdrängen heute die früher üblichen Terminal-Server, welche den zu Hause angesiedelten PC zu einem simplen Terminal degradierten. Da diese Terminal-Server keine Kommunikationsprotokolle wie TCP/IP oder IPX unterstützten, war es nicht möglich, die heutzutage üblichen Client-Server Anwendungen zu benutzen.

3.2. Access-Router heute

Aufgrund der Nachfrage nach Internetzugängen und der zunehmenden Verbreitung von Telearbeitsplätzen in den letzten Jahren wurden Access-Router technologisch stark weiterentwickelt. Die derzeitig am Markt befindlichen Geräte ermöglichen es, Access-Lösungen für tausende von Nutzern gleichzeitig ebenso zur Verfügung zu stellen, wie es kleinen Arbeitsgruppen an verschiedenen Standorten ermöglicht wird, untereinander zu kommunizieren. Hierbei müssen diese Access-Lösungen im allgemeinen in eine bereits vorhandene Netzstruktur eingebaut werden, sich also der bereits vorhandenen Netzwerktechnologie und den benutzten Protokollen anpassen.
Für die Bereitstellung von Internetzugängen wiederum ist es notwendig, daß als Netzwerkprotokoll TCP/IP und für die Verbindung das *Point to Point Protocol* (PPP) zur Verfügung steht. Als Authorisierungsverfahren sind das *Password Authentication Protocol* (PAP) und das *Challenge Handshake Authentication Protocol* (CHAP) üblich. PPP inklusive PAP oder CHAP werden auch von gängigen Betriebssystemen wie Windows 95/NT, MacOS und Linux unterstützt, der Vorgänger *Serial Line IP* (SLIP) kommt in modernen Installationen üblicherweise nicht mehr zum Einsatz.

[5] Remote-Access-Router werden aus Marketinggründen oft auch Remote-Access-Switches genannt, obwohl sie eigentlich auf OSI-Layer 3 arbeiten.

Wird erhöhte Sicherheit gefordert, ist die Verwendung von Call-back, dem automatischen Rückrufen des Access-Routers nach einem eingegangenen Ruf, oder die Überprüfung der eingehenden Telefonnummer (Call-ID), falls möglich, nötig. Des weiteren wird von diesen modernen Geräten erwartet, daß die früher gerade beim Einsatz sogenannter Modembatterien auftretenden Probleme wie das Hängenbleiben von Modems in undefinierten Zuständen, inaktive, aber weiterhin offene Verbindungen und vieles mehr von diesen Lösungen automatisch behandelt werden. Gemäß den Anforderungen ist es möglich, aus einer großen Palette von Produkten das passende Gerät auszuwählen.
Man kann folgende Geräteklassen unterscheiden:

- SOHO[6] Lösungen mit Unterstützung für wenige Nutzer, die sich via externer analoger oder digitaler Verbindung in ein LAN einwählen.
- MID-Range Access-Router, angesiedelt im Bereich kleinerer und mittlerer Unternehmen, die typischerweise verschiedene Netzwerkprotokolle verstehen und bis zu 30 Nutzer gleichzeitig unterstützen können.
- High-End Lösungen, wie sie von Providern benötigt werden, mit einem weiten Bereich von Möglichkeiten, wie Unterstützung der verschiedensten WAN-Schnittstellen, Fernwartbarkeit, Modularität, integriertem Firewall und der Fähigkeit, eine sehr große Anzahl von Nutzern gleichzeitig unter Beanspruchung völlig unterschiedlicher Anforderungsprofile Verbindung aufnehmen zu lassen.

Entsprechend den Anforderungen beim Einsatz eines Access-Routers ist es möglich, aus einer Gerätepalette auszuwählen, die die in Tabelle 2 aufgelisteten Fähigkeiten unterstützt.
Beim Access-Routing muß der Router Nutzerinformationen erhalten, da diese zur Authentifizierung der Einwählenden notwendig ist. Diese werden vom ISDN- oder Modemnutzer meistens mittels PAP oder CHAP übermittelt. Da ein Access-Router nur wenige Accounts speichern kann und diese im allgemeinen den Administratoren vorbehalten sind, ist es notwendig, diese Authorisierungsinformationen von einem Authentication-Server wie dem *Remote Acces*

[6]SOHO (Small Office Home Office) ist ein Begriff, der sich durch die Verbreitung von Telearbeitsplätzen gebildet hat.

Tabelle 2: Fähigkeiten von Access-Routern

Anforderung	**Beispiele**
LAN Schnittstellen	Ethernet, Fast Ethernet, ATM, Token Ring, FDDI
WAN Schnittstellen	Analog, ISDN S0, E1/T1, Frame-Relay
Unterstützte Protokolle	TCP/IP, IPX, AppleTalk, DECnet, SNA
Anzahl gleichzeitiger Sitzungen	Von einer bis ca. 1000
Komprimierung	Software/Hardware-Kompression
Remote-Access-Routing	Dynamische IP-Adressierung, Filter
Skalierbarkeit	Modularer Aufbau für Geräteerweiterungen
Nutzer Authentifizierung	Unterstützung für RADIUS- und TACACS-Systeme
Sicherheit	Eingeschränkter Zugriff, Nutzerüberprüfung via Call-ID, PAP, CHAP, Call-back
Management	Telnet, SNMP, proprietär
Internet Gateway	IP-Routing, Adresse- und Portfilter, PPP- und SLIP-Unterstützung

Dial In User Service (RADIUS) oder von einem *Terminal Access Controller Access Control System* (TACACS) zu erhalten.

RADIUS

RADIUS ist ein Standard für die Authentifizierung und das Accounting von Remote Access Dial-In Nutzern. RADIUS basiert auf dem Client/Server-Modell, wobei der Client hierbei der Access-Router, also der Einwahlpunkt ist. Der Client stellt die Anfragen an einen zentralen RADIUS-Server und gibt aufgrund der dort vorhandenen Nutzerinformationen entsprechende Resourcen auf das Netzwerk frei oder verhindert den Zugriff. Prinzipiell läuft die Anmeldeprozedur folgendermaßen ab:
Bei einem Login über RADIUS meldet sich der Benutzer wie gewohnt mit Namen und Paßwort über seinen Clienten, z.B. Windows95, am Access-Router (z.B via PAP oder CHAP) an. Dieser durchsucht seine lokalen Einträge nach dem Nutzernamen. Findet er ihn nicht, schickt er den Namen und das Paßwort in einem `Access-Request`-Paket zu einem RADIUS-Server. Der Ser-

ver konsultiert seine eigenen Daten und bewilligt oder verweigert den Zugriff (`Access-Accept` oder `Access-Reject`). Die Übertragung der Daten zwischen Server und Client erfolgt hierbei verschlüsselt. In Bild 8 ist als Beispiel der Eintrag eines Nutzers in die RADIUS-Users Datei zu sehen.

```
twiik   Password = "UNIX", Caller-Id = "003021750913"
        Ascend-Send-Auth = Send-Auth-PAP,
        Framed-Protocol = PPP,
        User-Service = Framed-User,
        Framed-Address = 141.45.240.1,
        Framed-Netmask = 255.255.255.255,
        Ascend-Metric = 2,
        Framed-Routing = None,
        Ascend-Idle-Limit = 720,
        Ascend-Dial-Number = 003021750913
```

Abbildung 8: RADIUS Nutzereintrag

RADIUS-Server sind in einer großen Anzahl am Markt verfügbar. Das beginnt mit Referenzimplementierungen für UNIX nach RFC 2138[9] und RFC 2139[10], und geht bis hin zu Versionen mit firmenspezifischen Erweiterungen inklusive Datenbankschnittstelle.

TACACS

TACACS und die daraus hervorgehenden Weiterentwicklungen XTACACS und TACACS+ bieten ebenso Features wie RADIUS. Das größte Problem der ersten beiden TACACS-Versionen war die unverschlüsselte Datenübertragung, was im Falle der Passwortübermittlung als unsicher angesehen werden muß. Während das originale TACACS-Protokoll ein offener Internet-Standard war, entwickelte sich TACACS+ durch Erweiterungen seitens der Firma Cisco immer mehr zu einem proprietären Protokoll. Dies ist auch ein Grund für die weite Verbreitung von RADIUS.

3.3. Ascend-Access-Router

Viele der in Tabelle 2 genannten Anforderungen werden durch die Produktpalette der Firma Ascend abgedeckt. Als Beispiel hierzu will ich einen Ascend

Access-Router vorstellen. Hierbei handelt es sich um den MAX4000. Dieser ist bei vielen kommerziellen Providern im Einsatz. Aber auch das Rechenzentrum der FHTW-Berlin und viele andere Universitäten in Deutschland setzen dieses Produkt ein.
Ein MAX4000 bietet folgende Schnittstellen:

- 1* Ethernet 10Mbit/s
- 4* E1 (S2M) je 2Mbit/s
- 1* V.35 8Mbit/s für Frame Relay
- 6* Expansion Slot u.a. für Modemkarten

Das Gerät kann bis zu 120 digitale (ISDN) oder 96 analoge Sessions gleichzeitig bedienen, wobei Mischbetrieb möglich ist. Die analogen Sessions werden durch digitale Modemkarten ermöglicht. Unterstützte Standards sind zur Zeit V34+ und K56Flex. K56Flex-Karten werden durch ein Softwareupgrade in Zukunft auch den neuen Standard V.pcm unterstützen. Die Einschränkung auf 96 analoge Sessions ist durch die digitalen Modem-Karten gegeben, die derzeit nur bis zu 16 analoge Zugänge je Karte zur Verfügung stellen.
Im Gegensatz zu einem MAX4000 unterstützt ein Access-Router der Klasse MAX TNT bis zu 672 Sessions gleichzeitig.
Alle gängigen Protokolle wie TCP/IP, IPX und AppleTalk werden unterstützt. Im Bereich des Routing ist neben statischem Routing das Verwenden von RIPv1/v2 und OSPF[7] möglich. Zum WAN-Verbindungsaufbau sind ebenfalls alle Standards implementiert, u.a. Async PPP, Sync PPP und SLIP. STAC-Kompression[8] über ISDN ist ebenso möglich wie Multilink PPP, als weitere Steigerung DBA (Dynamic Bandwidth Allocation) in Verbindung mit Multilink Protocol Plus (MP+). Somit ist es möglich, bei komprimierbaren Daten laut Dokumentation bis zu 512 Kbps über eine 128Kbps-Line, das entspricht einem ISDN S0-Anschluß, zu übertragen. Des weiteren können allen PCs im

[7] RIP (Routing Information Protocol) und OSPF (Open Shortest Path First) sind gängige Protokolle für dynamisches Routing in TCP/IP LAN-Umgebungen.

[8] STAC ist ein sich herausbildender Standard zur Datenkompression bei ISDN-Verbindungen.

LAN unter Windows 95/NT über die Ascend-Software MAXDial virtuelle Modems zur Verfügung gestellt werden, so auch zum FAX-Versand.

3.4. Management

Durch den großen Funktionsumfang von Access-Routern werden ebensolche Ansprüche auch an das Management gestellt. Diese sind sicherlich durch die vielfältigen Einsatzmöglichkeiten dieser Geräte weit dehnbar, doch wollen wir uns hier auf das Einsatzgebiet des Dial-In fokussieren. Diese Dial-In-Lösungen werden von den *Internet Service Providern* (ISPs) ebenso genutzt wie von Unternehmen, die ihren Mitarbeitern Telearbeitsplätze zur Verfügung stellen. Access-Lösungen stellen sich in einem Netzwerk ebenso wie andere aktive Netzwerkkomponenten dar. Sie haben eine Anzahl von Schnittstellen, verwalten intern verschiedene Tabellen wie zum Beispiel Informationen über das Routing. Dennoch heben sie sich von ihren Artverwandten aus dem LAN-Umfeld ab. So verfügen Access-Router über eine sehr große Anzahl von Schnittstellen und sie unterliegen ständig wechselnden Verbindungen. Diese Verbindungen können dabei die verschiedensten Protokolle, wie zum Beispiel IP oder AppleTalk, nutzen. Auch die Bandbreite dieser Verbindungen variiert, zum einen unter den verschiedenen Modemgeschwindigkeiten, andererseits erlauben Protokolle wie MP+ je nach Anforderung des Teilnehmers das dynamische Bündeln mehrerer ISDN-Kanäle.
Aufgrund dieser Tatsachen wollen wir die besonderen Anforderungen an das Management von Access-Lösungen nach den in Kapitel 2.1 auf Seite 11 genannten Punkten zusammenfassen.
Das Fehlermanagement trifft hier natürlich auch bei Access-Lösungen zu und ist ein sehr wichtiger Punkt. Gerade im Umfeld der Modemnutzung ist immer wieder von Problemen zwischen verschiedenen Geräte untereinander und zu Access-Routern zu hören. Dies beginnt mit ungewünschten Verbindungsabbrüchen, geht über schlechte Übertragungsraten hin zur absoluten Unverträglichkeit zwischen den Geräten. Hier ist es für den Kommunikationsanbieter sehr wichtig, einen Einblick über Verbindungszustände und Abbruchursachen zu erhalten. Dies ermöglicht dem ISP, dem Anwender Hinweise zur Umkonfiguration des Modems selbst geben zu können.

Ein anderer Punkt sind generelle Fehlkonfigurationen beim Nutzer. Das können zum Beispiel falsch eingestellte IP-Adressen, Subnetzmasken, Kommunikations- und Authorisierungsprotokolle sein. Einsicht in Daten über den Verbindungsaufbau lassen auf solche Ursachen rückschließen.
Da Access-Router abhängig von Authorisierungs-Servern sind, ist deren Verfügbarkeit ebenso wichtig wie die der Zugangsgeräte selbst. So ist bei einem allumfassenden Management auch die Überwachung der Kommunikation zu den Authorisierungs-Servern ein wichtiges Kriterium. Dies betrifft ebenso die Kommunikation mit den Accounting-Servern, falls der ISP eine volumen- und zeitabhängige Abrechnung durchführt.
Überschreiten die Anfragen an den Access-Router die Anzahl der verfügbaren Ports, wollen wir diesen Zustand als Fehler bezeichnen. Dieser Fehler betrifft sowohl die Nutzer von analogen, als auch die von digitalen Zugängen. Eine nicht ausreichende Anzahl von Ports hat das unbeliebte Besetztzeichen für den Nutzer zur Folge, was bei kommerziellen Anbietern schnell zum Verlust der Kunden führt. Ein großes Problem für ISPs sind hierbei jedoch Spitzenauslastungen zu bestimmten Tageszeiten, in Deutschland bedingt durch die Tarifpolitik der Telekom. Hier lassen sich jedoch nach Analysen Entlastungen schaffen, zum Beispiel durch günstigere Tarife seitens der ISPs für den Anwender zu Zeiten von Nichtauslastung. Diese Umstände machen es notwendig, Statistiken über die Auslastung der Modem- und ISDN-Ports der Geräte zu führen, um rechtzeitig Erweiterungen einplanen zu können. Frühzeitige Trendanalysen sind hierbei sehr wichtig, denn im Bereich der Telekomunikationsanbieter sind bei der Bereitstellung großer Zugangskapazitäten oft längere Wartezeiten einzuplanen.
Oft beklagen sich die Anwender von Einwahlzugängen über schlechte Datenübertragungsraten, womit wir beim Performancemanagement angekommen sind.
Diese Probleme können die verschiedensten Ursachen haben. Zum einen wäre hier eine zu geringe Kapazität im Backbone des Providers zu vermuten, das bedeutet er bietet mehr Einwahlzugänge an, als seine Anbindung an das Internet bewältigen kann.
Dies wird schnell durch den rasanten Ausbau von Zugängen aus dem Auge verloren.

So sollte immer der Datenverkehr, der über die Zugangsgeräte fließt, gemessen werden. Denn auch Standard-Ethernet mit seinen theoretisch 10MBit/s wird beim Einsatz mehrerer Geräte mit hohen Portdichten schnell zum Engpaß.
Auch kann ein schlechtes Hardware- oder Firmware-Design eines Access-Routers zu Engpässen bei hoher Geräteauslastung führen. Die Tolly-Group berichtet in einem ihrer Tests[16] über Durchsatzeinbrüche bei vielen gleichzeitigen Verbindungen. So fällt in diesem Test bei einem der Access-Router der Durchsatz je Modem von 55,57 kbit/s bei einer benutzten Verbindung auf 28,62 kbit/s bei 288 Verbindungen.
Gerade das Accounting rückt bei vielen Anbietern in den Vordergrund, da diese die Nutzung der Zugänge sowohl zeit- als auch volumenabhängig in Rechnung stellen. Gerade dies stellt im traditionellen Sinne eine hohe Anforderung an das Management dar. So ist es zum Beispiel nicht wie im LAN-Bereich möglich, den Datenverkehr über einen bestimmten Port einer Abteilung oder einem Nutzer zuzuordnen. Im Access-Bereich werden die Ports entsprechend der Verfügbarkeit vergeben. Deshalb wurde das Accounting mittels RADIUS eingeführt. Dies hat jedoch den Nachteil, daß diese Informationen nicht unbedingt über SNMP verfügbar sind.
Auch ein Sicherheitsmanagement ist oft von Interesse. So unterstützen Access-Router hier oft einen weiten Bereich von Möglichkeiten, denn sie sind oft die einzige Schnittstelle eines Netzes zur Außenwelt. Deshalb integrieren Hersteller oft sogenannte Firewalls in diese Geräte. Diese ermöglichen das Definieren von einfachen Portfiltern, bis hin zum Ausklammern ganzer Internet Adress-Bereiche.
Um die genannten Anforderungen zu erfüllen, lassen es die meisten Access-Router ebenso wie andere aktive Netzwerkomponenten zu, remote gemanagt zu werden. Ein Management über telnet soll hier als nicht zeitgemäß betrachtet werden, was nicht heißt, daß es für alle auf dem Markt vorhandenen Produkte selbstverständlich ist. Gängig ist das Management via SNMP. Die Nutzbarkeit des Managements ist abhängig von der Einbettung des Funktionsumfangs des Geräts in die herstellerspezifische MIB. Eine wichtige Eigenschaft ist auch die Möglichkeit, nicht nur statistische Werte aus dem Gerät herauszulesen, sondern auch Konfigurationen via SNMP vornehmen zu können, d.h. Werte in das Gerät zurückschreiben zu können.

4. Entwurf eines Managementmoduls

4.1. Anforderungen

Als Anforderung an ein Management des Ascend-Access-Routers im Rechenzentrum der FHTW stehen eine Anzahl von Gesichtspunkten. Durch ein Management soll ein besserer Einblick in die Aktivitäten des MAX4000 ermöglicht werden. Dieser bezieht sich durch den Einsatz als Dial-In-Knoten, im speziellen auf die Aktivitäten der Nutzer, auf mögliche Probleme bei der Einwahl, auf das Verhältnis der Nutzung von ISDN- und Modem-Ports, Datenraten der Verbindungen, Durchsätze im Netz, der Verfügbarkeit von freien Ports, Erkennung von Einwahlproblemen und deren Ursachen. Dabei soll möglichst eine komprimierte Darstellung all dieser Informationen geschaffen werden, so daß möglichst schnell aussagekräftige Fakten erkannt werden können. Das Ziel besteht also nicht nur in einer akkuraten Darstellung der Werte, sondern auch darin, nützliche und übersichtliche Ansichten zu schaffen.
Zur Verwirklichung dieser Anforderungen gibt es hierbei verschiedene Möglichkeiten. Die erste wäre die komplette Programmierung einer eigenständigen Managementapplikation, speziell für Ascend-Router z.B. in C++, die nächste eine webbasierte Anwendung, z.B. unter Nutzung des CGI (Common Gateway Interface), der Entwurf eines plattformunabhängigen Programms in Java, oder die Implementierung eines Moduls für eine gängige Managementplattform. Letzteres ist bereits durch die Aufgabenstellung der Diplomarbeit vorgegeben. Als Managementplattform kommt Cabletron Spectrum zum Einsatz. Die Verwendung dieser vorhandenen Managementplattform hat den Vorteil, daß die gesamte Umgebung dem Netzwerkadministrator schon bekannt ist, er muß sich nicht erst mit einem neuen Programm und dessen Eigenheiten vertraut machen. Auch erscheint das Gerät im Netzkontext, es ist zu sehen, wo genau es sich im Netz befindet.
Für den Entwickler wiederum ist es dann nicht nötig, alle bereits vorhandenen Elemente eines Managementsystems (wie z.B. Protokollinterface für SNMP, Datenspeicherung und Verwaltung, graphische Präsentation), neu zu entwickeln, es kann auf die bereits vorhandenen Entwicklungen zurückgegriffen werden. Dies allerdings erfordert Wissen über die Funktionsweise des zu

erweiternden Managementsystems und dessen Aufbau, ebenso wie über die Möglichkeiten und auch Einschränkungen, die die Entwicklungsumgebung dem Entwickler auferlegt. Allgemein läßt sich sagen, daß sich nach Einarbeitungszeit die Entwicklungszeit für ein gerätespezifisches Management verkürzt und es so möglich wird, sich auf das Wesentliche zu konzentrieren.
Zur Zeit erscheinen auf dem Markt verschiedene der genannten Applikationen, wobei sich alle noch mehr oder minder im Entwicklungsstadium befinden. Auch Ascend ist seit Mitte 1997 mit dem Programm `NetClarity` als Managementplattform nicht nur für Ascend-Produkte vertreten. Eine Demoversion allerdings zeigte, daß diese den Anforderungen des Rechenzentrums nicht genügt, da es in den meisten Teilen nicht mehr als die blanken SNMP-Daten anzeigt. NetClarity konnte aber als Anregung für Elemente genutzt werden.
Ohne eine spezielles Ascend Management Modul für Spectrum ist es bereits möglich, die in MIBII enthaltenen Werte zu managen. Das ist eine gute Möglichkeit, einen ersten Einblick in die Aktivitäten des Geräts zu erhalten, ist aber bei weitem nicht das, was nötig und auch möglich ist.

4.2. Konzeption

Die Basis für die Konzeption des Management Moduls liegt in den Enterprise MIBs.
So werden im vorliegenden Fall die Ascend-MIBs genutzt, um mit Hilfe der Level I Tools ein Management Modul für den vorhandenen MAX4000 und Cabletron Spectrum zu implementieren. Da Ascend seine MIBs funktionsorientiert gruppiert, können mit Hilfe dieses Moduls auch andere Ascend-Router wie zum Beispiel ein MAX TNT gemanaged werden.
Hierbei sollen die einzelnen MIBs durch entsprechende Icons im Application View widergespiegelt werden. Hinter jedem Icon verbergen sich dann weitere Views, die es ermöglichen sollen, sowohl Informationen über das Gerät zu erhalten, als auch Konfigurationen vornehmen zu können. Die Repräsentation der Daten kann dabei mittels verschiedenster Elemente erfolgen. Hierbei ist man auf das Angebot des GIB Editors beschränkt, was aber in den meisten Fällen ausreichen sollte. Durch den objektorientierten Ansatz von Spectrum ist es nicht notwendig, die in MIB-II vorhandenen Daten nochmals zu modellie-

Tabelle 3: Geplante Funktionen

MIB	**Funktionen**
Ascend-MIB	TFTP Down/Up-Load, SystemReset
Ascend-Call-MIB	Call Status (Data-Rate)
Ascend-Event-MIB	Liste aller Events, Anzahl von Calls
Ascend-LanModem-MIB	Verfügbarkeit der Modems
Ascend-Session-MIB	Aktive Sessions
Ascend-Radius-MIB	Informationen zur Radiusnutzung

ren. Durch Verwendung von GnSNMP als Ableitungspunkt sind diese bereits verfügbar. Die Selektion der Daten aus den Ascend-MIBs, die in elf verschiedene Gruppen unterteilt sind, erfolgt in erster Linie nach den Kriterien der Nutzung und somit auch der Testbarkeit im Rechenzentrum der FHTW. Das Gerät wird hier nur zur Einwahl via Modem und ISDN bereitgestellt, Möglichkeiten wie Kanalbündelung, Callback usw. sind nicht aktiviert.
Die ausgewählten MIbs sind in Tabelle 3 zu sehen. Da die MIBs *adslcap.mib, firewall.mib, mcast.mib, sdsl.mib* und *wan.mib* derzeit von dem vorhandenen MAX4000 nicht unterstützt werden, wurden sie nicht weiter beachtet.
Die Repräsentation der Daten soll im Application View des Spectrographen erfolgen. Hierbei soll ein Icon das sogenannte Major Application Icon die Daten der Ascend-MIB beeinhalten. Da die anderen MIBs nur dann vorhanden sein können, wenn auch die Ascend-MIB existiert, werden diese durch Minor Application Icons vertreten. Die Grundstruktur der Ascend-MIBs soll beibehalten werden, was die Weiterentwicklung des Management-Moduls vereinfachen wird. Desweiteren erleichtert dies Kennern von Ascend-Routern die Navigation, da die Struktur bereits bekannt ist. Die Basis der Entwicklung stellen die MIBs des MIB-Release 970903 dar. Diese sind auf dem ftp-Server[3] der Firma Ascend entsprechend der Firmwareversion des Gerätes verfügbar. Ein Slotmanagement der Expansionslots wird im Rahmen dieser Diplomarbeit nicht in Betracht gezogen, da dies nicht Hauptfocus des Themas der Arbeit ist und aus Zeitgründen nicht realisierbar scheint. Ebenso ist es nicht vorgesehen, ein Frame Relay Management mit in die Entwicklung einzubeziehen, da bereits ein Frame Relay Management Modul für Spectrum auf dem Markt verfügbar ist.

Als Beispiel für eine der MIBs will ich einen Ausschnitt aus der Ascend-Call-

```
callStatusHighWaterMark OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION     "The highest number of channels ever used on the
                     wide area network since power up. Its value can
                     be set to zero with a SET command."
    ::= { callStatusGroup 3 }
callCurrentAnalogOutgoing OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "The number of current analog outgoing calls
                     is returned."
    ::= { callStatusGroup 4 }
callCurrentAnalogIncoming OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "The number of current analog incoming calls
                     is returned."
    ::= { callStatusGroup 5 }
```

Abbildung 9: Ausschnitt aus der Ascend-Call-MIB

MIB wählen, um interessante Werte zu zeigen. In Abbildung 9 sind drei für den Betreiber eines Ascend-Access-Routers sehr interessante Werte zu sehen. Zum einen `callStatusHighWaterMark`, welches die maximale Auslastung an Kanälen im System seit Systemstart zeigt. Dieser Wert ist schreibbar, was bedeutet, daß er sich auf Null zurücksetzen läßt. Dieser Wert ermöglicht es auch, trotz niedriger Polling-Zeit einen Überblick über Spitzenwerte zu bekommen. Die Werte `callCurrentAnalogOutgoing` und `callCurrentAnalogIncoming` geben Aufschluß über die derzeitige Auslastung der im Gerät vorhandenen Modems, im weiteren Verlauf der MIB werden ebenso Daten für Verbindungen vom Typ Digital (ISDN) und Frame-Relay bereitgestellt.
Durch Sichten der MIBs entstand ein grober Überblick über mögliche Funktionen, wie sie auch in Tabelle 3 gezeigt werden. Als weiteres Hilfsmittel dienten

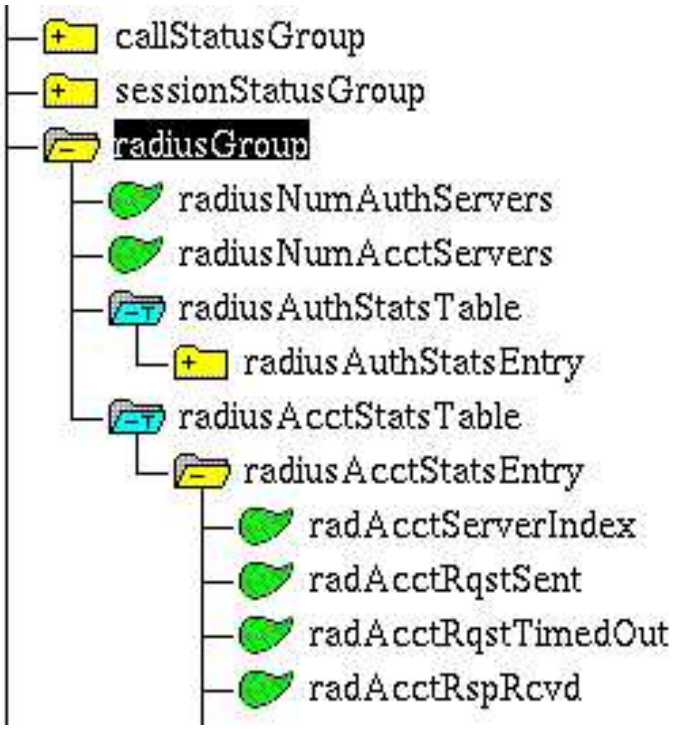

Abbildung 10: Ascend-MIB im MIB-Browser

hierbei auch Cabletron MIBtools, ein Bestandteil von Spectrum, deren MIB-Browser MIBs wie in Bild 10 baumartig darstellt.

5. Implementierung eines Managementmoduls

Entsprechend dem Entwurf soll nun die Umsetzung in ein Management Modul für Cabletron Spectrum erfolgen.
Dabei ist nach dem in Kapitel 2.4.3 auf Seite 29 geschilderten Entwicklungszyklus vorzugehen. Wichtig ist hierbei, daß vor dem eigentlichen Begin die Developer-ID der FHTW in die Datenbank des Spectroservers geladen wird. Für zukünftige Entwicklungen hier noch der Hinweis, daß immer alle unter dieser ID erzeugten und weitergegebenen Management Module in der Datenbank verbleiben müssen, um zu gewährleisten, daß keine Attribute IDs, Relation Handles usw. mehrfach vergeben werden.

5.1. Das Ascend Router Modul

Als erstes soll ein Modul erzeugt werden, welches sämtliche Ascend-Router abdeckt und sich später im Spectrographen als Icon wie in Bild 11 im Topology View zeigt.

Abbildung 11: Ascend-Router Icon im Spectrograph

Dazu ist es notwendig, im MTE einen neuen Model Type zu erzeugen, der von GnSNMPDev abgeleitet wird. Damit Spectrum später beim Modellieren erkennt, daß es sich um einen Ascend-Router handelt, müssen noch einige Attribute im Attribute-View des MTE mit Werten versehen werden. Signifikant ist in diesem Fall das Attribut `system_oid_verify`, welcher mit dem Wert `1.3.6.1.4.1.529.2.1` der OID der MIB-Variable `slotnumber` aus der Ascend-MIB versehen wird. Spectrum nutzt dieses Attribut, um festzustellen, ob für ein zu managendes Gerät ein gerätespezifisches Management Modul

vorhanden ist oder ob GnSNMP zu verwenden ist. Abschließend für diesen Teil muß noch das `mmbuild`-Script, wie in Kapitel 2.4.3 auf Seite 30 beschrieben, aufgerufen werden. Als Abbleitungstyp muß hier *GnSNMPDev* gewählt werden.

5.2. Der Application View

Der Zugriff auf die graphischen Darstellungen der Daten des Ascend-Access-Routers erfolgt über die Application Icons im Application View des Ascend Router Modells. Dieser ist in Abbildung 16 zu sehen.
Als erstes muß das Major Application Icon *AscendApp* kreiert werden, daraufhin die der Minor Application Icons, die zum Major Application Icons in der PROVIDES-Relation stehen. Dazu werden im MTE die neuen Model Types anhand der von GnSNMP angebotenen Ableitungspunkte geschaffen und die Ascend-MIBs importiert. Daraufhin sind noch einige Attribute der Model Types mit Einträgen zu versehen[14]. Sehr wichtig ist hierbei der Eintrag des default_attr im Attribute View des MTE für jeden Model Type. Dieser stellt das Discovery Attribut entsprechend system_oid_verify dar. In Tabelle 4 sind die Discovery Attribute der jeweiligen Application Icons zu sehen, einzutragen sind die entsprechenden Attribute-IDs.

Tabelle 4: Discovery Attribute

Application Icon	**Variable**	**OID**
AscendApp	slotNumber	1.3.6.1.4.1.529.2.1
CallApp	callStatusMaximumEntries	1.3.6.1.4.1.529.11.1
EventApp	eventMaximumNumberOfEvents	1.3.6.1.4.1.529.10.1
LModemApp	availLanModem	1.3.6.1.4.1.529.15.1
SessionApp	ssnStatusMaximumSessions	1.3.6.1.4.1.529.12.1
RadiusApp	radiusNumAuthservers	1.3.6.1.4.1.529.13.1

Diese Werte wurden mit der Hilfe des MIB-Browsers aus den MIBtools von Spectrum vor dem Eintrag auf Funktionalität geprüft. Nachdem die Arbeiten mit dem Model Type Editor abgeschlossen waren, mußte das `mmbuild`-Script hier für jeden erzeugten Model Type aufgerufen werden.

5.3. Die Information Views

Die Information Views – oder genau gesprochen: die GIB Views – können erst nach erfolgreichem Entdecken der Application Icons gestaltet werden. Hierzu werden den Application Icons entsprechend Menüpunkte hinzugefügt, um auf die GIB Views zugreifen zu können. In Tabelle 5 ist die Menüstruktur in Abhängigkeit von den Icons zu sehen.

Tabelle 5: Menüstruktur

Application Icon	**Menü**
AscendApp	TFTP
CallApp	CallList, Utilization
EventApp	EventList
LModemApp	LmodemView
RadiusApp	Accounting, Authorization
SessionApp	SessionList

Im folgenden will ich die implementierten Funktionen zeigen und dabei exemplarisch jeweils einmal auf Details der verwendeten graphischen Elemente und deren Eigenheiten eingehen. Die Spectrum Standardkonfiguration der Polling-Rate von zehn Sekunden wurde dabei durchgängig übernommen.

AscendApp

Durch das AscendApp Icon besteht Zugriff auf zwei neue Information Views. Das in Abbildung 17 gezeigte *Ascend TFTP Configuration View* bietet unter anderem die Möglichkeit, einen Firmwareupgrade des Ascend-Routers einzuleiten. Dazu müssen die IP-Adresse des TFTP-Servers, ggf. dessen Port-Nummer und der Dateiname der Firmware mit komplettem Pfad eingegeben werden. Dazu werden `Textfelder` mit der Read-Write-Option genutzt. Um das Upgrade nach Eingabe der Werte zu starten, muß der `Option Button` mit dem Titel *TFTP Operation* mit der Maus in die Stellung *loadcode* gebracht werden. Da der Option Button mit dem Flag `immediate write` versehen wurde, ist es nicht notwendig, im Hauptmenü den Schalter für den Punkt `Save all Changes` zu aktivieren. Dies hat den Nachteil, daß der Option Button nicht

den aktuellen Wert aus dem Gerät ausliest und anzeigt. Deshalb wurde noch ein Option Button mit dem Flag `read only` plaziert, der den Wert der letzten Operation anzeigt. Ein weiterer Option Button *Status* zeigt den Rückgabewert der eingeleiteten TFTP-Operation an. In Bild 12 sind die verwendeten graphischen Elemente dieses Views abgebildet.

Abbildung 12: Ausschnitt aus dem TFTP View

Option-Buttons im Allgemeinen bieten die Möglichkeit, Werte die gelesen oder geschrieben werden sollen und als Zahlen repräsentiert werden, in eine für Menschen verständliche Form zu bringen. Am Beispiel des Option Button *TFTP Operation* sieht dies wie folgt aus:

```
1, save, 2 , restore, 3, saveAll, 4, saveMib, 5, saveAllMib, 6, loadcode
```

Die genaue Bedeutung der einzelnen Kommandos ist der Dokumentation[1] des MAX4000 zu entnehmen.
Der *Ascend Configuration View* in Bild 18 ermöglicht es, Information über die Version der MIB im Ascend-Router zu erhalten. Ebenfalls in diesem View ein Button, der dazu dient, einen Neustart des MAX4000 zu erzwingen. Dieser Punkt wird im Moment nicht vom MAX unterstützt, wurde aber in Hinblick auf baldige Funktionalität implementiert. Da ein Slotmanagement nicht Bestandteil dieser Arbeit ist, blieben alle Slotvariablen dieser Gruppe unbeachtet, dies wird ebenso in den kommenden Gruppen der Fall sein.

CallApp

Hinter dem Icon *CallApp* verbergen sich zwei neu erzeugte Information Views. Zum einen der *Ascend Call View*, wie in Abbildung 19 zu sehen. In der erzeugten Tabelle ist in der ersten Spalte die Startzeit des Rufes (starting time) zu sehen. Das Problem hierbei ist, daß die Ascend-Call-MIB diesen Wert in Sekunden seit Gerätestart angibt, aber die Entwicklungsumgebung es nicht

gestattet, diesen Wert in einer Tabelle in die entsprechende Uhrzeit umzuwandeln[9]. Deshalb wird hier auch direkt der Wert in Sekunden angezeigt. In der nächsten Spalte ist die Nummer des benutzten Interface aus der MIB-II (*ifindex*) zu sehen, dann der Typ des Rufes und dessen Verbindungsgeschwindigkeit (*DataRate*). Nicht zu sehen, da von der aktuellen Firmewareversion des MAX4000 nicht unterstützt, ist die aktuelle Xmit-Rate. Als letzte Spalte ist die *ReferenceNumber* zu erkennen. Über diese Zahl ist die Verbindung zu Informationen aus anderen Views möglich, da die ReferenceNumber auch in anderen Bestandteilen der Ascend-MIBs vorkommt und somit Verknüpfungen zuläßt.
In dieser Tabelle werden die verschiedensten Elemente verwendet, wie in Kapitel 2.4.2 auf Seite 28 erwähnt, müssen Tabellen mit einem ASCII-Editor erstellt werden. In Abbildung 13 ist entsprechend dem View die ASCII-Definition der Tabelle zu sehen, ich will aber auf eine genauere Erläuterung der Elemente verzichten, diese Information ist in [12] enthalten. Der zweite View, der über das

```
{
ColInt.Ttm(10,12,0x02a3007c,"Time",BLUE,252,20,8,"")
ColInt.TIn(200,12,0x02a30084,"ifindex",BLUE,252,10,8,"")
ColInt.TEm(350,12,0x02a30086,"Type",BLUE,252,15,8,"","," \
1,outgoing call,2,incoming call",)
ColInt.TIn(500,12,0x02a3007e,"DataRate",BLUE,252,10,8,"")
ColInt.TIn(650,12,0x02a30087,"XmitRate",BLUE,252,10,8,"")
ColInt.TIn(800,12,0x02a3007d,"ReferenceNumber",BLUE,252,18,8,"")
}
```

Abbildung 13: Tabelle in einem Information View

Minor Application Icon *CallApp* aufzurufen ist, hat den Titel*Ascend Call Utilization View* und wird in Abbildung 20 gezeigt. Hier wird es möglich, einen genauen Blick auf die ein- und ausgehenden Rufe zu werfen. Gleich zu Anfang folgende Einschränkung: Da die gegenwärtig freigegebene Firmware des MAX4000 die meisten der verwendeten MIB-Variablen nicht unterstützt, sind in Abbildung 20 die betroffenen Elemente rot gekennzeichnet. Im linken Bereich ist die *HighWaterMark* Gruppe zu erkennen. Zur Anzeige des aktuellen

[9]Es gibt keine Möglichkeiten, die von SNMP gelieferten Daten in einer Tablle mit anderen Werten zu verrechnen.

Werts wurde das `Tachometer, bar` Element verwendet. Dieser Wert gibt den Maximalwert der benutzten Analog- und Digitalports seit Systemstart an, d.h. dieser Wert wird niemals kleiner.

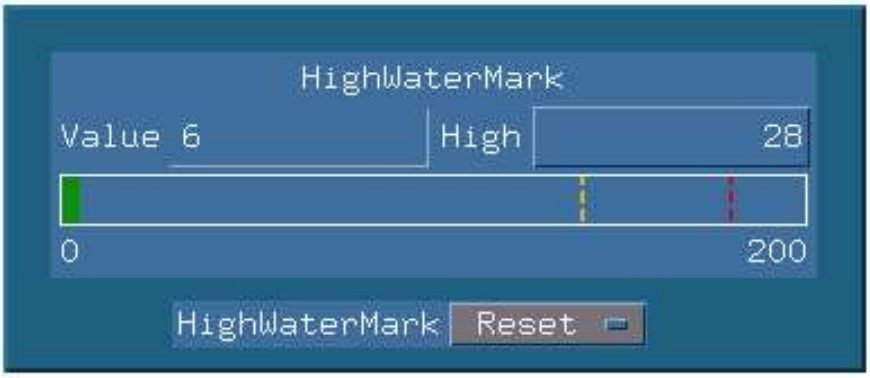

Abbildung 14: Ausschnitt aus dem Call Utilization View

Da diese Variable schreibbar ist, wurde ein *Reset*-Button implementiert, der es ermöglicht, diesen Wert im Ascend-Router auf den aktuellen Wert zurückzusetzen. Dabei hält der *High*-Button wie in Bild 14 dargestellt den vorherigen Wert gespeichert, und läßt sich durch Mausklick dem Wert *Value* anpassen. Neben dieser Grupe befindet sich der `Pie Chart` mit dem Titel *Total Calls*, der das Verhältnis aller ein- und ausgegangenen Rufe, sowohl digital (ISDN) als auch analog (Modem) darstellt. Der *Clear*-Button ermöglicht, die Werte zurückzusetzen, allerdings nur in dem aktuell geöffneten View, nicht im Ascend-Router selbst. Im unteren Bereich befindet sich ein `Multi-Attribute Graph` mit dem Titel *Current Utilization*. Hier werden die aktuellen Daten der ein- und ausgehenden Rufe dargestellt. Die Unterteilung erfolgt auch wieder in digitale und analoge Rufe. Das Element erlaubt das Umschalten zwischen logarithmischer und linearer Darstellung, ebenso wie das Springen zu bestimmten Zeitpunkten über *Graph Properties*. Auf die Darstellung der aktiven Frame Relay Calls wurde hier verzichtet.
Ein Problem, das mit dem Multi-Attribute Graph und dem Tachometer auftritt, ist, daß beide nicht dynamisch skalieren. Bei der Definition dieser Elemente müssen Maximalwerte angegeben werden, in unserem Fall wurde als Maximum 200 gewählt. Dieser Wert soll einen Kompromiß darstellen, um diesen View für die verschiedensten Ascend-Produkte nutzbar zu machen. So hat

ein MAX4000 hier einen Maximalwert von 120 gleichzeitigen Rufen, ein Ascend TNT hingegen schon 672, in Zukunft sind auch hier größere Zahlen aufgrund höherer Portdichten zu erwarten. Dieses Problem betrifft auch andere Elemente des GIB Editors.

EventApp

Durch das EventApp Icon läßt sich der *Ascend Event View* öffnen. In der in in Abbildung 21 ist eine Tabelle zu sehen, in der beim MAX4000 die ca. letzten 1500 Rufe aufgelistet sind. Somit kann dieser View zur Analyse von fehlerhaften Einwahlversuchen dienen.
In der ersten Spalte ist wieder die *starting time* mit den gleichen Einschränkungen wie im *Ascend Call View* zu sehen, dann der Name des Nutzers *(User)*, falls dieser übermittelt werden konnte. Ist dies nicht der Fall und es konnte die Nummer des Anrufenden festgestellt werden, ist diese unter *Calling PID* zu sehen. Leider ist diese Nummer nicht mehr verfügbar, wenn der Nutzername übermittelt wurde. Dies dient zur Einsparung von Speicher auf dem Router. Als nächstes sind der *Service*, *IP-Adresse* und Subnetzmaske *Mask* des Nutzers zu sehen. Im folgenden (auf der Abbildung nicht zu erkennen) erscheinen noch die Spalten *Connect* und *Disconnect*, diese geben den Status des Verbindungsaufbaus und Grund des Verbindungsabbaus wieder.
Durch Mausklick auf die entsprechende Zeile wird eine weitere Ansicht, der *Ascend Event Detail View* wie in Bild 22 zu sehen, geöffnet. Hier lassen sich alle Elemente aus der Tabelle und zusätzlich noch weitere Informationen wiederfinden. So ist unter *DataRate* die Verbindungsgeschwindigkeit abzulesen, hinter *OutOctets* und *InOctets* werden die übertragenen Bytes angezeigt. *CallerPartyID* gibt die Nummer an, die angerufen wurde an. *EventType* gibt den momentanen Status des angezeigten Events an. Auch diese weiteren Informationen erscheinen nie vollständig; sie sind immer abhängig vom Status des Events selbst. Daten zur Slotbelegung wurden wegen der besseren Lesbarkeit nicht dargestellt.
Ein Problem beim Zugriff auf den *Ascend Event Detail View* entsteht dadurch, daß Spectrum Probleme mit Tabellen mit sehr vielen Zeilen hat. Spectrum kann dann nicht mehr auf die Daten zugreifen, entsprechend werden alle

Felder rot, der View gibt also keine Informationen mehr aus und ist somit unbrauchbar. In diesem Zusammenhang noch die Anmerkung, daß der Zugriff auf den *Ascend Event View* im Durchschnitt etwas mehr als eine Minute dauert. Dies ist aber nicht Spectrum anzulasten, sondern der Antwortzeit des SNMP-Agenten auf dem Ascend-Router und der besonders großen Länge der Tabelle.

LModemApp

Der durch LModemApp aufrufbare LanModem View, wie in Abbildung 23 zu sehen, ist schnell erklärt. Ein Multi-Attribute Graph zeigt dem Nutzer die im Moment verfügbaren (avail) und belegten (busy) Modems an. Auch hier wurde wieder der Maximalwert von 200 festgelegt. Weitere Informationen wurden aus Gründen der Übersichtlichkeit nicht in diesen View aufgenommen.

RadiusApp

Es gibt zwei Arten von RADIUS-Servern, Accounting- und Authorisierungs-Server. Die Ascend-Radius-MIB ermöglicht es, auf Statistiken von beiden zurückzugreifen. Über den Menüpunkt *Accounting* wird der View *Ascend Accounting Stats* geöffnet. In der Tabelle in Abbildung 24 wird für jeden der konfigurierten Accounting-Server eine Zeile mit den statistischen Daten dargestellt. In der ersten Spalte ist die IP-Adresse des Hosts sichtbar, in der zweiten dessen Status. Da diese Server redundant aufgesetzt sind und der Ascend-Router im Fehlerfalle zum nächsten Server schaltet, hat entsprechend immer nur einer das Server-Flag. In unserem Fall ist RADIUS-Accounting nicht aktiviert und alle Server haben das *Standby*-Flag. Entsprechend sind die Anzahl der Requests (Rqst), der Requests, die einen Time Out erhalten haben (Rqst Timeo), erhaltene Antworten (Rsp Rcvd) und unerwartet erhaltene Antworten (Unexp Rsp). Durch Klicken auf die entsprechende Zeile gelangt man zu einem weiteren View *Ascend Accounting Detail View*, wie in Bild 25 zu sehen.
Hier ist nochmal die IP-Adresse des gewählten Servers zu sehen. Daneben befindet sich ein Option Button, der es ermöglicht, einen Accounting-Server vom Standby-Status in den Server-Status zu bringen.
Des weiteren sind in zwei `Pie Charts` Statistiken über die Verhältnisse von An-

fragen und Zeitüberschreitungen (Requests), und Antworten, die der Ascend Router wie gewünscht und unerwartet erhalten hat (Responses) zu sehen.
Ähnlich wie der Accounting View verhält sich der *Ascend Radius Authentication View*, in Bild 26 zu erkennen. Hier werden jedoch genauere Auskünfte über die Kommunikation gegeben. Neben der IP-Adresse der Server sind *Login Rqst*, *Other Rqst Rqst Timeo* und *Other Timeo* zu sehen. Other bedeutet, daß die Anfragen keine Nutzeranfragen, sondern z.B. solche nach globalen Parametern wie zum Beispiel Routing waren. Entsprechend unterteilen sich die Antworten auch in *Rsp Rcvd* und *Other Rcvd*. *Unexp Rsp* sind unerwartet erhaltene Pakete vom RADIUS-Server. Hinter *Bad Rsp* verbergen sich Pakete mit ungültiger Verschlüsselung. *Ack Rsp* schließlich sind alle erfolgreich beantworteten Anfragen an den RADIUS-Authorisierungs-Server.
Auch hier gelangt man durch Mausklick auf die entsprechende Zeile zum *Ascend Radius Authentication Detail View*, wo wieder die IP-Adresse des aktuellen Servers und ein Button zum Ändern des Server-Status wiederzufinden ist. In den drei Pie Charts sind die *Requests*, zum einen unterteilt nach Login und den übrigen Anfragen, zum anderen nach Antworten und Zeitüberschreitungen dargestellt. Die Antworten teilen sich auf *Ack Bad* und *OtherRsp*.
In dieser Gruppe wurden alle Informationen der Ascend-Radius-MIB in den Tabellen dargestellt, die Graphen sollen dem Administrator einen schnellen Rückschluß auf Probleme ermöglichen.

SessionApp

Hinter dem SessionApp Icon verbirgt sich der Session View. In Abbildung 28 sind in der Tabelle *Active Session* die eingeloggten Nutzer (User), deren Verbindungsprotokoll (Service), IP-Adresse, Subnetzmaske und auch wieder die Referenznummer des Anrufs zu sehen. Durch Mausklick auf die entsprechende Zeile gelangt man zum Session Detail View, wie in Bild 29 sichtbar. Neben den eben genannten Daten erhält man zusätzlich die Möglichkeit, über den Option Button *User remove* den aktuell aktiven Nutzer via Spectrum auszuloggen. Allerdings wird auch die hier verwendete MIB-Variable von der aktuellen Firmewareversion nicht unterstützt, das Ausloggen der Nutzer ist also im Moment nicht möglich. Diese Ascend-Session-MIB ermöglicht noch das Managen von

Multilink-Verbindungen (MP+), welches aber aufgrund der mangelnden Testbarkeit nicht implementiert wurde.

5.4. Alarme

Alarme haben in Spectrum eine besondere Stellung. So können für jeden Model Type Alarme vom Anwender konfiguriert werden. Dazu sind keine Teile der Level I oder Level II Entwicklungsumgebungen notwendig. Alarme werden ausglöst durch Events. Events können ausgelöst werden durch SNMP-Traps. Die Konfiguration ist hierbei über das Spectroserver Configuration Panel möglich. In unserem Fall ist nur der `radiusServerChange` Trap aus der Ascend-Trap-MIB von Interesse. Dieser wird vom Agent des Access-Routers dann versendet, wenn der RADIUS-Server z.B. aufgrund einer Zeitüberschreitung wechselt. In diesem Fall wird ein Alarm im Alarmmanager vom Spectrographen mit sichtbar.

6. Test

6.1. Discovery

Der erster Schritt im Test auf erfolgreiche Implementierung ist die Prüfung auf erfolgreiches Discovery im Spectrographen. Dies gilt sowohl für das Ascend-Router Modul, als auch für die Major- und Minor Application Icons im Application View. Der Test der Application Icons ist unabhängig vom Ascend Router Modul, funktioniert also auch, wenn der Ascend Router mittels GnS-NMP im Spectrographen modelliert wurde. Das Entdecken des Ascend Router Moduls verlief ohne Probleme, ebenso wie die des Major Application Icons `AscendApp`. Im ersten Lauf wurden allerdings keine Minor Application Icons sichtbar, was zu folgenden Überprüfungen führte:
Als erstes wurde das Vorhandensein des Discoveryattributes in der MIB mittels der Cabletron MIB-Tools überprüft. Mit deren Hilfe kann man eine Anfrage auf eine bestimmte MIB-Variable starten, man erhält dann den aktuellen Wert dieser Variable vom Gerät zurück.
Hier wurde bestätigt, daß die eingetragenen OIDs vorhanden sind. Daraufhin wurde geprüft, ob die `PROVIDES`-Relation zum Major Application Icon *AscendApp* korrekt gesetzt ist. Dies war auch der Fall. Ein genaueres Studium der Dokumentation führte dann zu dem Schluß, daß der MIB-Compiler des MTE die `IMPORTS`-Anweisung aus den MIBs ignoriert und somit nicht die vollständigen OIDs in Attribute-IDs abgebildet hatte. Deshalb mußten alle MIBs, die Ascend-spezifische `IMPORTS`-Anweisungen beinhalten, editiert werden. Hierzu wurden die `IMPORTS`-Anweisungen auskommentiert und die entsprechenden `OBJECT IDENTIFIER` in die MIBs hinzugefügt. In Bild 15 ist dies am Beispiel der Ascend-Radius-MIB zu sehen. Es bleibt zu hoffen, daß in zukünftigen Versionen von Spectrum dieses Manko behoben wird, zumal der MIB-Compiler des MTE keine Warnungen ausgegeben hat und die Spectrum MIB-Tools dieses Problem nicht haben.
Nachdem diese Modifikationen durchgeführt und die MIBs neu über den MTE importiert waren, wurden auch die Minor-Application Icons nach dem Discovery korrekt modelliert.

```
    IMPORTS
--      radiusGroup
--          FROM ASCEND-MIB.

    ascend OBJECT IDENTIFIER ::= { enterprises 529 }
    radiusGroup OBJECT IDENTIFIER ::= { ascend 13 }
```

Abbildung 15: Änderungen an den MIBs

6.2. Information Views

Nachdem die Application Icons korrekt entdeckt wurden, war es möglich, Information Views zu implementieren und zu testen. Der Test zielt hierbei auf Funktionalität und Korrektheit. Da bei der Entwicklung neuer Views mittels des GIB Editors gleich nach dem Einfügen eines neuen Elements der Inhalt des Attributs – welches bei externen Attributen auch eine MIB-Variable sein kann – dargestellt wird, ist im Gegensatz zu herkömmlichen Entwicklungsumgebungen kein separater Test auf Funktionalität nötig. Um die dargestellten Variablen auf Korrektheit zu überprüfen, ist es jedoch möglich, einen Vergleich mit den von den MIB-Tools gelieferten durchzuführen.
So zeigte es sich während des Entwicklungszyklus, daß eine Vielzahl der in den Ascend-MIBs vorhandenen Variablen nicht vom Gerät geliefert werden. Dies hat zur Folge, daß die Felder in den GIB-Views rot markiert werden. Als Ursache für dieses Problem stellte sich die Firmware des MAX4000 heraus. Die installierte Firmware der Version 5.0Ap38 unterstützt aufgrund unvollständiger SNMP-Implementierung eine große Anzahl der MIB-Variablen nicht. Eine Firmwareversion mit korrekter SNMP-Implementierung konnte aufgrund mangelnder K56Flex-Modem-Unterstützung nicht installiert werden. Die Versuche, die Beta-Versionen des Releases 6.0 zu etablieren, führten ebenfalls zu Problemen mit der Modem-Nutzung. Da leider kein Ascend-Router als Testgerät zur Verfügung stand, war ein Test der von der Firmware Version 5.0Ap38 nicht unterstützten Variablen nicht möglich, aber im Hinblick auf ein baldiges Erscheinen einer funktionsfähigen Version 6.0 wurden auch diese in Spectrum modelliert, um ein möglichst weites Feld von Daten repräsentieren zu können.

7. Zusammenfassung

In dieser Arbeit wurde versucht, Konzepte zum Management von Access-Routern zu entwickeln und eine Implementierung für einen Ascend-Router mit Cabletron Spectrum vorzustellen. Hierzu wurden im theoretischen Teil zunächst die Grundlagen des Netzwerkmanagements diskutiert.
In der praktischen Implementierung wurde ein funktionsfähiges Modul entwickelt, mit dessen Hilfe es möglich ist, einen weiten Einblick in die Aktivitäten des vorhandenen MAX4000 zu bekommen. Durch die Generalität der Ascend-MIBs ist es mit diesem Modul möglich, die komplette Produktfamilie von Ascend-Access-Routern zu managen. So erhält man jetzt einen Überblick über die eingeloggten Nutzer, die Auslastung der Ports, und es wird ermöglicht, Eingriffe im aktiven Betrieb vorzunehmen. Während der Implementierung traten einige Schwierigkeiten auf: Zum einen unterstützt die gegenwärtig aktuelle Firmware nicht die komplette MIB, was dazu führte, daß eingebaute Funktionen nicht getestet werden konnten. Aber in Hinblick auf ein baldiges Erscheinen einer Firmware mit vollem Funktionsumfang wurden auch diese Werte graphisch dargestellt. Eine kleine Einschränkung stellten auch die verfügbaren Elemente im Spectrographen dar. So wäre es wünschenswert gewesen, Elemente mit dynamischer Skalierung zur Verfügung zu haben, um Werte wie die Anzahl der belegten Zugänge besser visualisieren zu können.

8. Ausblick

Gerade im Hinblick auf die angekündigten Erweiterungen im SNMP-Bereich mit Erscheinen der Firmware Version 6.0 für Ascend-Router und der zu erwartenden Funktionalität der bisher fehlerhaften Implementierung sollte es als reizvoll erscheinen, das bestehende Modul um weitere Funktionen zu erweitern. Auch im Bereich der Kanalbündelung bieten die Ascend-MIBs noch weite Möglichkeiten des Managements, welche aufgrund fehlender Testmöglichkeiten nicht mit in das Modul eingebaut wurden. Auch ein Chassis oder Slotmanagement ist als Option noch zu entwickeln, hier müssen dann allerdings Geräteunterschiede zwischen den verschiedenen Ascend-Access-Routern gemacht werden.
Auch wird mit Sicherheit die Anwendung des Moduls weitere Wünsche seitens der Administratoren hervorbringen, die in zukünftige Erweiterungen mit einfließen sollten.
Desweiteren verkündet Cabletron das Erscheinen von Spectrum in der Version 5.0 für das zweite Quartal 1998 in Deutschland. Es bleibt zu hoffen, daß zumindest einige Erweiterungen im Bereich des Level I-Toolkit vorgenommen werden, gerade für das Design der Information Views würden einige neue Elemente mit dynamischer Skalierung dem Entwickler neue Freiheiten und Möglichkeiten, dem Spectrograph-Anwender eine verbesserte Übersicht geben.
Ein interessantes Feature am Rande: Gerade bei Providern sind nicht gerade wenig Access-Router im Einsatz. Zum Teil sind hier Hunderte von Geräten an den verschiedensten Standorten zu managen. Hier setzt Cabletron Enterprise Configuration Manager (ECM) an. ECM ermöglicht es, zentral Konfigurationen für alle Geräte, zum Beispiel eben für alle Ascend-Router zu halten und zu bestimmten Uhrzeiten für alle oder einen Teil der Geräte zu aktivieren.

A. Abbildungen

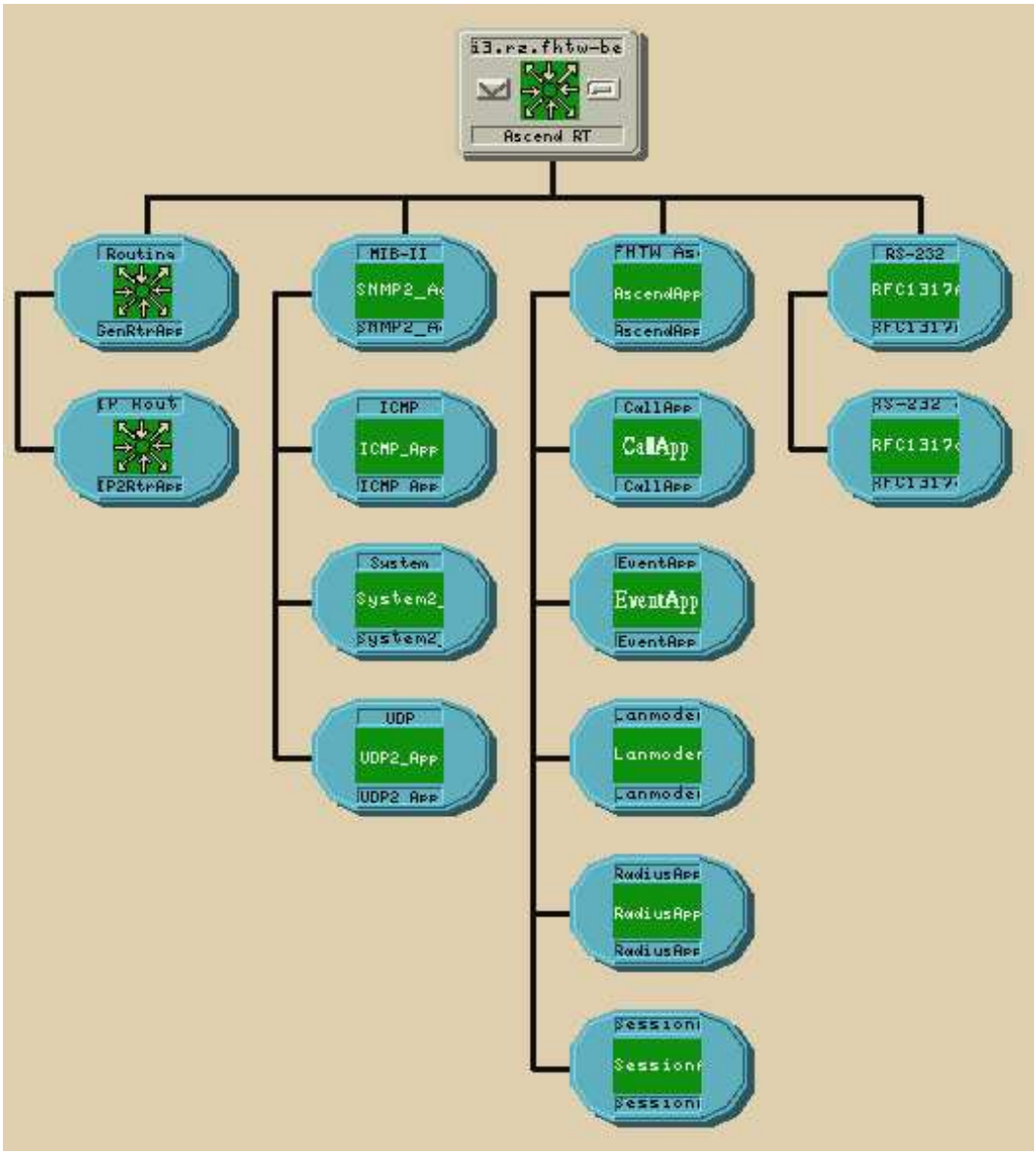

Abbildung 16: Application View

Abbildung 17: TFTP View

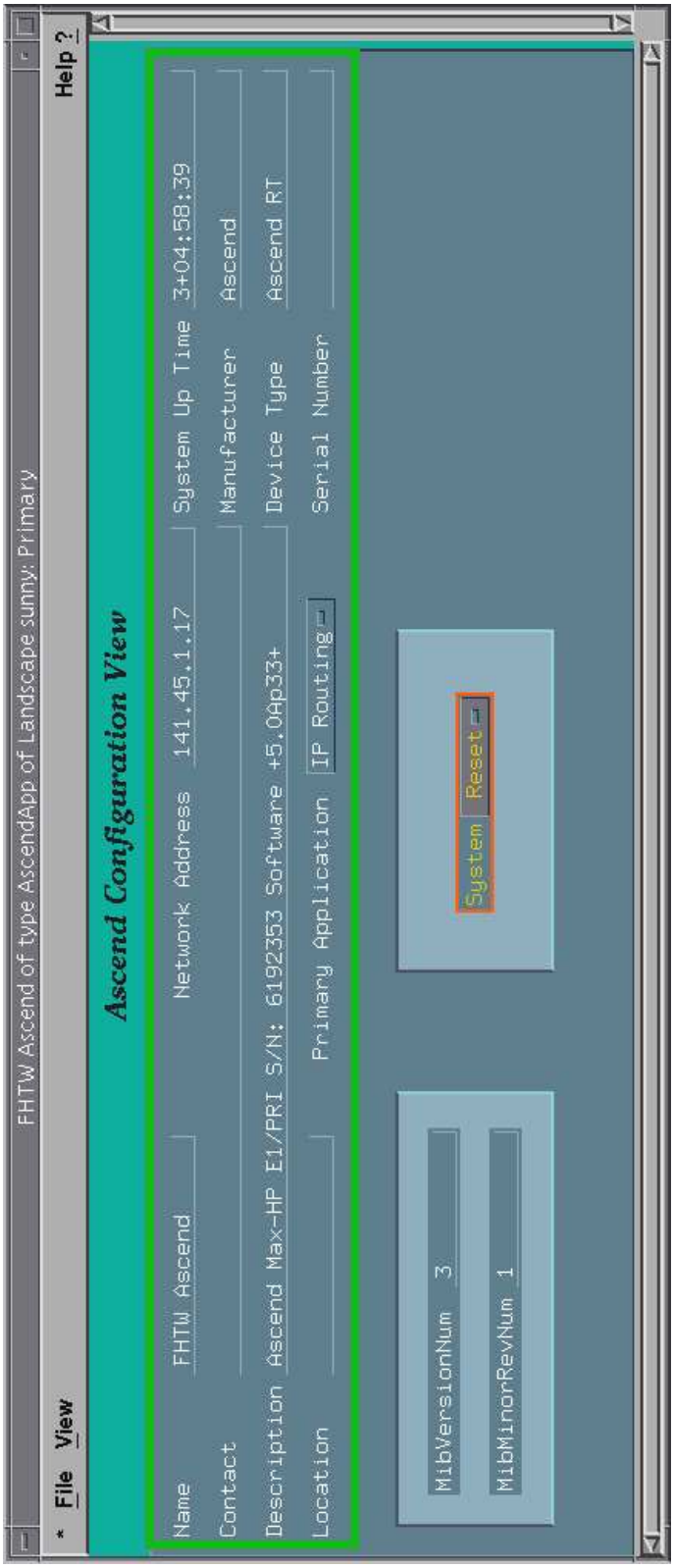

Abbildung 18: Configuration View

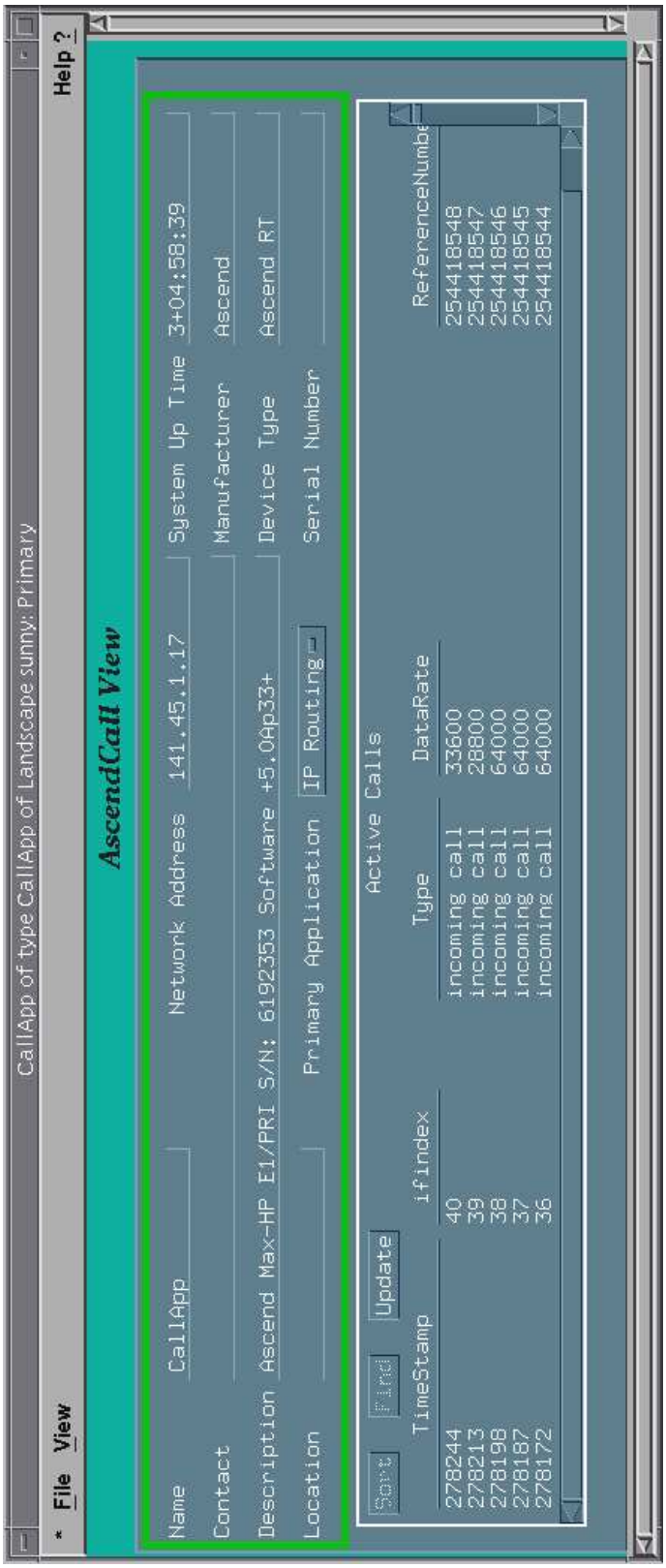

Abbildung 19: Call View

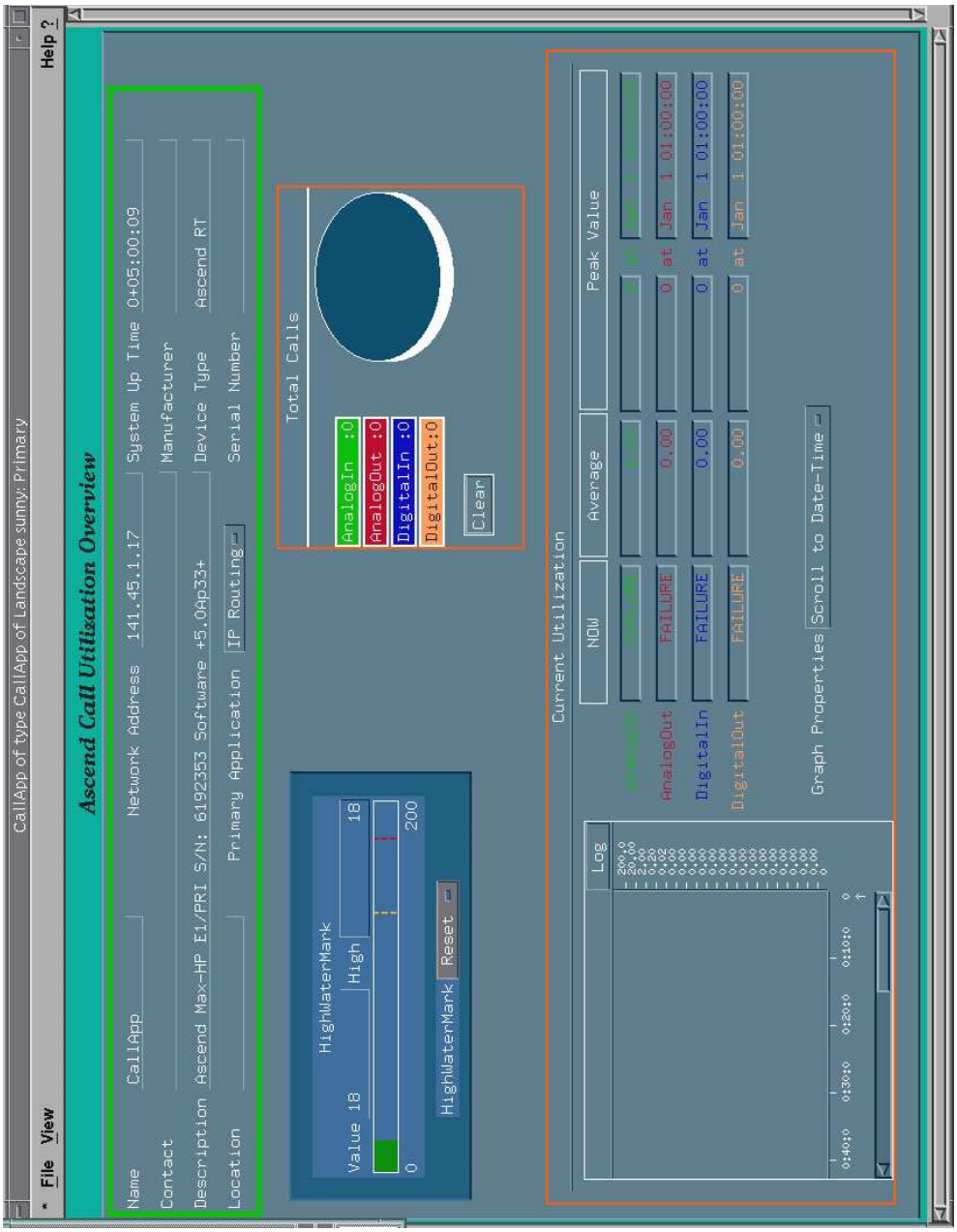

Abbildung 20: Call Utilization View

Abbildung 21: Event View

Abbildung 22: Event Detail View

Abbildung 23: Lanmodem View

Abbildung 24: RADIUS Accounting View

Abbildung 25: RADIUS Accounting Detail View

Abbildung 26: RADIUS Authentication View

Abbildung 27: RADIUS Authentication Detail View

Abbildung 28: Session View

Abbildung 29: Session Detail View

B. Entwicklungsumgebung

Zur Entwicklung des Managementmoduls wurde folgende Hardware eingesetzt:

- Entwicklungsrechner
  Sun Sparc 5
  1 CPU 110Mhz, 128 MB RAM, 2GB HD
  Betriebssystem Solaris

- Access-Router Ascend MAX4000
  2*12-Port K56Flex-Modemkarten, maximal 60 ISDN Zugänge
  Firmeware 5.0Ap38

Als Software kamen folgende Produkte zum Einsatz:

- Betriebssystem
  Sun Solaris 2.5.1

- Netzwerkmanagementsystem
  Cabletron Spectrum Version 4.03 MS1

- Entwicklungsumgebung
  Spectrum Level I Toolkit bestehend aus Model Type Editor, Generic Information Block Editor und Icon Information Block Editor

- ASCII-Editor
  Xemacs 19.21

- LaTeX als Satzsystem für diese Arbeit

C. Literatur

Literatur

[1] *MAX4000 Guide*
Ascend Communications, 1996

[2] *Corporate Remote Access Guide*
Ascend Communications, 1997

[3] Ascend Communications
Ascend Enterprise MIB
ftp://ftp.ascend.com/pub/Doc/SNMP/MIBS/

[4] Lundy Lewis
Managing Computer Networks – A Case-Based Reasoning Approach
Archtech House, 1995

[5] Marshall T. Rose
The Simple Book – An Introduction To Internet Management
Prentice Hall, 1994

[6] Marshall T. Rose, Keith McCloghrie
RFC-1155; Structure and Identification of Management Information for TCP/IP-based internets
http://ds.internic.net/rfc/rfc1155.txt

[7] Marshall T. Rose, Keith McCloghrie
RFC-1156; Management Information Base for Network Managment of TCP/IP-based internets
http://ds.internic.net/rfc/rfc1156.txt

[8] Jeffrey D. Case
RFC-1157; A Simple Network Management Protocol
http://ds.internic.net/rfc/rfc1157.txt

[9] Carl Rigney, Livingston Enterprises
RFC-2138; Remote Authentication Dial In User Service (RADIUS)
http://ds.internic.net/rfc/rfc2138.txt

[10] Carl Rigney, Livingston Enterprises
RFC-2139; RADIUS Accounting
http://ds.internic.net/rfc/rfc2139.txt

[11] *Spectrum Administrator's Reference*
Cabletron Systems, 1997

[12] *Spectrum GIB Editor Guide*
Cabletron Systems, 1997

[13] *Spectrum Level II Toolkit Overview*
Cabletron Systems, 1997

[14] *Spectrum Modeling with th GnSNMPDev Toolkit*
Cabletron Systems, 1996

[15] *Spectrum Operator's Reference*
Cabletron Systems, 1997

[16] The Tolly Group, 1997
Remote Access Concentrator Performance and Scalability, Paper No. 7309
http://www.tolly.com/

D. Selbständigkeitserklärung

Ich erkläre, daß ich die vorliegende Diplomarbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Berlin, 29.01.1998 Thorleif Wiik