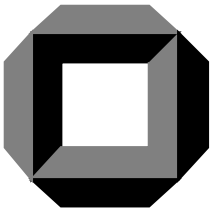


Prof. Dr. Dr. h.c. G. Krüger
Institut für Telematik
Universität Karlsruhe (TH)

Zirkel 2
76128 Karlsruhe
Tel.: 0721/608-3835
e-mail: krueger@telematik.informatik.uni-karlsruhe.de

Netzwerk–Management
und
Hochgeschwindigkeits–Kommunikation
Teil XI

Seminar WS 1994/95

Markus Hofmann

Günter Schäfer

Claudia Schmidt

Jochen Seitz

Institut für Telematik

Universität Karlsruhe

Interner Bericht 18/95

Zusammenfassung

Der vorliegende Interne Bericht enthält die Beiträge zum Seminar "Netzwerk-Management und Hochgeschwindigkeits-Kommunikation", das im Wintersemester 1994/95 zum elften Mal abgehalten wurde.

Im Mittelpunkt stehen zuerst aktuelle Entwicklungen im Internet, die zukünftige Protokollarchitekturen sowie die Möglichkeit zur Gruppenkommunikation und zur realzeitfähigen Datenkommunikation umfassen. Dabei spielt auch das Problem der Dienstgüte, wie sie beispielsweise von Multi-Media-Anwendungen gefordert wird, eine große Rolle.

Der zweite Block befaßt sich mit dem Problem der Sicherheitsvorkehrungen in Kommunikations– und Rechnernetzen. Auch hier werden aktuelle Forschungsergebnisse vorgestellt.

Weiterhin wird mit der Common Object Request Broker Architecture eine zukunftsweisende Architektur beschrieben, die umfassendes System– und Netzwerkmanagement ermöglicht.

Den Abschluß bildet ein Beitrag zum Management breitbandiger Weitverkehrsnetze, wodurch der Kreis vom Netzwerk-Management hin zur Hochgeschwindigkeits-Kommunikation wieder geschlossen wird.

Abstract

This Technical Report includes student papers produced within small lessons called seminar of "Network Management and High Speed Communications". Concerning the eleventh time of this seminar students showed an increased interest – again many thanks to them in this term – which proved, that issues concerning topics of network management and high speed communications are of broad and accepted interest.

This time the papers begin with actual problems and solutions for the internet. First, the IP Next Generation approaches are discussed, second, the MBone architecture is introduced, and third, the Real–Time Transport Protocol RTP is explained. Concerning multi-media applications the aspects of Quality of Service (QoS) must be considered as well, which is the topic of the forth paper.

The second part of the report deals with security problems. Several security architectures are illustrated and international standards are discussed. Furthermore, the Common Object Request Broker Architecture, an object oriented architecture for systems and network management, is the topic of the seventh paper.

Finally the topic "management of broadband wide area networks" brings the research areas network management and high speed communications together.

Vorwort

Das Seminar "Netzwerk-Management und Hochgeschwindigkeits-Kommunikation" erfreute sich in den letzten Jahren immer größerer Beliebtheit. Gerade heutzutage sind Stichworte wie "Datenautobahn", "Multi-Media-Kommunikation" oder "Breitband-ISDN" in aller Munde. Daher sind die Forschungsgebiete in diesen Bereichen auch von allgemeinem Interesse, so daß sie eine derartige Vielzahl an innovativen Arbeiten aufweisen können, deren Behandlung in anderen Lehrveranstaltungen so detailliert nicht möglich ist.
Jetzt liegt auch der nunmehr elfte Seminarband als interner Bericht vor. Durch das engagierte Mitarbeiten der beteiligten Studenten konnte so zumindest ein Ausschnitt aus dem komplexen und umfassenden Themengebiet klar und übersichtlich präsentiert werden. Für den Fleiß und das Engagement der Seminaristen sei daher an dieser Stelle recht herzlich gedankt
Die ausgesprochen gute Resonanz bei den Studenten hat uns veranlaßt, auch im Sommersemester 1995 ein derartiges Seminar — natürlich mit geändertem Inhalt — durchzuführen, so daß bald ein weiterer interner Bericht mit neuen Forschungsergebnissen aus aktuellen Tagungsbeiträgen erscheinen wird. Doch vorerst sollen im vorliegenden Band folgende Themengebiete vorgestellt werden:

Zukünftige Protokollarchitekturen für das Internet

Mit zunehmender Bedeutung der weltweiten Rechnerkommunikation ("Datenhighway") wächst die Anzahl der Teilnehmer und der zu verbindenden Netzsegmente im Internet. Der Einsatz etablierter Protokollarchitekturen, wie beispielsweise TCP/IP, führt in diesem Zusammenhang zu Problemen im Bereich der Adressierung und der Vermittlung von Datenpaketen.
Der vorliegende Beitrag stellt Arbeiten verschiedener Internet-Gruppen vor, die den Entwurf einer geeigneten Architektur für das zukünftige Internet-Protokoll (IP:Next Generation) zum Ziele haben. Im Mittelpunkt steht hierbei das TUBA-Projekt, welches den Einsatz des OSI-Vermittlungsprotokolls im Internet vorsieht. Ebenso wird der Ansatz der CATNIP-Gruppe untersucht, die eine Architektur zur Integration der Internet-, OSI- und Novell-Protokolle entwickelt.

MBone — Gruppenkommunikation im Internet

Moderne Multimedia-Anwendungen erfordern im Zusammenhang mit ihrem Einsatz in Arbeitsgruppen von dem zugrundeliegenden Kommunikationssystem eine leistungsfähige Unterstützung der Gruppenkommunikation. Das Multicast-Backbone (MBone) ist ein Versuch, die Protokollfamilie im Internet um die Fähigkeit der Gruppenkommunikation zu erweitern.
Im Rahmen des Beitrags werden sowohl die Struktur und die Verwaltung des Multicast-Backbone als auch die Details der eingesetzten Protokolltechniken vorgestellt. Ein Überblick über verfügbare Anwendungen und die derzeit aktuelle Ausbauphase des MBone runden die Arbeit ab.

RTP — Übertragung von Audio/Video im Internet

Aktuelle Protokolle der Internetwelt bieten keine Unterstützung für Multimedia-Anwendungen, wie z.B. die Garantie von Qualitätsparametern (Durchsatz, Verzögerung, Jitter). Aus diesem Grund formierte sich die "Audio/Video Transport Group (AVT)" der IETF (Internet Engineering Task Force) mit der Zielsetzung, experimentelle Protokolle zu entwickeln, die eine Übertragung von Multimedia-Daten in der Internetwelt erlauben. Ein Ergebnis dieser Aktivitäten ist das Real-Time Transport Protocol, das in diesem Beitrag vorgestellt wird.

Der QoS-Broker — Handel mit Dienstgüte?

Multimedia-Anwendungen sind durch die gleichzeitige Übertragung mehrerer Datenströme mit unterschiedlichen Anforderungen an die Dienstqualität gekennzeichnet. Dieser Seminarbeitrag stellt eine an der University of Pennsylvania entwickelte Komponente — den sogenannten QoS-Broker — vor. Der QoS-Broker ist als Vermittlungsinstanz zwischen den an der Kommunikation beteiligten Instanzen für die Aushandlung der Dienstqualität zuständig.

Vergleich zweier Sicherheitsarchitekturen für offene verteilte Systeme

Um einen möglichst effektiven Schutz in einem verteilten System zu erreichen, sollten einzelne Sicherheitsdienste im Rahmen einer Sicherheitsarchitektur aufeinander abgestimmt werden. Im vorliegenden Beitrag werden zwei mögliche Architekturen vorgestellt und auf ihre spezifischen Vor– und Nachteile hin untersucht.

Sicherheitsmanagement in offenen Systemen

Dieser Seminarbeitrag führt in die grundlegenden Problemstellungen des Sicherheitsmanagements, wie Gefahren in offenen Systemen, Sicherheitsdienste und das OSI-Sicherheitsmanagement ein. Desweiteren wird das europäische Projekt "Samson" vorgestellt, welches sich um eine Integration bestehender Sicherheitsdienste bemüht.

Object Management

Die Gebiete des Netzwerk– und des Systemmanagements nähern sich immer mehr an, was vor allem durch den Trend begründet wird, daß heutige Systeme in der Regel netzwerkfähig ausgeliefert werden. Somit sollte auch die Verwaltung derartiger Systeme beide Aspekte integriert betrachten. Andererseits sind die Abhängigkeiten zwischen einem funktionierenden System und einem intakten Netzanschluß derartig umfangreich, daß die getrennte Verwaltung von System und Netzwerkanbindung eigentlich nicht mehr möglich ist.
Ein Ansatz dazu wird im "Common Object Request Broker" der Object Management Group realisiert, der von einem objektorientierten Modell ausgeht. Der Beitrag hierzu stellt die zugrundeliegende Architektur, kurz "CORBA" vor. Anhand von Beispielen wird die Funktionsweise des Request Brokers beschrieben und abschließend das "Distributed Management Environment" der Open Software Foundation vorgestellt, welches als erste Architektur einen Object Request Broker enthält.

Breitband-Netzwerkmanagement

Der abschließende Beitrag zum vorliegenden Seminar befaßt sich mit dem Management von Hochgeschwindigkeitsnetzen. Hierbei zeigt sich, daß konventionelle Ansätze, welche für die Verwaltung bisheriger, also langsamerer Weitverkehrsnetze oder für das Management lokaler Netze ausgelegt sind, auf dem Gebiet der breitbandigen Hochgeschwindigkeitsnetze bald an ihre Grenzen stoßen. Hierfür wurde das sogenannte "Distributed Broadband Management" (DBM) definiert, welches die Defizite bisheriger Ansätze minimieren soll.
Der DBM-Ansatz, der weg vom zentralen Überwachungssystem hin zum Management einzelner Schichten geht, wird in seinen Eigenschaften vorgestellt und anhand von Beispielen erläutert. Hierzu wird vor allem das sich noch in der Entwicklung befindende Managementsystem der "Federal Aviation Administration" herangezogen. Der Beitrag bildet somit sozusagen den Brückenschlag zwischen Netzwerk-Management und Hochgeschwindigkeits-Kommunikation.

Inhaltsverzeichnis

Zukünftige Protokollarchitekturen für das Internet 1
Multicast Backbone — Gruppenkommunikation im Internet.....17
RTP — Übertragung von Audio/Video im Internet 31
Der QoS-Broker — Handel mit Dienstgüte? 49
Vergleich zweier Sicherheitsarchitekturen für offene, verteilte Systeme...63
Sicherheitsmanagement in offenen Systemen.....79
Object-Management.....97
Breitband-Netzwerkmanagement 109

ZUKÜNFTIGE PROTOKOLL-ARCHITEKTUREN FÜR DAS INTERNET

René Michel

Das Internet hat sich in den letzten Jahren vom reinen Forschungsnetzwerk zur globalen Computer-Infrastruktur der Wissenschaft und Ausbildung entwickelt. Schon heute sind über 2,3 Millionen Rechner am Internet angeschlossen, und durch die bevorstehende öffentliche und private Nutzung des Internet's ist weiterhin eine hohe Wachstumsrate zu erwarten. Etablierte Protokollarchitekturen wie TCP/IP werden deshalb nur noch begrenzte Zeit in der Lage sein, die Adressierung und die Vermittlung von Datenpaketen im Internet durchzuführen. Im Folgenden werden zwei Projekte vorgestellt, deren Ziel es ist, das aktuelle Internet Protokoll durch eine neue Protokollarchitektur zu ersetzen. Beide Architekturen sollten, auch auf längere Sicht, in der Lage sein, den Anforderungen, die das stetige Wachstum des Internet's aufwirft, zu genügen.

1 IP : Next Generation (IPng) - Motivation

Die von IP Version 4 (IPv4) verwendeten Adressen haben eine Länge von 32 Bit und bestehen aus einer Netzwerk-Identifikationsnummer (netid) und einer Host-Identifikationsnummer (hostid), welche innerhalb des adressierten Subnetzes eindeutig ist. Die jeweilige Länge dieser Felder wird durch folgende Adreßklassen festgelegt :

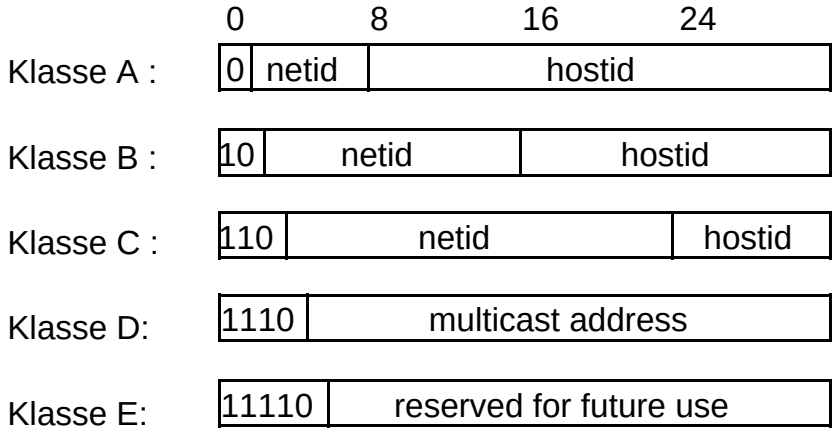


Abb.1 : IP-Adreßklassen

Um effizientes Routing zu ermöglichen, ist eine hierarchische Adressierung nötig. Bei IP ist eine solche durch die Aufspaltung in netid und hostid gegeben. Das heißt die Menge an Information, die zum Routing benötigt wird steigt mit zunehmendem Wachstum des Internet‘s relativ stark an (jährlich etwa um den Faktor 2). Theoretisch lassen sich mit 32 Bit etwa 4 Milliarden Hosts eindeutig adressieren. Allerdings ist dieser Adreßraum, durch die Einteilung in verschiedene Adreßklassen, nicht effektiv nutzbar. Wenn der Adreßraum der Klasse B erschöpft ist, müssen neuen Systemen Adressen der Klasse C zugewiesen werden. Bei der Verwendung von Adressen der Klasse C, müssen jedoch Institutionen mit mehr als 254 Hosts mehrere *netids* zugewiesen werden. Da in den Routing Tabellen jede Subnetzwerkadresse einen Eintrag erfordert, werden diese so stark anwachsen, daß effizientes Routing nicht mehr möglich sein wird. Auch die Verwendung von CIDR (Classless Inter-Domain Routing), bei dem auf die Einteilung in verschiedene Adreßklassen verzichtet wird, würde den Überlauf des Adreßraumes nur bis etwa zum Jahr 2000 hinauszögern. Um also längerfristig den Anforderungen, die das ständig wachsende Internet aufwirft gerecht werden zu können, reichen 32 Bit Adressen nicht aus. Deshalb beschäftigen sich mehrere Internet-Gruppen mit dem Entwurf neuer Internet-Protokolle.

2. TUBA

Ein solcher Entwurf ist TUBA (TCP and UDP with bigger addresses), welcher als Netzwerkprotokoll das OSI-Protokoll CLNP (Connectionless Network Layer Protocol) vorsieht. Mit TUBA sollen folgende Ziele verwirklicht werden :

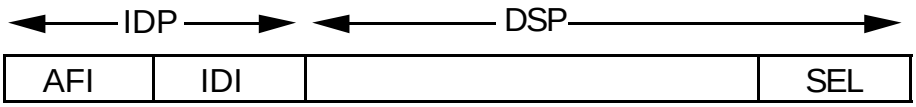
- Adressierbarkeit beliebig großer Internetzwerke

- Effizientes Routing in beliebig großen Internetzwerken
- Unterstützung von mehreren Protokollen
- Minimales Risiko und möglichst geringe Kosten bei der Übergangsphase
- Bereitstellen einer Basis für die Entwicklung weiterer Internet-Dienste

Als Protokoll wurde CLNP gewählt, da es viele Merkmale mit IP gemeinsam hat (IP diente als Vorlage) und außerdem Adressen variabler Länge (NSAPs) verwendet. Ein weiterer Grund ist, daß CLNP schon standartisiert ist und für viele Systeme bereits Implementierungen existieren. Bei CLNP handelt es sich, wie bei IP, um ein unzuverlässiges Datagramm-Protokoll, bei dem mit jedes Paket die volle Quell- bzw. Zieladresse enthalten muß.

2.1 Adressierung

TUBA verwendet als Netzwerkadressen wie schon erwähnt OSI-NSAPs. Allerdings wird die in der OSI Norm vorhandene Längenbeschränkung der NSAPs auf 20 Oktetts aufgehoben. Ein solcher NSAP hat folgendes Format :



- AFI Authority and Format Identifier
- IDI Initial Domain Identifier
- IDP Initial Domain Part
- DSP Domain Specific Part
- SEL Selektor

Abb.2 : NSAP

AFI und IDI bilden zusammen den IDP, welcher die administrative Autorität, die für den jeweiligen Teil des Adreßraumes zuständig ist, festlegt. Das Format des DSP wird nicht global festgelegt, sondern wird nur durch die vom IDP adressierte Autorität bestimmt. Das letze Oktett SEL, wird verwendet um eine von mehreren Transportschicht-Entitäten auszuwählen. Im Gegensatz zu IP, welches Netzwerkinterfaces adressiert, beziehen sich OSI-NSPAs auf Systeme, wodurch zum Beispiel bei Ausfall eines Interfaces dessen Arbeit automatisch durch ein anderes Interface übernommen werden kann. Außerdem vereinfacht dies die Konfiguration der Hosts, da nicht jedes Interface einzeln konfiguriert werden muß.

2.2 Routing

TUBA verwendet die OSI Routing-Protokolle ohne Änderung. Die globale Routing Struktur wird in Routing Domänen unterteilt. Innerhalb einer Routing Domäne, welche ihrerseits wieder in "Areas" unterteilt ist, wird nach genau einer Routing Strategie verfahren, die sogar bei tausenden von Host innerhalb der Domäne effizientes Routing ermöglichen muß.

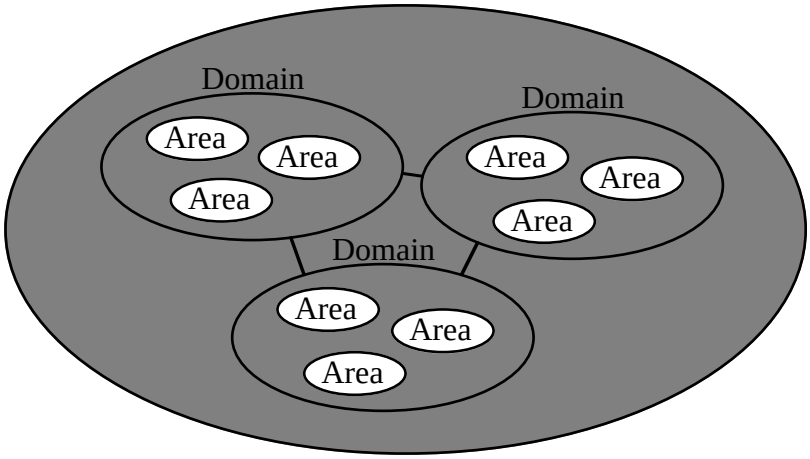


Abb.3 : Globale Routing Struktur

2.2.1 Subnetz Routing

Das Routing zwischen Host und Router wird durch das ES-IS (end system to intermediate system) Protokoll durchgeführt. Dieses stellt folgende Funktionalität bereit:

- Herstellung und Aufrechterhaltung der Erreichbarkeit
  von Hosts und Routern. Dazu werden bei ES-IS periodisch sogenannte "Hallo Pakete" verschickt. Hosts senden periodisch ESH (End system hello) Pakete an alle Router. Damit machen sie den Routern ihre NSAPS und ihre physikalischen Adressen, sowie die Dauer derer Gültigkeit bekannt. Auch die Router senden Hallo-Pakete (ISH : Intermediate system hello) an alle Hosts, die somit die Identität und die physikalische Adresse "ihrer" Router kennen. Um zu verhindern, daß Systeme sich mit hello packets, die für sie nicht von Belang sind, auseinandersetzen müssen, werden die Multicast-Adressen "all hosts" und "all routers" eingesetzt. Falls ein neuer Router hinzukommt, verschickt dieser, via Multicast, sein Hallo Paket an alle Hosts des Subnetzes, welche ihrerseits dann per Unicast antworten. Mit den ES-IS Paketen von Router zu Host kann auch die Gültigkeitsdauer, die der Host für seine ESHs verwenden soll, übertragen werden, so daß ein ganzes Subnetz von seinen Routern aus "getuned" werden kann.

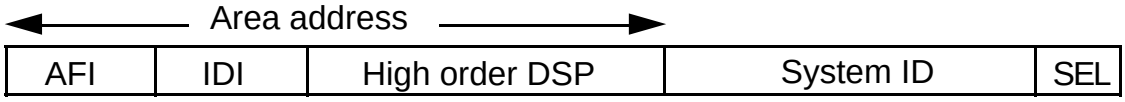
- Füllen der Host Routing Caches
  Jeder Host benutzt einen Cache, in dem er Information über die ihm bekannten NSAPs speichert. Will nun ein Host ein Zielsystem adressieren, so schaut er unter dessen NSAP in seinem Cache nach. Ist dort ein Eintrag vorhanden, so findet er unter diesem die physikalische Adresse des Routers, über welchen er das Paket verschickt. Falls er keinen Eintrag im Cache findet, schickt er sein Paket an irgendeinen Router aus seiner Liste (die er durch empfangene ISHs gebildet hat). Falls dieser Router nicht auf dem optimalen Pfad liegt, schickt er ein "ES-IS redirect packet" zurück, mit welchem er dem Host den optimalen Router, oder falls das Zielsystem in der eigenen Area liegt, dessen Subnetzwerkadresse selbst, mitteilt. Die darauffolgenden Pakete können, durch den nun vorhanden Eintrag im Cache, direkt an das richtige System geschickt werden. Durch die eintreffenden Datenpakete kann der Cache ständig aktualisiert werden. Hat der Host noch keine Router in seiner Adjaszenzliste, adressiert er einfach "all routers". Um das Füllen der Caches zu beschleunigen, kann ein Router zu einem von ihm generierten

"ES-IS redirect packet" eine Äquivalenzklasse von Hosts bilden, die alle Hosts enthält, für welche dieses "redirect packet" gültige Informationen enthält. Anstatt das "redirect packet" nur an den initierenden Host zu senden, kann der Router es nun an dessen ganze Äquivalenzklasse schicken und so gleich mehreren Hosts ermöglichen ihren Cache zu füllen.

- Dynamische Zuweisung von Netzwerkadressen
  Anders als die physikalischen Adressen können Netzwerkadressen nicht fest vorgegeben werden, da diese unter topologischen Gesichtspunkten vergeben werden müssen. Durch die Fähigkeit der Hosts selbstständig ihre eigenen Netzwerkadressen zu lernen, wird die Möglichkeit geschaffen problemlos die Adressen aller Hosts in einem Netzwerk zu ändern und den administrativen Aufwand zu reduzieren. Zum Beispiel kann sich eine Diskless Station, die beim Einschalten ihre Netzwerkadresse nicht kennt, sich diese einfach zuweisen lassen.
  Dies geschieht folgendermaßen: Braucht ein Host eine Netzwerkadresse schickt er ein "request address packet" an alle Router. Falls mehrere Router ein "assign address packet", welche auch die Gültigkeitsdauer der zu vergebenden Adresse enthält, zurückschicken, kann der Host selbst eine der angebotenen Adressen auswählen. An die Adresse gebunden ist er allerdings erst nach Senden eines ESH. Ein Router darf eine einmal angebotene Adresse vor Ablauf derer Gültigkeit nicht mehr verwenden, unabhängig davon, ob er nun ein ESH bekommt oder nicht. Um zu verhindern, daß unberechtigte Hosts eine Netzwerkadresse erhalten, ist es möglich nur solchen Hosts eine Netzwerkadresse zuzuweisen, welche in einem dafür vorgesehenen Server eingetragen sind. Wie der Router eine solche Netzwerkadresse bestimmt, wird im Standard nicht festgelgt. Eine einfache Möglichkeit wäre zum Beispiel das Anfügen eines Prefixes an die Subnetzwerkadresse.

2.2.2 Intradomain Routing

Innerhalb der einzelnen Domänen wird das Routing vom IS-IS (intermediate system to intermediate system) Protokoll übernommen. Dieses verlangt folgende Struktur von den NSAPs :



- High order DSP :
  Bildet zusammen mit AFI und IDI die Area-Adresse.
- System ID :
  Dient zur eindeutigen Identifikation eines Systems innerhalb einer Area.

Abb.4 : IS-IS NSAP Adress Struktur

IS-IS kennt drei verschiedene Typen von Routern :

- Intra-Area (Level 1) Router
- Inter-Area (Level 2) Router
- Exterior Router

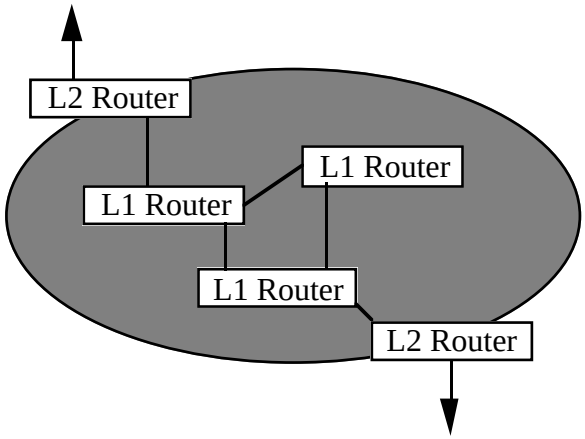


Abb.5 : IS-IS Area

Der Adreßraum innnerhalb der Areas (Level 1) ist eindimensional strukturiert. Um ein System zu erreichen muß man also die Area fluten. Dadurch ist es möglich, die Toplogie einer Area ohne Rekonfiguration der Systeme zu ändern. Um den Aufwand gering zu halten wird nur die System ID der Adresse übertragen. Die System IDs müssen areaweit eindeutig sein und sind von fester Länge im Bereich von einem bis 8 Oktetts, wobei meistens 6 Oktetts benutzt werden, da dann IEEE 802 MAC Adressen verwendet werden können. Ein Intra-Area Router weis nur, ob ein Zielsystem innerhalb seiner Area lokalisiert ist, oder nicht. Falls dies nicht der Fall ist leitet er das entsprechende Paket an den nächsten Inter-Area Router weiter. Der Level 2 Adreßraum ist auch wieder eindimensional ausgelegt, so daß auch hier zu allen Level 2 Routern geflutet wird. Wenn sich das Zielsystem in der Domäne befindet werden die entsprechenden Pakete zum nächten Zugangspunkt der Zielarea weitergereicht, von wo sie entsprechend dem schon beschrieben Level 1 Routing befördert werden. Befindet sich das Zielsystem nicht innerhalb der Domäne, werden die Pakete an einen Exterior Router weitergeleitet.

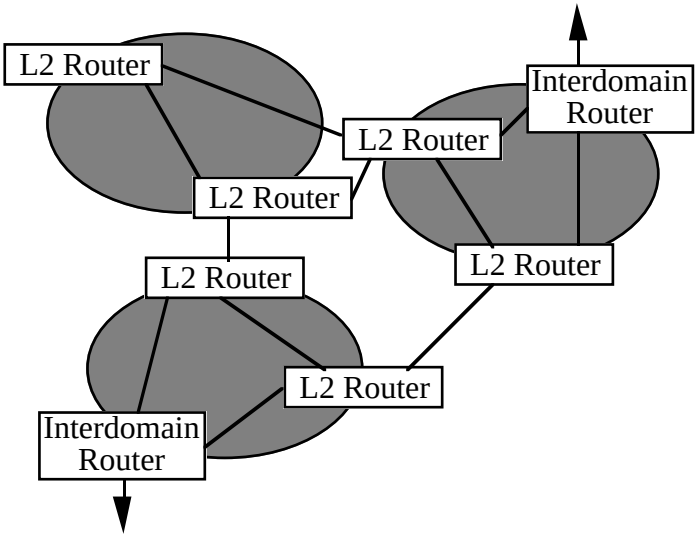


Abb.6 : Inter-Area Routing

Eine Area kann mehrere Adressen haben, wodurch Änderung von Area-Adressen und Aufspalten bzw. Zusammenfassen von Areas (innerhalb einer Domäne) einfacher zu realisieren sind. Wenn eine Area durch den Ausfall von Netzwerkkomponenten in Partitionen zerfällt, wird automatisch versucht diese Partitionierung auf Level 2 zu "reparieren". Eine Level 2 Partitionierung ist jedoch weder auf Level 1 noch durch Interdomain Routing aufzuheben.

2.2.3 Interdomain Routing

Das Routing zwischen den Domänen wird von IDRP (Inter Domain Routing Protocol) übernommen. Router, die am Interdomain Routing teilnehmen, werden als *Border Intermediate Systems* (BIS) bezeichnet. Eine Domäne kann mehrere BIS enthalten, die dann aber alle einer konsistenten Routing Strategie folgen müssen. IDRP regelt den Austausch von PDUs zwischen den einzelnen BIS, die Konstruktion von Routen und die Behandlung von Protokoll-Fehlern. Bei der Auswahl der Routen, handeln die BIS jedoch nach ihrer eigenen lokalen Strategie. Bei der Konstruktion von Routen werden alle bisher durchquerten Domänen im sogenannten RD (Routing Domain) Pfad festgehalten. Wird nun, beim Austausch von Routing Informationen, einem BIS eine Route angeboten, in deren RD Pfad es schon eingetragen ist, so ignoriert es die angebotenen Informationen. Dadurch wird die Bildung von Schleifen unterdrückt. Um die Komplexität der benötigten Routing Information zu reduzieren, können mehrere Domänen zu einer *Routing Domain Confederation* (RDC) zusammengefaßt werden, die dann nach außen wie eine einzelne Domäne erscheint.

2.3 Abbildung der IP Funktionalität auf CLNP

Die Verwendung von CLNP anstelle von IP verlangt, daß von der Transportschicht aus gesehen alle Funktionalität von IP genauso von CLNP bereitgestellt werden muß. Diese Anforderung ist ohne goßen Aufwand zu erfüllen, da CLNP von IP abgeleitet wurde und der Großteil der IP Funktionalität sich 1:1 auf CLNP abbilden läßt. Einige Änderungen sind dennoch nötig. Zum Beispiel muß die Aufgabe der Protokoll ID, welche bei IP das übergeordnete Protokoll identifiziert (z.B. TCP : 6, UDP : 17), bei CLNP vom SEL Feld in der Adresse übernommen werden. TCP und UDP machen bei der Berechnung von

Prüfsummen für ihre Segmente von sogenannten Pseudo Headern Gebrauch, um neben der Korrektheit der empfangenen Segmente auch überprüfen zu können, ob diese am richtigen Ziel angekommen sind. Ein solcher Pseudo-Header besteht aus folgenden Informationen :

- Zieladresse
- Länge der Quelladresse
- Quelladresse
- Protokoll-ID (zwei Oktetts)
- TCP/UDP Segmentlänge

Um diese Semantik zu erhalten, werden bei Verwendung von TUBA die Quell- und Ziel-NSAPs (inclusive Länge und NSEL) in die Prüfsummenberechnung von TCP und UDP mit einbezogen.

2.4 Der Übergang zu CLNP

Um einen nahtlosen Übergang von IPv4 zu CLNP zu ermöglichen, sieht TUBA eine so-genannte "Dual Stack Architektur" vor. Das heißt ein Host soll sowohl CLNP als auch IPv4 unterstützen. Nach und nach sollen dann die DNS (Domain Name Server), welche zur Auflösung der Hostnames zu IP-Adressen dienen, neben den IPv4 Adressen, auch noch die entsprechenden NSAPs liefern. Wenn nun ein Dual Stack Host auf eine Anfrage eine IPv4 Adresse und einen NSAP erhält, so weiß er, daß er sein Zielsystem über die CLNP Infrastruktur erreichen kann, erhält er nur eine IPv4 Adresse, verwendet er wie bisher IPv4. Diese Strategie erlaubt es, einen Host nach dem anderen der CLNP Infrastruktur hinzuzufügen, ohne ihm den Zugang zu den Systemen die noch nicht fähig sind CLNP zu benutzen, zu nehmen.

3 CATNIP (IPv7)

Ein anderer Ansatz führt zu CATNIP (Common Architecture for Next-Generation Internet Protocol), einer Architektur, die es ermöglicht jedes der Transportprotokolle TCP, UDP, IPX , SPX, CLTP und TP4 über jedem der Netzwerkprotokolle CLNP, IP, IPX und dem CATNIP Protokoll einzusetzen und somit auch auf eine schon durch TUBA fortgeschrittene CLNP Infrastruktur aufsetzen. Ein wichtiger Aspekt bei der Entwicklung von CATNIP ist, daß die Ersetzung einer Netzwerkkomponente durch ein CATNIP System keine Rekonfiguration der anderen Komponenetn nach sich ziehen soll.

3.1 Das CATNIP Datagrammformat

Die Adressierung in CATNIP geschieht ähnlich wie bei TUBA mittels NSAP-Adressen, denen noch ein Feld, mit der Länge der Adresse vorangestellt wird. Allerdings existiert hier kein NSEL Feld. Welche Routing Protokolle im Zusammenhang mit CATNIP eingesetzt werden sollen, wird nicht spezifiziert. Die von CATNIP erzeugten Datagramme sind wie folgt aufgebaut:

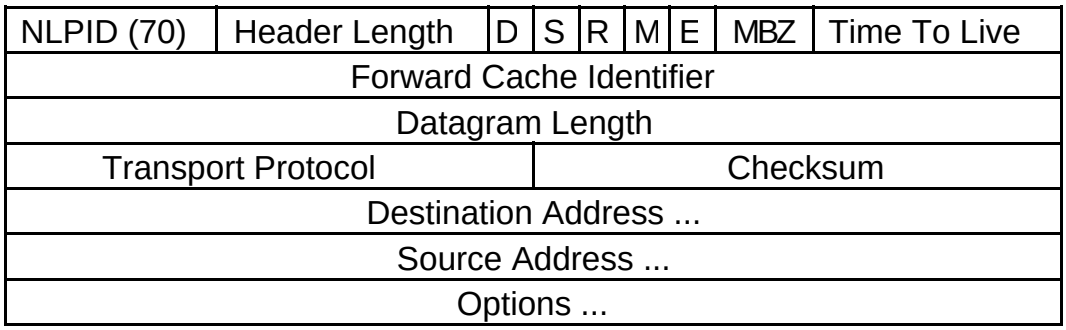


Abb.7 : Catnip Datagramm Format

- NLPID (8 Bit) : Network Layer Protokoll Identifier.
  Die Konstante 70h bedeutet Internet Version 7
- Header Length (8 Bit) : Anzahl der 32 Bit Wörter im Header
- Flags :
  - *Destination Address Omitted*
    Dieses Bit wird gesetzt, um anzuzeigen, daß die Zieladresse nicht im Header enthalten ist.
  - *Source Address Omitted*
    Dieses Bit gibt an, ob die Quelladresse im Header ausgelassen wurde.
  - *Report Fragmentation Done*
    Wenn dieses Bit gesetzt ist, sollte ein Router der das Datagramm fragmentiert, dies mit der ICMP Message "Datagram to big" melden.
  - *Mandatory Router Option*
    Ist dieses Flag gesetzt, so muß ein Router der das Datagramm weiterreicht dessen Optionen auslesen.
  - *Error Report Supression*
    Ist dieses Bit gesetzt, so sollen keine ICMP Messages gesendet werden.

  Die letzten 3 Bit sind reserviert und müssen auf 0 gesetzt werden.
- Time To Live (8 Bit)
  Zähler der bei jedem Hop um mindestens 1 dekrementiert werden muß.
- Forward Cache Identifier (32 Bit)
  Die Verwendung dieses Feldes wird im folgenden Abschnitt erklärt.
- Datagram Length (32 Bit)
  Länge des gesamten Datagramms.
- Transport Protocol (16 Bit)
  Kennung des übergeordneten Protokolls, z.B. 6 für TCP
- Checksum (16 Bit)
  Prüfsumme über den gesamten Header.

- Destination Address
  Dieses Feld besteht aus dem Ziel-NSAP, dem ein Byte mit dessen Länge vorangestellt wird. Eventuell folgen noch bis zu 31 Padding Bits. Dieses Feld ist nicht im Header enthalten, wenn das D-Flag gesetzt ist.
- Source Address
  Quell Adresse im gleichen Format wie bei Destination Address. Bei gesetztem S-Flag fehlt dieses Feld.
- Options
  Jede hier enthaltene Option besitzt einen eigenen 32 Bit Header, gefolgt von den Optionsdaten und gegebenenfalls bis zu 31 Bits Padding.

3.2 Der Forward Cache Identifier

Der Forward Cache Identifier ist ein 32 Bit Datenfeld im CATNIP Datagramm, welches - sofern im Hop die nötigen Informationen vorhanden sind - bei jedem Hop aktualisiert wird. Der Wert für dieses Feld kann aufgrund von ICMP Messages die vom nächsten Router zurückgeschickt werden, mit Hilfe eines Routing Protokolles oder einer anderen Methode bestimmt werden. Der FCI ermöglicht dem Router Entscheidungen bezüglich der Wegwahl zu beschleunigen und Datagramme in reservierten Flüssen zu halten.

3.2.1 Rückkopplung via ICMP

Benachbarte Router können sich mittels ICMP (Internet Controll Message Protocol) gegenseitig Cache Setup Messages zuschicken. Eine solche Message hat folgenden Aufbau:

Type	Code	Checksum
Forward Cache Identifier		
Valid Time in Seconds		
Addresses and Options ...		

Abb.8 : ICMP Cache Setup Message

Um die ICMP Message als Cache Setup Message zu kennzeichnen, enthält das Feld Type den Wert <tba>. Um sicherzugehen, daß die Message nur bis zum nächsten Router weitergereicht wird, wird im CATNIP-Header TTL (Time To Live) auf 1 gesetzt. Wenn ein Router nun ein Datenpaket empfängt, kann er für die Zieladresse des Paketes einen FCI einrichten. Unter diesem speichert er dann zum Beispiel die Zieladresse sowie die Optionen des Paketes und schickt eine Cache Setup Message mit dem Code 2 zurück an den Router von dem das Paket kam. Will dieser dann weitere Pakete an dieselbe Adresse schicken, so kann er in diesen Paketen, statt der Adresse und den Optionen, den FCI einsetzen; allerdings nur für den im Feld "Valid Time" der ICMP Message angegebenen Zeitraum. Code 3 berechtigt den Empfänger der ICMP Message zur Auslassung von Optionen, Quell- und Zieladresse. Bei Code 4 dürfen nur Quelladressse und Optionen ausgelassen werden. Die Codes 0 und 1 dienen zum Löschen aller Einträge bzw. Verbieten der Benutzung eines einzelnen FCIs.

3.2.2 Beispiel für die Nutzung der FCIs durch ein Routing Protokoll

Als Beispiel für die Nutzung der FCIs durch Routing Protokolle, wird im folgenden die mögliche Zusammenarbeit von CATNIP mit RAP (Route Access Protocol) erläutert. Dazu wird folgende Notation verwendet:

R(r,d,i,h)	Route von Router r zum Zielsystem d über Hop h wobei der FCI im Datagramm mit i beschrieben wird
Ri(r,d)	Interner FCI von Router r zum Zielsystem d
Ri(dgram)	Der FCI im Datagram dgram

Seien nun Router A,B,C und die Endsysteme X,Y gegeben.

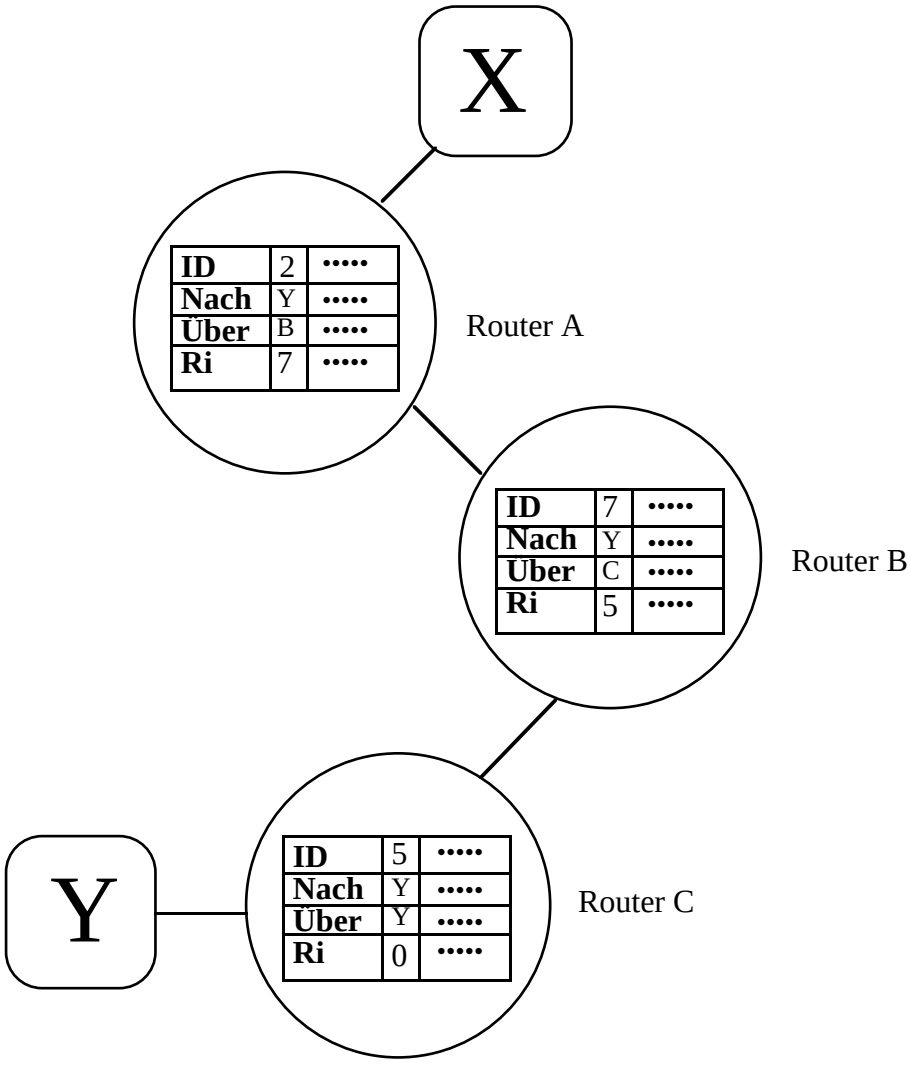


Abb.9 : Beispiel einer Route

Obige Situation könnte sich folgendermaßen ergeben haben :

- Router C gibt B die Route R(C,Y,0,Y) und die ID Ri(C,Y) (hier : 5) bekannt.

- Daraufhin bildet Router B die Route R(B,Y,Ri(C,Y),C) (hier : R(B,Y,5,C)) und gibt diese wiederum Router A bekannt. Mit der Route schickt B auch wieder Ri(B,Y) (hier : 7).
- In A wird letztendlich R(A,Y,Ri(B,Y),B) (hier : R(A,Y,7,B)) gebildet.

Die Übertragung eines Datagrammes von X nach Y sieht nun aus wie folgt :

- Das Endsystem X hat keine Routing Information und schickt das an Y adressierte Paket (mit Ri(dgram) = 0) zu Router A.
- Da dieser im Datagram nun keinen nutzbaren FCI vorfindet, sucht er in seinen Routing Tabellen nach Y. Dort findet er die Route R(A,Y,7,B). Daraufhin setzt er Ri(dgram) auf 7 und leitet das Paket weiter an B.
- Router B braucht nun nur Ri(dgram) auszulesen um damit sofort die Route R(B,Y,5,C) aufzufinden. Er setzt dann seinerseits Ri(dgram) auf 5 und leitet das Datagramm zu C weiter.
- Liest nun Router C den FCI Ri(dgram) kann er sofort das Paket mit Ri(dgram) = 0 über R(C,Y,0,Y) an Y ausliefern.

Um nicht Routen zu allen möglichen Zielen verwalten zu müssen, ermöglicht RAP den Routern, die ihnen bekanntgegebenen Routen zu filtern (falls das Ziel z.B. zu weit entfernt liegt oder die Route zu speziell ist) und Routen zu Aggregieren.

3.3 Übersetzung der Datagramme

Bei der Kommunikation mit nicht-CATNIP-fähigen Systemen, übersetzen CATNIP Systeme vor dem Versenden ihre Datagramme, je nach Typ ihres Kommunikationspartners, in IPv4-, CLNP- oder IPX-Datagramme. Dabei müssen fragmentierte Datagramme vor dem Übersetzen reassembliert werden. Das heißt aber auch, daß Datagramme, deren Fragmente auf unterschiedlichen Pfaden weitergeleitet werden, verlorengehen können.

3.3.1 CLNP

Die Konvertierung zwischen CATNIP und CLNP Datagrammen ist haupsächlich durch Verschieben von Feldern zu bewerkstelligen, da die Adressen nicht konvertiert werden müssen. Handelt es sich jedoch um ein CLNP Datagramm von Typ "Error Report", so muß dieses in die entsprechende ICMP-Message konvertiert werden. Bei der Konvertierung von CATNIP nach CLNP ist diese Tatsache natürlich auch zu berücksichtigen.

3.3.2 IPv4

Bei der Übersetzung zwischen IPv4 und CATNIP müssen natürlich die verschiedenen Adressen aufeinander abgebildet werden. Dazu muß von der ISO ein neuer AFI an die IANA vergeben werden. Dieser identifiziert alle heute exitierenden IPv4 Adressen als zum Internet gehörend. Gleichzeitig legt er die Länge des IDIs auf 2 Bytes fest, welche die AD (Adressing Domain) Nummer aufnehmen sollen, fest. Diese AD Nummern werden dann von der IANA

vergeben. Anfangs existiert nur die AD Nummer 0.0, welche dem InterNIC zugewiesen wird, und somit das ganze heute existierende Internet betrifft. Die Abbildung zwischen den verschidenen Adressen sieht folgendermaßen aus :

length	AFI	IDI ...	DSP ...
7	xyz	AD number	version 4 address

Abb.10 : Abbildung der IPv4-Adressen auf CATNIP-Adressen

Hierbei sei *xyz* der von der IANA zugewiesene *Authority and Format Identifier*.
Der DSP, der anfänglich immer die 4 Bytes der IPv4 Adresse enthält, kann beliebig erweitert werden. Systeme deren DSPs länger als 4 Bytes sind können dann aber nicht mehr mit IPv4 Systemen kommunizieren. Wenn ein Datagramm nach IPv4 konvertiert wird, können diejenigen Bytes die über die normale IPv4 Adresse hinausgehen, im Optionenfeld des IPv4 Headers gesichert werden. Dadurch kann bei einer Rückkonvertierung die gesamte Adresse wiederhergestellt werden. Die direkte Implementierung dieser "Address Extention Option" in IPv4, führt zu sogenanten IPv4 Hybridsystemen. Diese sind dann fähig, mit Hilfe von in der Nachbarschaft lokalisierten CATNIP Systemen, auch mit Hosts zu kommunizieren, deren DSPs länger als 4 Bytes sind.

3.3.3 IPX

Das von Novell entwickelte IPX hat viele Fähigkeiten mit den Internetprotokollen gemeinsan. Bei IPX/SPX werden jedoch die Schichten 3 und 4 nicht strikt getrennt. IPX entspricht in etwa UDP/IP während SPX, mit dem Netzwerkteil von IPX, ungefähr TCP/IP entspricht. Dies wirkt sich aber hauptsächlich auf Reihenfolge und Namen von Feldern in den Datagrammen aus, und nicht so sehr auf die eigentliche Protokoll Architektur. Dennoch ist eine gegenseitige Abbildung der Transportschichten über IPX und IP nicht so sinnvoll, wie es erscheinen mag, da IP keine IPX-Adressen versteht und umgekehrt. Wenn sich aber die CATNIP Infrastruktur etabliert hat, können IPX Systeme über diese miteinander kommunizieren. CATNIP Hosts können sich auch eine IPX-Adresse zuweisen lassen, was sie dann befähigt mit IPX Systemen zu kommunizieren. Die Netzwerkadressen in IPX bestehen aus einer 32 Bit Netzwerknummer und einer 48 Bit langen ID, welche bei LANs meistes eine MAC Layer Adresse enthält. Diese Netzwerkadresse wird folgendermaßen auf eine CATNIP Adresse abgebildet :

length	AFI	IDI ...	DSP ...
13	47h	Novell ICD	Netzwerk + MAC Adresse

Abb.11 : Abbildung von IPX-Adressen auf CATNIP-Adressen

Bei der Übersetzung der Pakete muß hier ein wenig mehr Aufwand betrieben werden. Zum Beispiel hat IPX anstelle eines TTL Feldes das "Transport Control Field", welches mit 0 initialisiert und bei jedem Hop inkrementiert wird, bis es den Wert 16 erreicht hat. Bei jeder Konvertierunng müssen also diese Werte umgerechnet werden.

3.3.4 SIPP

SIPP (IPv6) ist ein IPng-Ansatz einer anderen IETF (Internet Engineering Task Force)-Working Group. Falls CATNIP und SIPP, beide zum Einsatz kommen, muß CATNIP auch in der Lage sein, von bzw. nach SIPP zu übersetzen. Auf SIPP wird hier nicht näher eingegangen, da dies in einem der folgenden Seminarartikel geschieht. Im NSAP Adressierungsplan erhält SIPP den gleichen AFI wie IPv4 und als AD Nummer wird SIPP 0.1 zugewiesen.

3.4 Anpassung auf Transportebene

Das von IPv7 verwendete ICMP Protokoll unterscheidet sich kaum von dem bei IPv4 benutzten, so daß in den meisten Fällen keine Konvertierung von Nöten ist. Einzig bei der Interoperation mit CLNP muß übersetzt werden, da CLNP statt ICMP Messages eigene Fehlermeldungs-Pakete versendet. Da in ICMP Error Messages die IP Datagramme, die den Fehler verursacht haben, eingebettet sein können, kann es vorkommen, daß die Überstzungsroutinen von CATNIP rekursiv aufgerufen werden müssen. Bei TCP und UDP muß natürlich wieder, wie schon bei der Beschreibung von TUBA erläutert, die Berechnung der Prüfsummen angepaßt werden. Es wird vorgeschlagen, daß Hosts, welche von erweiterten Adressen Gebrauch machen, nur die letzten 4 Bytes zur Prüfsummenberechnung heranziehen.

4 Conclusio

Nach aktuellen Schätzungen der IETF, wächst das Internet derzeit so stark an, daß die letzte freie IPv4 Adresse in 6 - 12 Jahren vergeben werden müßte. Allerdings werden schon früher massive Probleme beim Routing auftreten, da die Menge der vom Router zu haltenden Daten mit der Anzahl der Subnetzwerke im Internet wächst. Deshalb ist IPng momentan eines der hauptsächlichen Betätigungsfelder der IETF. Die in diesem Beitrag beschriebenen Ansätze werden aber wahrscheinlich nicht als IPng zum Einsatz kommen, da zur Zeit SIPP als IPng Ansatz eindeutig favorisiert wird.

5 Literatur

/KaFo93/ Katz, Dave ; Ford, Peter S.
TUBA-Replacing IP with CLNP
IEEE Network, May 1993.

/FRK94/ Ford, Peter S. ; Rekhter, Yakov ; Knopper, Mark
TUBA : CLNP as IPng
ConneXions Vol. 8, No. 5, May 1994.

/Ull94/ Ullmann, Robert L.
CATNIP - Common Architecture for Next-generation Internet Protokoll
Internet Draft , 21 March 1994.

/Har92/ Hares, Susan
Components of OSI : IDRP
ConneXions, Vol. 6 No. 5 May 1992.

MULTICAST BACKBONE — GRUPPENKOMMUNIKATION IM INTERNET

Kay Uwe Fischer

Bisher gab es im Internet zur Verbreitung von Daten nur zwei Möglichkeiten: “Unicast” (an einen Teilnehmer) und “Broadcast” (an alle Teilnehmer). Multicast ist die für die Gruppenkommunikation nötige Kommunikationsbeziehung, die die Übermittlung von Daten an eine beliebige Zahl von Internetteilnehmern ermöglicht. Das Multicast Backbone (MBone) stellt ein virtuelles Netzwerk dar, welches aus miteinander vernetzten, das Multicast-Protokoll unterstützenden IP-Routern aufgebaut ist. Es bildet die Grundlage für neue Anwendungen der (Gruppen-) Kommunikation im Internet. Diese beinhalten sowohl Audio- als auch Audio-/Video- Übertragungen und Konferenzen. Für diese Arten der Kommunikation sind neue Protokolltechniken und hohe Netzleistungen sowie gute Hard- bzw. Softwarekompressionsalgorithmen nötig. Die erforderlichen Applikationen vom Verzeichnisprogramm (sd), das alle Übertragungen des jeweiligen Teilnetzes auflistet, bis hin zum Videokonferenzprogramm existieren bereits und werden kontinuierlich weiterentwickelt.

Einführung - Multicast

In den letzten Jahren gewannen die multimedialen Anwendungen im Bereich der modernen Datenverarbeitung immer mehr an Bedeutung. Es ist dabei die Rede vom “Information Superhighway”, interaktiven Fernsehen, Video on Demand oder wie in diesem Beitrag von audiovisueller Gruppenkommunikation. Diese Entwicklung macht auch vor dem Internet als dem größten weltweiten Netzverbund nicht halt. Um die großen Datenmengen dieser multimedialen Applikationen verarbeiten zu können, müssen jedoch neue Verfahren, Protokoll- und Übertragungstechniken entwickelt und weltweit installiert, sowie die vorhandenen Netzkapazitäten ausgebaut werden.

Multicast stellt nun eine, für das Internet neue, Kommunikationsbeziehung dar. Es ist als Ergänzung zu den im Internet üblichen Unicast (Datenübertragung zwischen zwei Teilnehmern) und Broadcast (Übertragung der Daten an alle angeschlossenen Teilnehmern) zu sehen und erlaubt die Übertragung von Daten innerhalb einer Gruppe von Internetteilnehmern über das normale Internet.

Das Multicast-Verfahren hat unter anderem folgende Vorteile:

- geringere Verbindungsanzahl
  Es müssen nicht wie beim Unicast mehrere völlig voneinander getrennte Verbindungen parallell zueinander aufgebaut und verwaltet werden. Dies stellt eine Entlastung der Netzressourcen dar und verschafft dem Sender mehr Transparenz, da er die Daten nur einmal senden muß, was für ihn eine Entlastung darstellt.
- geringere Netzbelastung
  Die an die Gruppenteilnehmer zu sendenden Daten werden nicht an alle Netzteilnehmer gesendet, wie dies beim Broadcast der Fall ist, sondern nur an die Netzteilnehmer, die der Gruppe angehören. Dies hat klare Vorteile in Bezug auf die Auslastung des Netzes, da weniger Daten über das Netz versendet werden als bei mehreren Unicast-Verbindungen wird dieses durch die Multicast-Verbindung weniger belasten.

Wie diese Vorteile im Einzelnen zustande kommen, wie sie genutzt werden können und welche Voraussetzungen erfüllt sein müssen wird in der Folge erläutert.

MBone - Multicast Backbone

Um Multicast-Verbindungen im Internet zu unterstützen wurde das Internet-Protokoll erweitert. Das Problem dabei ist jedoch, daß die meisten der heute vorhandenen Internet-Router die Multicast-Pakete noch nicht verarbeiten können. Dafür ist ein zusätzliches Programm, der Multicast Routing-Daemon “mrouted”, sowie eine Multicast-Unterstützung im Kernel des Rechners nötig. Da diese gegenwärtig noch nicht generell implementiert ist, bedient man sich eines relativ einfachen Verfahrens um trotzdem Multicast-Verbindungen im Internet aufbauen zu können: Der Multicast-Router verpackt die MBone-Multicastpakete in ganz normale Internet-Protokoll-Datenpakete (siehe Abbildung 1) und verschickt diese so, über die nicht

multicastfähigen Router, zu einem anderen Multicast-Router, der diese dann auspackt und als MBone-Multicastdatenpakete weiterverarbeitet.

Abbildung 1: In IP-Paket gekapseltes Multicast-Datenpaket

Dabei setzt der sendende Multicast-Router dem MBone-Multicastdatenpaket einen normalen IP-Header voran und trägt in diesen als Zieladresse die IP-Adresse des nächsten Multicast-Routers und als Next-Protocol-Id die Zahl vier ein. Dies bedeutet, daß das nächste Protokoll das IP-Protokoll ist. Der Multicast-Router, der auf den zu überbrückenden IP-Router folgt, empfängt das eingepackte MBone-Multicastpaket und erkennt an der Next-Protocol-Id und der Multicast-Zieladresse, daß es sich um ein eingepacktes MBone-Multicastpaket handelt. Er entfernt den IP-Header und leitet das Multicast-Paket weiter, wenn nötig wieder über einen IP-Router hinweg (siehe Abbildung 2).

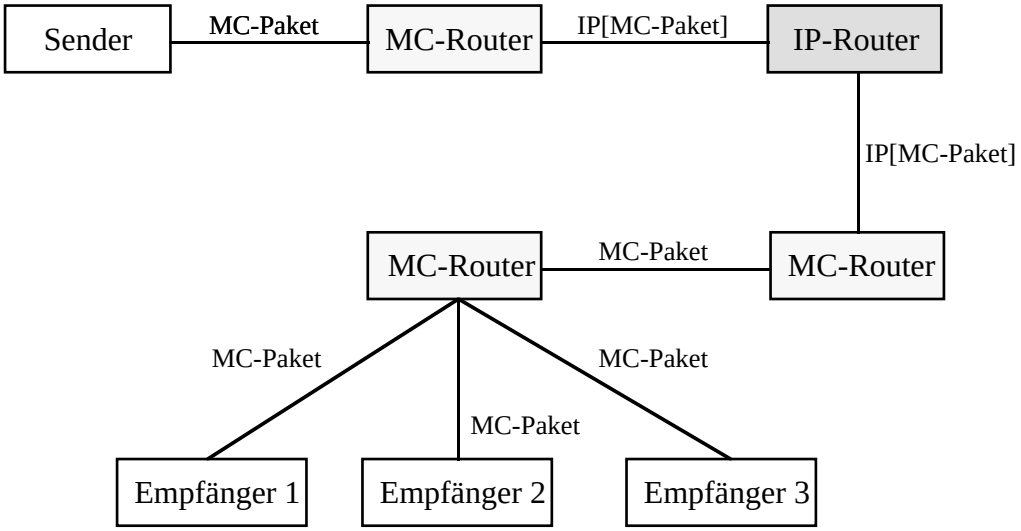


Abbildung 2: Beispiel für das Einpacken von MC-Paketen in IP-Pakete

Diese Verbindungen zwischen zwei Multicast-Routern über einen oder mehrere normale IP-Router hinweg werden als “Tunnel” bezeichnet, wobei mehrere Multicast-Router zusammen ein virtuelles Multicast-Netzwerk bilden (siehe Abbildung 3). Das Multicast Backbone (MBone) ist solch ein virtuelles Netzwerk aus Multicast-Routern in der ganzen Welt.

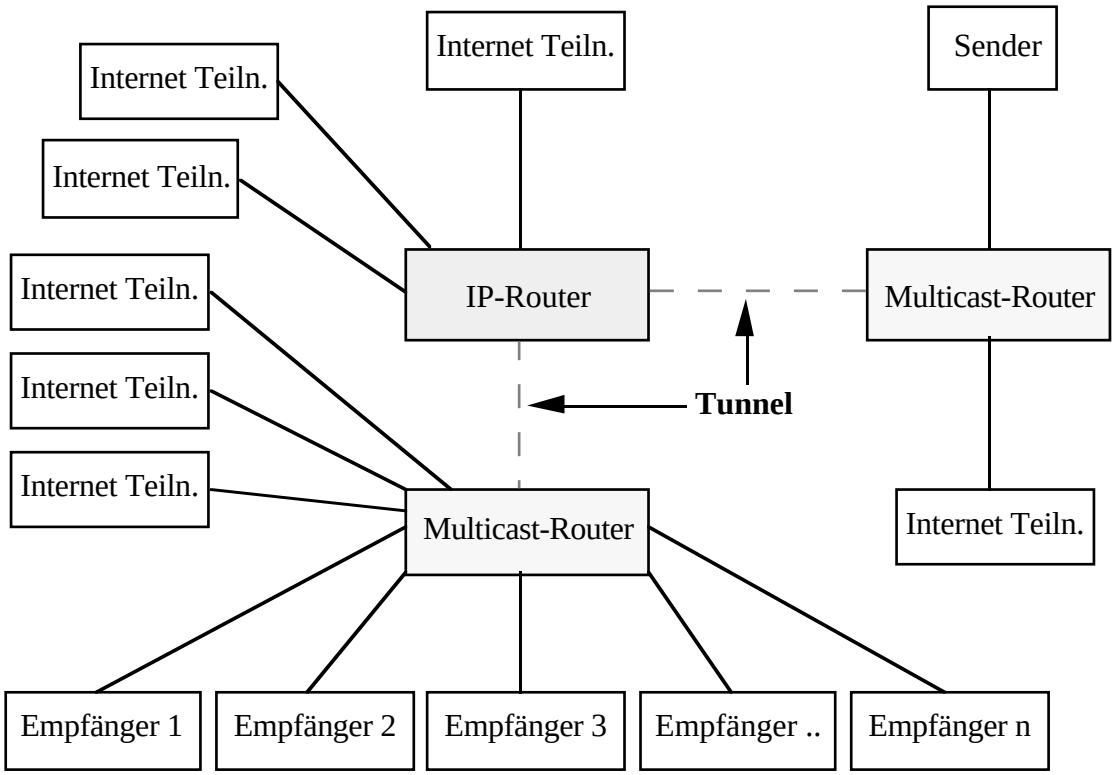


Abbildung 3: Multicast-Netzwerk mit Tunnel

Anforderungen an das Netzwerk

Durch die großen Datenmengen, die bei Videoübertragungen auftreten, sind die Anforderungen an die benutzte Netzinfrastruktur relativ groß. Eine Übertragung von Video- und Audiodaten (25 Bilder/s mit 24 Bit Farbtiefe) schlägt komprimiert (JPEG oder MPEG) mit zirka 1 MBit/s zu Buche - ohne Kompression wäre das ein Datenaufkommen von fast 20 MBit/s (Angaben siehe /Thissen/), was also höchstens von FDDI, ATM, DQDB oder Fast Ethernet verarbeitet werden könnte. Damit die Netzbelastung nicht zu groß ist, wird die Videobandbreite bei weltweiten Videoausstrahlungen heute freiwillig auf 128 kBit/s beschränkt, das entspricht 4 bis 6 Bildern pro Sekunde. Als weitere Maßnahme zur Begrenzung der Netzlast wird die Lebenszeit (time-to-live) der Multicast-Pakete beschränkt und von jedem Router den sie passieren heruntergesetzt. Dabei bedeutet ein Wert von ca. 16 für die Lebenszeit eines Paketes dessen Verbreitung auf einem relativ kleinen Bereich des MBone (es kommt nur 16 Multicast-Router weiter), wohingegen ein Wert von 128 - 255 den Datenstrom an jedes Subnetz des MBones weiterleitet. Diese Restriktionen und Möglichkeiten zur Restriktion sind notwendig, da sonst die bestehenden Netzwerke durch eine oder mehrere Videokonferenz völlig überlastet bzw. lahmgelegt würden.

Aus den oben genannten Gründen sollte das zugrundeliegende Netzwerk in der Lage sein mindestens 2 MBit/s an Daten verarbeiten zu können. In Deutschland kommen deshalb als Benutzer vornehmlich Universitäten mit Zugang zum Wissenschaftsnetz (WIN) oder BelWü-Netz (Baden-Württemberg Extended LAN) in Betracht. Im Wissenschaftsnetz sind die Übertragungsraten jedoch für die meisten Teilnehmer nicht höher als 64 kBit/s. Diese hohen Anforderungen an die benötigte Bandbreite reduzieren den Kreis der möglichen Teilnehmer am Multicast Backbone drastisch. Im Gegensatz dazu sind in den USA MANs oder WANs mit Kapazitäten von 10 MBit/s keine Seltenheit, weshalb die MBone-Technik dort weitaus verbreiteter ist als in Deutschland und anderen Ländern.

Nachteile des MBone

Das im letzten Abschnitt erwähnte hohe Datenaufkommen bei Multimediaübertragungen über das Internet und die teilweise noch relativ geringen Netzkapazitäten erfordern von den Netzteilnehmern eine freiwillige Selbstbeschränkung, um nicht das gesamte Netz lahmzulegen. Ein Beispiel dafür wäre die Beschränkung der Videobandbreite auf 128 kBit/s. Dieses Gebot der Fairneß gegenüber den anderen Netzteilnehmern ist eine Grundvoraussetzung für das reibungslose Funktionieren des Multicast-Backbone. Es können jedoch keine Garantien dafür gegeben werden, daß sich alle Multicast-Netzteilnehmer an diese freiwilligen Beschränkungen halten und nicht durch Audio-/Videoübertragungen die gesamte Bandbreite eines Teilnetzes erschöpft wird. Es besteht nämlich keine praktikable Möglichkeit zur Kontrolle der durch die Multimediaanwendungen benutzten Bandbreite.

Umgekehrt können den Multimediaanwendungen keine Garantien über die angeforderten Verbindungen gegeben werden. So ist zum Beispiel nicht sichergestellt, das eine Verbindung dauerhaft mit einer bestimmten Datenrate betrieben werden kann, was sich zum Beispiel bei wichtigen Multimedia-Konferenzen negativ auswirkt. Dies gilt vor allem da die Anzahl der Multimediaübertragungen in naher Zukunft stark zunehmen wird. So kann beispielsweise der nicht zeitkritische Filetransfer auf einem Teilnetz der parallel dazu laufenden Realzeit-Videoübertragung die erforderliche Bandbreite nehmen, diese stören oder die Übertragung sogar ganz unterbinden.

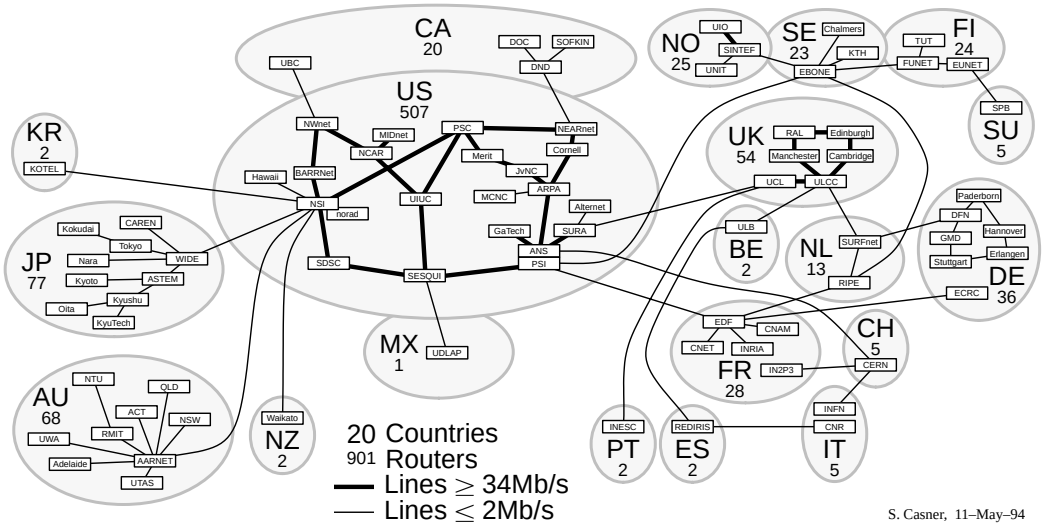
Ein weiterer Nachteil des Multicast Backbone besteht im heute noch nötigen Overhead in der Verwaltung und dem Betrieb, z. B. den parallel zu den IP-Routern laufenden Multicast-Routern und den teilweise benötigten Kernelpatches, die nötig sind, um die Rechner multicastfähig zu machen.

Struktur

Aufgrund der weiter oben erwähnten hohen Anforderungen an die Bandbreite des Netzes kommen nur wenige Teile des Internet als Übertragungsstrecken oder Netzknoten in Betracht. Dabei gibt es für jedes Land, das an MBone angeschlossen ist einen oder mehrere Knotenrechner, die die Verbindung in die angrenzenden Länder aufbauen und die Daten im eigenen Land weiterverteilen. In Deutschland übernimmt diese Aufgabe der MBone-Router des DFN-Vereins Düsseldorf, der die internationalen MBone-Pakete aus Amsterdam erhält und sie nach Paderborn, Stuttgart und/oder Köln weiterleitet, von wo aus diese weiterverteilt werden - unter anderem auch nach Karlsruhe.

Es gibt zwischen den Knoten Primär- und Backupverbindungen, sogenannte "Backup-Links". Die Backupverbindungen dienen dazu, bei Störungen in einzelnen Netzsegmenten, eine weitere Übertragung im Gesamtnetz sicherzustellen. Die Struktur des MBone in Deutschland zeigt die Abbildung 4 die dem Stand vom 26.10.1994 entspricht.

Major MBONE Routers and Links



S. Casner, 11–May–94

der Weitergabe der Liste der Gruppenteilnehmer an die anderen Multicast-Router nicht mehr an, womit er der Gruppe nicht mehr angehört.

Dieses Verfahren ist ziemlich aufwendig, da die Multicast-Router für jeden Teilnehmer an einer Multicast-Gruppe dynamisch einen Quellpfad berechnen müssen, was bei der heutigen Größe des MBone kein Problem darstellt aber in Zukunft mit wachsenden Teilnehmerzahlen, zunehmender Größe des Netzes und den daraus resultierenden Gruppenwechseln schnell zu einem Engpaß werden kann. Die Forschung und Entwicklung auf diesem Gebiet geht deshalb stetig weiter.

Multicast - Anwendungen

Existierende Anwendungen

Zur Zeit sind die folgenden auf dem Multicast Backbone basierenden Applikationen verfügbar. Die Weiter- und Neuentwicklung dieser und anderer Anwendungen im Multicast-Bereich wird jedoch ständig fortgeführt.

SD - Session Directory

Dieses einfach zu bedienende Verzeichnisprogramm listet alle gerade aktiven Übertragungen des jeweiligen MBone-Teilnetzes auf und bietet zusätzlich die Möglichkeit, eine neue Übertragung zu starten. Es ist dabei möglich festzulegen, wie weit die Ausstrahlung im MBone gehen soll und welches der nachfolgend aufgelisteten Programme dazu verwendet werden soll.

WB - White Board

Dieses Programm stellt eine virtuelle Wandtafel (White Board) dar, auf der mehrere Personen mit unterschiedlichen Farben schreiben, zeichnen oder Bilder aufkleben können. Dabei stehen die Standardzeichenfunktionen (Linie, Kreis, Text,...), sowie eine Importmöglichkeit für PostScript- und ASCII-Dokumente zur Verfügung. Bei Videofernkonferenzen kann es dazu benutzt werden, Folien zu zeigen oder die Ausführungen, die über die anderen Audio- oder Videoapplikationen übertragen werden, zu illustrieren. Durch mitführen einer Folgenummer in jeder übertragenen Mitteilung ist es möglich festzustellen, ob eine Nachricht auf dem Übertragungswege verlorengegangen ist. Wenn einer der Empfänger von Nachrichten bemerkt, daß eine Nachricht nicht angekommen ist, so kann er nach einer gewissen Wartezeit eine neuerliche Übertragung der Daten beim Sender anfordern.

VAT - Visual Audio Tool

Dieses Tool erlaubt die Kommunikation von mehreren Teilnehmern in einer Audiokonferenz über das Multicast Backbone, wobei der gerade sprechende Teilnehmer automatisch angezeigt wird. Die Qualität der Übertragung ist dabei durch die Samplingrate von 8000 Hz sehr gut. Die Eingabe erfolgt über ein Mikrofon oder den Line-in-Eingang, die Ausgabe über den Lautsprecher, den Kopfhörer oder über den Line-out-Ausgang zum Verstärker.

Die Audiodaten werden vor dem Versenden über das MBone komprimiert (mittels Adaptive Delta Pulse Code Modulation - ADPCM), um das Datenaufkommen und die Laufzeit zu reduzieren. Des weiteren überträgt VAT nur dann Daten, wenn auch wirklich etwas gesprochen wird. Die Audiodatenpakete werden numeriert, um beim Empfänger durch ordnen der ankommenden Datenpakete, einen sequentiellen Strom von Audiodaten erzeugen zu können.

NEVOT - Network Voice Terminal

NEVOT ist wie VAT ein Tool für Audiokonferenzen und unterstützt die von VAT versendeten Audiopakete. Durch sein modulares Design und den frei zugänglichen Sourcecode ist NEVOT gut geeignet, um sich unverbindlich erste Eindrücke zu verschaffen.

NV - Network Video

Diese Anwendung wurde zur Übermittlung von Videosignalen entwickelt und unterstützt die Versendung der Videosignale im PAL- und NTSC-Format (deutsches bzw. amerikanisches Fernsehformat). Das Videosignal kann dabei wahlweise in 8-Bit-Graustufen- oder 24-Bit-Farbdarstellung ausgegeben werden. Die Videoeingabe erfolgt über spezielle Videokarten, die je nach Rechnertyp unterschiedlich sind. Die Auswahl an solchen Karten ist jedoch zur Zeit noch recht gering.

Die zu versendenden Videodaten werden wie beim folgenden IVS mittels MPEG oder JPEG in Echtzeit komprimiert. Dazu werden Videoboards mit speziell dafür ausgelegter Hardware benutzt.

Die Übertragung von Audiosignalen unterstützt das Net Video Tool nicht, dazu wird üblicherweise das Visual Audio Tool benutzt.

IVS - Inria Videoconference System

IVS stellt wie Net Video ein Videokonferenzsystem dar. Es gibt jedoch einige Unterschiede zu NV. So benutzt IVS einen anderen Kompressionsalgorithmus und liefert daher auch andere Auflösungen und Farbtiefen. Die Bildwiederholraten von IVS liegen leicht über denen von NV. Da IVS jedoch nicht so aktuell ist wie Net Video, wird für Videoübertragungen derzeit überwiegend Net Video benutzt.

IVS integriert zusätzlich zur Videoübertragung die Audiodatenübertragung, die jedoch in der Praxis wegen ihrer, durch die PCM-Kodierung verursachten, schlechteren Qualität nicht

benutzt wird. Statt dessen wird dazu VAT benutzt, dies bedeutet Videoübertragung mit IVS und gleichzeitig Audioübertragung mit dem VAT.

Weitere mögliche Multicast-Anwendungen für die Zukunft

Im Folgenden seien einige mögliche Anwendungen für das Multicast Backbone genannt, die vielleicht schon bald implementiert werden.

Network News

Anstelle des heute üblichen NNTP zur Verbreitung von Nachrichten über das Internet, könnten die News-Server als eine Multicast-Gruppe betrachtet, und so die Übertragung gezielter und schneller vorgenommen werden.

Abfrage verteilter Datenbanken

Das Multicast-Verfahren ist ein sehr effizienter Weg um verteilte Datenbanken zu durchsuchen. Man könnte sich zum Beispiel vorstellen, eine Anfrage in einer Multicast-Datenbankgruppe zu stellen, deren Lebenszeit (time-to-live) relativ niedrig ist. Sollte keine Antwort einer Datenbank auf die Anfrage erfolgen, so erhöht man diese Lebenszeit und damit die Verbreitung der Anfrage. Mit diesem Verfahren würden die Netzresourcen geschont.

Weitere Entwicklung des MBone

Durch die zunehmende Bedeutung und Verbreitung der Multimediaanwendungen (Stichwort Information-Highway) und die ständig wachsenden Rechner- und Netzwerkkapazitäten werden sich die Multicast-Verbindungen und Anwendungen auf dem Internet in naher Zukunft weiter etablieren. Die Zahl der multicastfähigen IP-Router wird schnell steigen und die heute noch nötigen “Tunnel” zwischen zwei multicastfähigen Routern über einen IP-Router hinweg, werden mehr und mehr überflüssig. Das virtuelle Netzwerk bestehend aus multicastfähigen Routern - das Multicast Backbone - wird das gesamte Internet überspannen.

Das bedeutet, daß das Multicast Backbone in den nächsten Jahren im Internet aufgeht und als eigenständiger Bestandteil zwangsläufig an seiner heutigen Bedeutung verliert, da das Internet dessen im Moment noch spezielle Funktionen mit übernimmt.

Das heißt, das Internet wird zukünftig generell multicastfähig sein und unterstützt damit auf längere Sicht alle Multimediaanwendungen, die diese speziellen Funktionen erfordern.

Die Entwicklung des Multicast Backbone ist noch lange nicht abgeschlossen. So ist zum Beispiel die Überwachung der erforderlichen Bandbreite und deren Bereitstellung ein noch nicht gelöstes Problem, welches nur durch die Einführung und Handhabung von Dienstgüteparametern entschärft werden kann.

Weitere Informationen zum Multicast Backbone

Weitergehende Informationen zum Multicast Backbone und den dazu erhältlichen Programmen und Tools sind über die folgenden World-Wide-Web-Server zugänglich:

- http://www-ks.rus.uni-stuttgart.de/mbone/mbone.html
  Diese WWW-Seite kann als deutscher Einstiegspunkt in die MBone-Sektion des World-Wide-Web dienen.
- http://www.fokus.gmd.de/mtl/mbone
  Dies ist die Informationsseite der Gesellschaft für Mathematik und Datenverarbeitung Deutschland zum Multicast Backbone. Sie enthält unter anderem Links auf die aktuellen Karten der MBone-Struktur und Beispielsessions mit verschiedenen Multicasttools (nv, vat, ...).

Über die beiden angegebenen WWW-Seiten können fast alle Informationen zum Multicast Backbone erreicht werden. Diese wird durch Links auf verschiedene WWW-Seiten in der ganzen Welt realisiert. Hier besteht Zugriff auf einführende Informationen, FAQs, Grundlagen, Berichte von Anwendern, Beispielsitzungen und vieles mehr.

Literatur

/Thissen/ Thissen, Thomas;
In Hörweite
iX April 1994, Seite 90-96

/Bunn/ Bunn, Jean;
MBONE (Multicast Backbone)
Geneva University, 27 Januar 1994

/MacBrutz/ Macedonia, Michael R.; Brutzman, Donald P.;
MBone Provides Audio and Video Across the Internet
IEEE Computers, April 1994, Seite 30-36

/FAQ/ Casner, Stephen
Frequently Asked Questions (FAQ) on the Multicast Backbone (MBone)
Anonymous ftp from nic.sura.net in /pub/mbone, Mai 1993

/CasDee/ Casner, Stephen; Deering, Stephen;
First IETF Internet Audiocast
ACM Sigcomm Computer Communications Review, Vol. 22, Seite 92-97, Juli 1992

/Lockwood/ Lockwood, John;
Proposed Mbone/Xunet Extensions Revision 1.7.6
Anonymous ftp from ipoint.vlsi.uiuc.edu in /pub/ipoint/Documents/xbone.ps,
07. September 1993

RTP — Übertragung von Audio/Video im Internet

Alfio De Pasquale

Der Wunsch auch über größere Entfernungen effizient Realzeitdaten austauschen zu können, gewinnt immer mehr an Bedeutung. Daher wird in naher Zukunft die Internetfamilie um das Real-Time Transport Protocol (RTP) und das Real-Time Control Protocol (RTCP) erweitert, die dies unterstützen sollen. RTP und RTCP werden von der Audio/Video Transport Working Group (AVT) einer Arbeitsgruppe der Internet Engineering Task Force (IETF) entwickelt. Mit dem weit gefaßten Begriff "Realzeitdaten" ist schon angedeutet, daß bei der ursprünglichen Idee nur die Anwendungsklasse, nicht aber der eigentliche Verwendungszweck der Nutzdaten interessiert hat. Bei der Audio- und Videoübertragung handelt es sich aber wohl um eine Standardanwendung, die sich zudem besonders als Beispiel eignet, weswegen im folgenden häufig darauf Bezug genommen wird.

1. Einleitung

Die Übertragung von Multimedia-Daten über das Internet, wie z.B. die Echtzeitübertragung von Sprach- und Bildinformation, wird durch aktuelle Protokolle der Internetfamilie nicht unterstützt, d.h diese bieten unter anderem keinerlei Mechanismen zur Garantie von Qualitätsparametern wie Durchsatz, Verzögerung, Jitter, etc. Aus diesem Grund schloß sich die Audio/Video Transport Working Group (AVT) zusammen, mit der Zielsetzung experimentelle Protokolle zu entwickeln, die die Übertragung von Multimedia-Daten in der Internetwelt erlauben. Ergebnisse dieser Aktivitäten sind das Real-Time Transport Protocol (RTP) und das Real-Time Control Protocol (RTCP). Zunächst liegt das Interesse bei den Eigenschaften von RTP/RTCP und deren Anforderungen an die unterliegenden Protokolle, sowie der Frage, warum zum Zwecke der Audio- und Videoübertragung überhaupt ein spezielles Transportprotokoll nötig ist und nicht etwa TCP direkt dazu verwendet werden kann. Die "Entwurfsentscheidungen" sollen einen Überblick darüber geben, an welchen Forderungen sich die Entwickler der beiden Protokolle orientieren bzw. einige im Zusammenhang damit speziell auftretenden Begriffe erklären. Anhand eines Benutzungsszenarios wird ein Beispiel für eine RTP-Anwendung gegeben und abschließend einige wichtige Details von RTP und RTCP herausgestellt.

2. Übersicht

2.1 Eigenschaften von RTP und RTCP

RTP regelt den Austausch von Nutzdaten, ist jedoch nicht von vorneherein auf bestimmte Anwendungen wie etwa Konferenzschaltungen festgelegt, sondern soll allgemein die Übertragung von Realzeitdaten unterstützen. Nach Absicht der Enwickler wird, trotz der Tatsache, daß es sich bei RTP um ein "Realzeitprotokoll" für interaktive Anwendungen handelt, auch die Speicherung von Realzeitdaten und die Übertragung aufgezeichneter Realzeitdaten möglich sein, in dem Sinne, daß bestimmte Felder im RTP-Protokollkopf, wie etwa der Zeitstempel, in diesem "off-line-Modus" anders als im "on-line-Modus" behandelt werden. RTCP überwacht einerseits die Qualität der Nutzdaten-Verteilung, damit evtl. entsprechend korrigiert werden kann, andererseits erhält jeder Teilnehmer regelmäßig Informationen über den "Zustand" der anderen Teilnehmer. Die Überwachung der Qualität ist aber nur im Sinne einer Anzeige gemeint, d.h. es obliegt der Anwendung diese Informationen zu verwerten, RTCP greift selbst nicht korrigierend in eine RTP-Übertragung ein. Der Entwurf von RTP zielt darauf ab, daß das Protokoll (mit eingeschränkter Funktionalität) auch ohne RTCP funktionstüchtig ist. Es ergibt sich folgende Aufgabenverteilung auf die beiden Protokolle:

RTP:

- Austausch von Realzeitdaten

RTCP:

- Überwachung von Qualitätsparametern
- Übermittlung von Teilnehmerinformation

RTP und RTCP sind unzuverlässige, verbindungslose Protokolle. In keinem der beiden Protokolle ist ein Mechanismus vorgesehen, der eine Auslieferung der Datenpakete innerhalb einer vorgegebenen Zeit sicherstellt oder mittels dem irgendwelche Dienstqualitäten garantiert werden können. Weiterhin wird weder eine reihenfolgegemäße Auslieferung der Pakete, noch deren Auslieferung überhaupt garantiert. Die Protokolle gehen auch nicht von einem in dieser Hinsicht zuverlässigen unterliegenden Netzwerk aus, das die Pakete reihenfolgegetreu ausliefert. RTP unterstützt jedoch Sequenz-Nummern, die es dem Endsystem erlauben die Paketfolge zu rekonstruieren, die beim Sender erzeugt wurde. RTP soll auf eine Vielzahl von Netzwerk- und Transportprotokollen aufsetzen können, wie z.B. IP, ST-II oder UDP. Abschließend sei ausdrücklich darauf hingewiesen, daß sich sowohl RTP als auch RTCP gegenwärtig noch in der Entwicklung befinden. Mit Veränderungen einiger der im folgenden besprochenen Details muß also gerechnet werden. Allerdings wird man wohl an der gezeigten Grundfuktionalität und den angestrebten Entwicklungszielen festhalten.

2.2 Anforderungen von RTP und RTCP

RTP benötigt in der Regel die Dienste eines Ende-zu-Ende-Transportprotokolls wie beispielsweise UDP, TCP, OSI TP1 bzw. TP4 o.ä. Das Transportprotokoll sollte speziell die folgenden Dienste zur Verfügung stellen:

- Ende-zu-Ende Auslieferung
- Segmentierung
- Multiplex-Mechanismus
- Multicast-Dienst (Gruppen-Benachrichtigung)

Wie oben schon angesprochen wird davon ausgegangen, daß das zugrundeliegende Netzwerk unzuverlässig ist, d.h. es wird erwartet, daß es Pakete verliert, verfälscht, willkürlich verzögert und bezüglich der Reihenfolge umordnet. Die Unterstützung eines Multicast-Mechanismus durch die Netzwerkschicht ist zwar wünschenswert, aber nicht unbedingt notwendig. Alternativ soll RTP auch als Transportprotokoll benutzt werden können, das direkt auf IP aufsetzt, wodurch der Paketkopf-Overhead reduziert und damit die Leistung erhöht wird. Dies könnte insbesondere dann attraktiv sein, wenn die von UDP bereitgestellten Dienste wie Prüfsummenbildung und Demultiplexing für Multicast-Realzeit-Konferenzanwendungen überhaupt nicht benötigt werden. Diese Frage ist noch Gegenstand weiterer Überlegungen. Eine *mögliche* Einordnung von RTP und RTCP in den Internet-Protokollturm wird in Abb. 1 gezeigt. Konferenzen, die über mehrere Medienströme (z.B. Audio- und Videodaten) geführt werden, sollen darüber hinaus durch ein (zuverlässiges) "Konferenz-Kontrollprotokoll" gesteuert werden (siehe Abb. 1), auf dessen Definition aber hier nicht näher eingegangen wird.

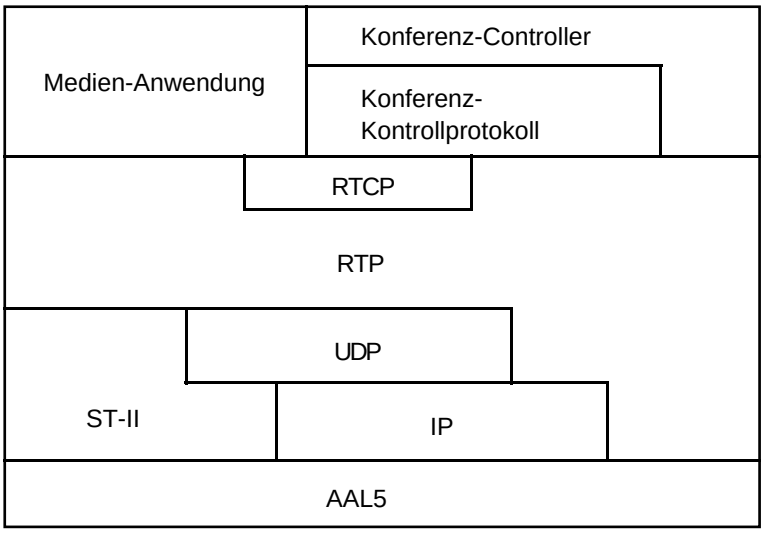


Abb. 1: Mögliche Einordnung von RTP und RTCP in den Internet-Protokollturm

2.3 Warum kann nicht TCP direkt zur Audio- und Videoübertragung verwendet werden?

Zwar ist die Verwendung von TCP für die Übertragung von Audio- und Videodaten zum Zwecke einer Überspielung grundsätzlich möglich, für eine Realzeitübertragung sind jedoch TCP und andere zuverlässige Transportprotokolle eher ungeeignet. Die drei Hauptgründe dafür sind:

- Realzeit-Audio- und Videodaten sind sehr verzögerungsempfindlich wofür eine zuverlässige und reihenfolgegetreue Auslieferung im allgemeinen nicht geeignet ist. Hat der Sender nämlich entdeckt, daß ein Paket nicht beim Empfänger angekommen ist und dieses erneut gesendet, dann ist mindestens die Dauer für eine erneute Übertragung des Pakets, wahrscheinlich aber noch mehr Zeit vergangen. Der Empfänger muß auf das fehlende Paket warten, was eine Verzögerung und somit eine wahrnehmbare Lücke bei der Abspielung verursacht.
- TCP unterstützt keinen Multicast-Dienst
- Der TCP Staukontroll-Mechanismus verkleinert das Flußkontroll-Fenster beim Erkennen von Paketverlusten ("slow start"). Audio- und Videodaten haben aber feste Übertragungsraten die nicht plötzlich verringert werden können. Beispielsweise wird bei einer Übertragung von in Standard-PCM codierten Audiodaten eine "feste" Übertragungsgeschwindigkeit von 64 kbit/s benötig (unter Vernachlässigung des Paketkopf-Overhead). Bei Videodaten ist eine Verringerung der Übertragungsrate zwar auch unerwünscht, doch resultiert diese dann (was weniger kritisch ist) in einer geringeren Bildfrequenz beim Empfänger.

Eine weiterer Nachteil ist, daß TCP-Paketköpfe größer sind als UDP-Paketköpfe. Auch enthalten zuverlässige Transportprotokolle nicht die notwendigen Zeitstempel und Codierungsinformation die von der empfangenden Anwendung benötigt werden, so daß sie RTP nicht ersetzen können. (Auf Sequenznummern könnte verzichtet werden, da diese Protokolle ja selbst garantieren, daß keine Paketverluste oder Umordnungen auftreten). Selbst der Einsatz von TCP zur Datenübertragung in einem LAN macht wenig Sinn. Denn LANs

besitzten in der Regel eine ausreichend große Bandbreite und eine ausreichend niedrige Paketverlustrate, so daß Paketverluste, -verfälschungen, -verzö-gerungen und -umordnungen die Übertragung nicht extrem beeinträchtigen. Aber auch sonst bringt der Einsatz von TCP in einer solchen Umgebung auch keinerlei Vorteile mit sich (mit der Ausnahme die sowieso schon niedrige Verlustrate auszugleichen). Im Gegenteil, denn selbst in einem LAN mit einer Paketverlustrate von Null würde sich der "slow start" bei jedem Übertragungsbeginn negativ auswirken.

3. Entwurfsentscheidungen

3.1 Ziele

Die AVT-Arbeitsgruppe ist bemüht sich bei der Entwicklung von RTP/RTCP an den nachfolgenden Zielen zu orientieren.

- Inhaltliche Flexibilität:
  Obwohl Audio- und Videoübertragungen primär das Protokolldesign motivieren, sollen dennoch durch die Vermeidung von zu speziellen Annahmen auch andere Anwendungen, wie z.B. verteilte Echtzeitsimulationen von den angebotenen Diensten profitieren können (d.h., von den eigentlichen Nutzdaten soll weitestgehend abstrahiert werden). Man beachte die Möglichkeit, das gleiche Datenfeld in einem Paketkopf abhängig vom Inhalt des Pakets auf unterschiedliche Weise zu interpretieren (beispielsweise könnte ein Synchronisierungsbit sowohl den Beginn eines Teilstücks Audiodaten als auch den Beginn eines Videoframes anzeigen). Ausserdem muß bereits die Möglichkeit neuer Datenformate wie z.B. solche zur Darstellung von hochqualitativen Mehrkanal-Audiodaten oder von kombinierten Audio- und Videodaten berücksichtigt werden.

- Erweiterbarkeit:
  Momentan sind Entwickler und Implementierer erst im Begriff sich mit Realzeit-Multimedia-Diensten wie etwa Videokonferenzen zu beschäftigen. Folglich sollte es auch in Zukunft noch möglich sein weitere Dienste in die Protokolle aufzunehmen, falls nämlich der Kreis der Anwendungen um solche erweitert wird, die ursprünglich nicht berücksichtigt wurden. Die Erweiterbarkeit wird auch die Bemühungen um eine Standardisierung beschleunigen, da die von einer Anwendergruppe erwarteten Anforderungen an die Protokollfunktionalität zu einem späteren Zeitpunkt hinzugefügt werden kann.

- Unabhängigkeit von unterliegenden Protokollen:
  RTP sollte sowenig Annahmen über die zugrundeliegenden Transport- und Netz-werkprotokolle machen wie möglich. Es sollte möglichst gut mit UDP, TCP, ST-II, OSI TP, VMTP sowie verschiedenen experimentellen Protokollen zusammenarbeiten, auch solchen, die eine Ressourcenreservierung und die Garantie von Dienstqualitäten unterstützen. Natürlich sind nicht alle Transportprotokolle gleichermaßen zur Erbringung von Realzeit-Diensten geeignet, wie oben erläutert, könnte insbesondere TCP unakzeptable Verzögerungen verursachen. Es ist noch eine offene Frage, ob RTP

die evtl. von unterliegenden Protokollen angebotenen Dienste (wie etwa Zeitstempel oder Sequenznummern) für seine eigenen Zwecke verwenden sollte.

- Mixer- und Übersetzer-Kompatibilität:
  RTP und RTCP sollten möglichts gut mit sog. Mixern und Übersetzern zusammenarbeiten, da Praxiserfahrungen gezeigt haben, daß diese aus mehreren Gründen nötig oder zumindest wünschenswert sind (siehe Abschnitt 3.2).

- Effizienz bezüglich der Übertragung:
  Gewöhnlich wird RTP wohl in relativ leistungsfähigen Netzwerken zum Einsatz kommen. Ungeachtet der Tatsache, daß Netzwerke immer größere Übertragungsraten bieten, spielt dennoch eine gute Übertragungseffizienz weiterhin eine große Rolle, beispielsweise bei dem Betrieb über ein (Hochgeschwindigkeits-) Modem oder eine internationale Verbindung. Um Ende-zu-Ende-Verzögerungen sowie die Auswirkungen von Paketverlusten zu minimieren, sollten die Pakete möglichst klein gehalten werden, etwa durch kurze Packungsintervalle und einer effizienten Codierung. Allerdings entwickelt sich bei einer größeren Anzahl kleinerer Pakete der zusätzliche Verarbeitungsaufwand pro Paket zu einem neuen Problem. Beim Beispiel einer Audioübertragung werden Pakete, die etwa 16 bis 32 ms an Sprachinformation enthalten als optimal angesehen. Typischerweise sind Pakete mit Videodaten um einiges größer, so daß dann der Overhead an Protokollinformation keine so große Rolle spielt. Die Übertragungseffizienz kann erhöht werden, indem nicht unbedingt benötigte oder sich nur langsam ändernde Protokollzustandsdaten in optionalen Feldern des Paketkopfes oder über ein separates Protokoll mit einer geringeren Übertragungsrate ausgetauscht werden. Auch eine Kompression des Paketkopfes kann in Erwägung gezogen werden.

- Internationalität:
  Vor dem Hintergrund eines weltweiten Einsatzes scheint angebracht bereits bei der Protokollentwicklung auf bestimmte nationale Besonderheiten zu achten, wie z.B. die Verwendung verschiedener Zeichensätze bei der Übertragung von Textdaten in einem RTCP-Paket.

- Bearbeitungseffizienz:
  Mit Ankunftsraten in der Größenordnung von 40 bis 50 Paketen pro Sekunde von einer einzigen Sprach- oder Videoquelle, kommt dem Overhead für die Bearbeitung eines Pakets eine besondere Bedeutung zu, gerade dann wenn das Protokoll nicht auf einer High-End-Workstation implementiert wurde. Deshalb sollten, wo immer möglich, bei der Implementierung aufwendige Rechenoperationen vermieden werden.

- Implementierbarkeit:
  Das Protokoll sollte mit heutiger Hardware und Betriebssystemen implementierbar sein. Dies schließt natürlich nicht aus, daß zukünftige Hardware oder Betriebssysteme die Leistungsfähigkeit des Protokolls erhöhen.

3.2 Synchronisierungsquellen und Inhaltsquellen, Mixer und Übersetzer

Synchronisierungsquellen

Alle Pakete ein und derselben Synchronisierungsquelle sind Teil des gleichen Sequenznummernraums und besitzen das gleiche Timing. Beispiele von Synchronisationsquellen sind ein Mikrophon, ein Mixer (siehe unten) oder eine Kamera. Der Empfänger gruppiert die ankommenden Pakete zum Zwecke der Wiedergabe nach Synchronisierungsquellen. Typischerweise versendet eine einzelne Synchronisierungsquelle nur einen einzigen Medienstrom (z.B. Audio- oder Videodaten). Eine Synchronisierungsquelle kann zwischenzeitlich ihr Datenformat z.B. die Audiocodierung ändern. Synchronisierungs-quellen werden durch den SSRC-Bezeichner (synchronization source identifier, SSRC identifier), einen numerischen Wert im RTP-Paketkopf (siehe Abb. 4) identifiziert. Der SSRC-Bezeichner ist eine per Zufall bestimmte 32 Bit-Größe, die global eindeutig sein sollte. Erkennt eine Quelle, daß eine andere Quelle den gleichen SSRC-Bezeichner benutzt, so wählt sie wiederum per Zufall einen neuen Bezeichner. Hat sie aber bereits Pakete mit dem nicht eindeutigen Bezeichner verschickt, dann sendet sie ein BYE-Kontrollpaket mit dem alten SSRC-Bezeichner bevor sie auf den neuen umschaltet. Die Wahrscheinlichkeit einer solchen Mehrdeutigkeit ist jedoch sehr gering.

Inhaltsquellen

Eine Inhaltsquelle ist die ursprüngliche Quelle der Daten, die in einem RTP-Paket befördert werden, beispielsweise also die Anwendung, die ein Teilstück von Audiodaten erzeugt hat. Inhaltsquellen werden durch den Inhaltsquellen-Bezeichner (contributing source identifier, CSRC identifier) identifiziert. Daten von einer oder mehreren Inhaltsquellen können zur Effizienzsteigerung durch einen Mixer zu einem einzigen RTP-Paket kombiniert werden, in diesem Fall wird der Mixer zur Synchronisierungsquelle. Die Inhaltsquellen werden dann in die CSRC-Liste eingetragen und identifizieren die logische Quellen der Daten. In Abb. 3 sind E1 und E2 die Inhaltsquellen der von E7 (über den Mixer M1) empfangenen Daten, wobei M1 die Synchronisierungsquelle ist.

Mixer:

Ein Mixer empfängt RTP-Pakete von einer oder mehreren Quellen, ändert möglicherweise deren Datenformat, kombiniert sie in einer bestimmten Art und sendet sie als neues RTP-Paket weiter. Man betrachte folgendes Beispiel: einige Teilnehmer einer Audiokonferenz sind über eine weniger leistungsfähigere Verbindung mit der Mehrheit der Konferenzteilnehmer verbunden, die wiederum über ein Hochgeschwindigkeitsnetz kommunizieren. Anstatt nun alle Teilnehmer dazu zu zwingen eine qualitativ niederwertigere Audiocodierung zu verwenden, wird nahe des angesprochenen Gebiets ein Mixer installiert. Dieser Mixer resynchronisiert die ankommenden Audiopakete verschiedener Quellen, "mixt" (überlagert) die Nutzdaten der entsprechenden Pakete, übersetzt die Audiocodierung in eine solche geringerer Qualität und schickt die so entstandenen neuen Pakete an die Konferenzteilnehmer des weniger leistungsfähigen Netzes. Da der Mixer einen neuen (gemixten) Strom von Audiodaten erzeugt hat, ist er nun die Synchronisierungsquelle dieses Stroms. Damit die Anwendungen der Empfänger nachvollziehen können, welche Teilnehmer gerade sprechen, fügt der Mixer den Inhaltsquellen-Bezeichner jedes Sprechers in die sog. CSRC-Liste eines ausgehenden Pakets ein (siehe Abb. 4, "RTP-Datenpaket-Kopf"). Diese Inhaltsquellen-Bezeichner sind wiederum die Synchronisierungsquellen-Bezeichner (synchronization source identifiers, SSRC identifiers) der Quellen die zu dem gemixten Paket beigetragen haben. In

Abb. 2 ist die Arbeitsweise eines Mixers anhand von RTP-Datenpaketen erklärt und ein Beispiel ist durch die Mixer M1,..., M3 in Abb. 3 gegeben.

Übersetzer:

Als Übersetzer werden hier Gateways auf RTP-Ebene bezeichnet, die nicht die Aufgabe haben, Pakete von unterschiedlichen Quellen zu mixen. Übersetzer werden gebraucht, um Anpassungen zwischen verschiedenen Transportprotokollen vorzunehmen (z.B., wenn eine Gruppe von Endsystemen, die mit IP/UDP arbeitet mit einer anderen Gruppe kommuniziert, die mit ST-II arbeitet), bzw. wenn die paketweise Übersetzung der Codierung einzelner Quellen durchgeführt werden muß. Ein Übersetzer empfängt RTP-Pakete und sendet sie weiter ohne ihre Synchronisierungsquelle zu ändern. In Abb. 3 sind die Endsysteme T1 und T2 Übersetzer (Translators). Beispiele für Übersetzer sind Geräte, die Codierungen umwandeln, ohne die Daten verschiedener Pakete zu mixen oder solche die Konvertierungen von Multicast zu Unicast vornehmen.

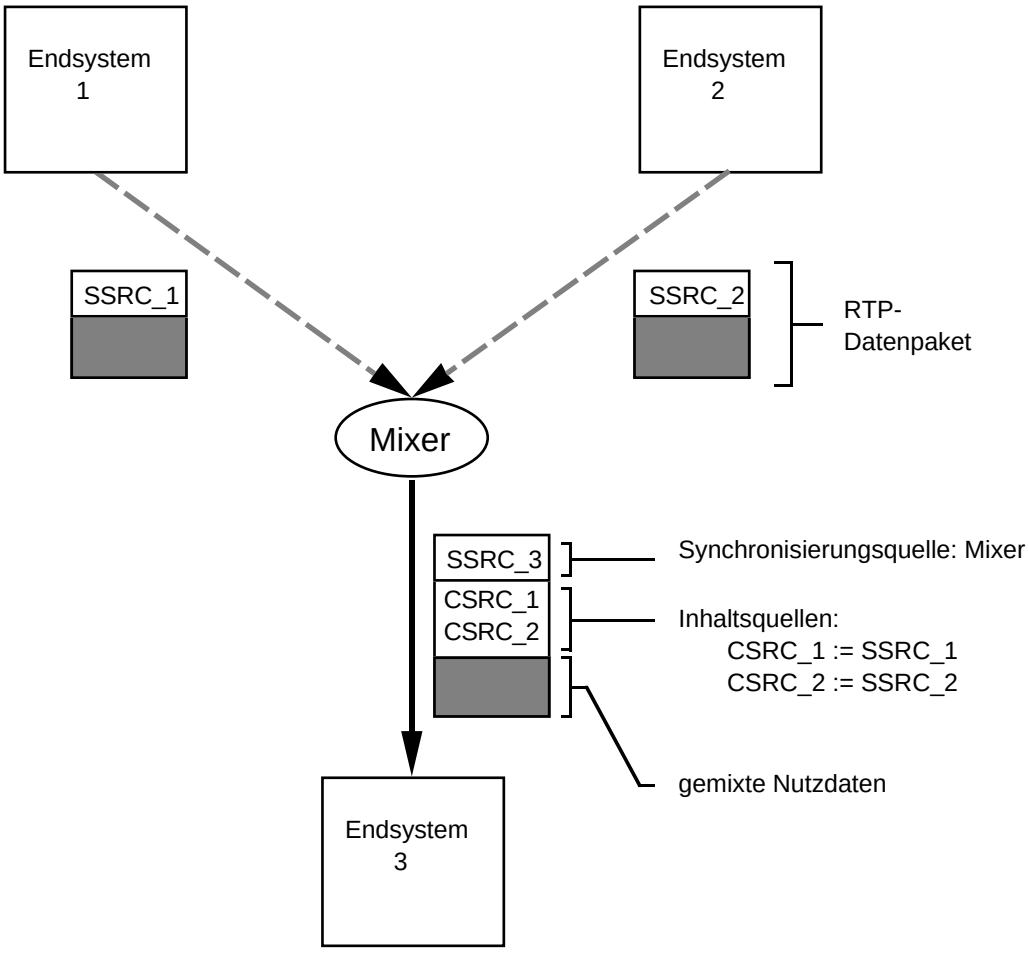


Abb. 2: Arbeitsweise eines Mixers anhand von RTP-Datenpaketen

3.3 Duplex oder simplex?

Protokolle können von der Richtung des Informationsflusses her gesehen in drei Kategorien eingeteilt werden:

- Für jede Protokollinstanz wandern Pakete nur in eine Richtung.
  D.h. der Empfänger hat keine direkte Möglichkeit den Sender zu beeinflussen. UDP ist ein Beispiel eines solchen Protokolls.

- Datenpakete werden nur in eine Richtung übertragen, Empfänger antworten mit Kontrollpaketen. ST-II in seinem Standard-Simplex-Modus ist ein Beispiel.

- Das Protokoll ist vollständig symmetrisch.
  Sowohl Benutzerdaten als auch Kontrollinformation wandern in beide Richtungen. TCP ist ein symmetrisches Protokoll.

Man beachte, daß ein bidirektionaler Datenfluß gewöhnlich durch zwei oder mehrere unidirektionale Datenflüsse mit entgegengesetzten Richtungen simuliert werden kann. Muß ein Empfänger Kontrollinformation zur Quelle übermitteln, so wird allerdings ein entkoppelter Strom in die entgegengesetzte Richtung allein nicht ausreichen, denn es muß zusätzlich Zustandsinformation aktualisiert werden. Für die meisten Anwendungen, die ein Realzeit-Transportprotokoll benutzen, wird wohl ein unidirektionaler Datenfluß ausreichend sein. Bidirektionale Datenflüsse sind bei 1:n-Beziehungen wie sie in Konferenzen auftreten ohnehin schwierig zu realisieren. Zudem ist es durch die Verzögerungszeiten des Netzwerks und die harten Echtzeitanforderungen wohl unmöglich eine zuverlässige Echtzeit-Datenübertragung durch die wiederholte Anforderung von verlorengegangenen oder verfälschten Datenpaketen zu erzielen, was ein weiteres Argument gegen einen bidirektionalen Kanal darstellt. Aus diesem Grund entschied sich die AVT-Arbeitsgruppe für einen unidirektionalen Datenfluß und je einen Kontrollfluß von den Empfängern der Daten zum Sender. Es treten innerhalb einer Multimedia-Konferenz mindestens zwei Fälle auf, bei denen ein Empfänger Kontrollinformation zurück zum Sender übermitteln muß. Einerseits möchte oder muß der Sender wissen, wie gut die Übertragung stattfindet, da aus Effizienzgründen nicht wie sonst üblich Quittungen zur Bestätigung eingesetzt werden. Andererseits sollte es dem Empfänger möglich sein, fehlende Zustandsinformation anzufordern. Wie kann aber nun ein Sender die Kontrollpakete (RTCP-Pakete) seiner Empfänger von Datenpaketen eines anderen Senders unterscheiden? Folgende drei Möglichkeiten bieten sich an:

- Kontrollport und Senderport sind gleich, markierte Pakete:
  Der gleiche Port wird sowohl für Daten- als auch für Kontrollpakete benutzt. Damit das Endsystem die Pakete auseinander halten kann, müssen diese entsprechend markiert werden. Da evtl. verschiedene Konferenzen zwar unterschiedliche Multicast-Adressen aber den gleichen Port verwenden, muß ein Empfänger, der ein RTCP-Paket zum Sender schicken will zusätztlich die Multicast-Adresse angeben. Mit einem weiteren Bezeichner muß dann noch der Datenstrom identifiziert werden, zu dem das Paket gehört. Diese Methode hat den großen Nachteil, daß jede Anwendung alle an die entsprechende Multicast-Adresse gesendeten Pakete zuerst empfangen und dann evtl. ignorieren muß, falls sie nicht der korrekte Empänger ist.

- Ein separater Kontrollport:
  Alle Empfänger nutzen für die RTCP-Pakete zum Sender dessen speziellen Kontrollport, der aber vom Datenport verschieden ist. Da der Pakettyp nun durch die Portnummer identifiziert werden kann, müssen nur noch die Multicast-Adresse und der

Datenstrom-Identifikator angegeben werden. Nachteil hierbei ist die Notwendigkeit eines weiteren Ports und der zusätztliche Aufwand für das Demultiplexen der beiden Ports.

- Unterschiedliche Ports für jeden Kontrollstrom:
  Dazu muß jede Quelle jedem Empfänger bekanntgeben über welchen Port sie Kontrollpakete empfangen will, auch solchen die evtl. erst zu einem späteren Zeitpunkt in eine Konferenz eintreten. Ein Vorteil ist die Möglichkeit der Anwendung nur bestimmte Kontrollströme beachten zu können.

Die AVT entschied sich mit der zweiten Methode - also mit der Verwendung je eines Ports für Daten- und Kontrollpakete - für einen Mittelweg.

3.4 Dienste

Die Dienste, die von RTP angeboten werden sollen, werden im folgenden aufgelistet. Nicht alle Dienste werden als zwingend erforderlich angesehen, die welche optional angeboten werden könnten, sind mit einem Stern gekennzeichnet.

- Segmentierung (*)
- Demultiplex-Mechanismus
- Ermittlung der Mediencodierung
- Synchronisierung der Abspielung zwischen der Quelle und den Empfängern
- Fehlererkennung (*)
- Verschlüsselung (*)
- Qualitätsüberwachung der Dienste (*)

Die Zustandsinformation, die innerhalb einer Konferenz übermittelt werden soll kann grob unterschieden werden in Information, die sich mit jedem Datenpaket ändert und in solche, die für längere Zeit konstant bleibt. Zustandsinformation, die sich nicht mit jedem Paket ändert, kann auf verschiedene Arten transportiert werden:

- als fester Bestandteil des RTP-Paketkopfes:
  Diese Methode ermöglicht eine Decodierung am einfachsten und stellt die Zustandssynchronisierung zwischen Sender und Empfänger(n) sicher. Allerdings kann sie bzgl. der Übertragungsleistung ineffizient sein und zudem beschränkt sie die Menge der übermittelbaren Zustandsinformation.
- als Paketkopf-Option:
  Hier wird die Information nur dann übermittelt, wenn sie tatsächlich gebraucht wird. Dies erfordert erhöhten Rechenaufwand, sowohl beim Sender als auch beim Empfänger.
- innerhalb eines RTCP-Pakets:
  Betrachtet man diese Methode in Bezug auf den Rechenaufwand und die Bandbreite, so ist sie dem Verfahren mit Paketkopf-Optionen in etwa gleichwertig. Ein Mechanismus, der feststellt, ob eine bestimmte Option innerhalb des Datenstroms auftritt muß hier berücksichtigt werden.

- durch eine Gruppen-Benachrichtigung
  Information über eine angehende Konferenz könnte an die entsprechende Multicast-Adresse gesendet werden.
- durch eine Konferenzkontrolle:
  Die Zustandsinformation wird übertragen, wenn die Konferenz zustandekommt, und dann nur noch wenn die Information sich ändert.
- durch ein Konferenzverzeichnis:
  Bei dieser Variante hat jeder Teilnehmer Zugang zu einem Verzeichnis, das die Zustandsinformation über eine angehende Konferenz enthält. Die Änderung der Zustandsinformation während der Konferenz ist hier vermutlich schwieriger als mittels einer Konferenzkontrolle, weil die Teilnehmer darauf hingewiesen werden müssen, daß sie in dem Verzeichnis nach veränderter Information Ausschau halten müssen. Deshalb ist ein Verzeichnis wahrscheinlich am besten für Information geeignet, die während der ganzen Konferenz konstant bleibt, z.B. die Multicast-Adresse, eine Liste der Mediencodierungen oder Titel und Organisator der Konferenz.

3.5 RTP-Profile

RTP soll für eine Vielzahl von Anwendungen eingesetzt werden können, die alle etwas unterschiedliche Anforderungen stellen. Die Flexibilität sich diesen Anforderungen anzupassen, soll erreicht werden durch mehrere Wahlmöglichkeiten in der Hauptprotokollspezifikation. Diese erlauben dann gewisse Parameter explizit durch sog. Profile zu setzen und damit das Protokoll auf eine bestimmte Anwendungsklasse und Umgebung "einzustellen". So gibt es beispielsweise für Audioübertragungen bereits spezielle Profile, mit denen etwa die Art der Codierung und die Interpretation bestimmter Felder im RTP-Paketkopf definiert oder die Verwendung eines bestimmten Netzwerk- oder Transportprotokolls festgelegt werden können. Weitere mögliche Verwendungen für Profile sind:

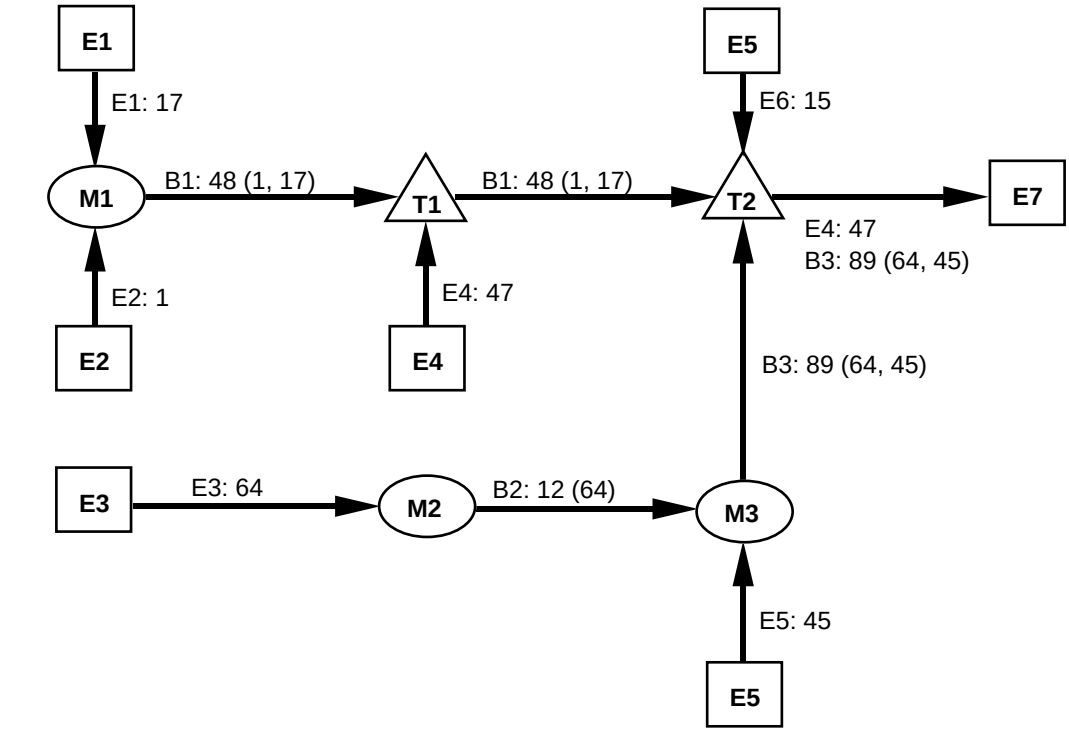
- Definition von neuen für eine Anwendungsklasse spezifische RTCP-Pakete oder Datenformate.
- Spezifikation einer Abbildung von RTP und RTCP auf die Transportschicht , z.B. auf UDP-Ports.
- Spezifikation der Kapselung von RTP-Paketen bezogen auf ein bestimmtes unterliegendes Protokoll.

Allerdings wird nicht erwartet, daß für jede Anwendung ein neues Profil benötigt wird. Innerhalb einer Anwendungsklasse wird es besser sein, ein bereits existierendes Profil zu erweitern als ein neues zu definieren.

4. Ein Benutzungsszenario

Das folgende Beispiel beschreibt einige Aspekte der Benutzung von RTP und soll die weiteren Erläuterungen motivieren. Es ist nur zur Illustrierung der Arbeitsweise des Pro-

tokolls gedacht und soll nicht als eine Beschränkung des Protokolls auf ein bestimmtes Anwendungsgebiet verstanden werden. RTP setzt hier auf IP und UDP auf.



Erklärung:

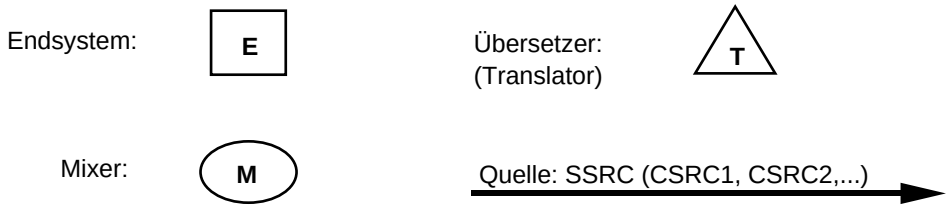


Abb. 3: RTP-Netzwerk mit Endsystemen, Mixern, und Übersetzern

4.1 Beispiel: Eine Multicast-Audiokonferenz

Eine Arbeitsgruppe möchte über ein aktuelles Thema diskutieren und nutzt dazu den IP-Multicast-Dienst für eine Sprachübertragung. Durch einen Reservierungsmechanismus werden dem Vorsitzenden der Arbeitsgruppe eine Multicast-Gruppenadresse und zwei Ports bereitgestellt. Ein Port ist für das Kontrollprotokoll (RTCP) vorgesehen, der andere für die Übertragung von Audiodaten. Diese Adreß- und Portinformation wird an die zukünftigen Teilnehmer versendet. Die genauen Details der Reservierungs- und Versendemechanismen gehören jedoch nicht zum Aufgabenbereich von RTP. Die Audio-Konferenzanwendung jedes Konferenzteilnehmers versendet Audiodaten in kleinen Teilstücken von ca. 20 ms Dauer. Jedes Teilstück wird durch einen RTP-Paketkopf angeführt, wobei Paketkopf und Daten wiederum in ein UDP-Paket eingebettet sind. Das Internet verliert in Überlastsituationen

Pakete, ebenso werden in der Regel Pakete bezüglich der Reihenfolge umgeordnet und um unterschiedliche Zeitdauern verzögert. Um diese Beeinträchtigungen ausgleichen zu können, enthält der RTP-Paketkopf Zeitinformation und eine Sequenznummer. Dadurch wird es den Empfängern ermöglicht, die korrekte Reihenfolge und Synchronisierung so wie sie beim Sender vorlagen zu rekonstruieren, damit also im vorliegenden Beispiel alle 20 ms ein Teilstück der Audiodaten an den Lautsprecher ausgeliefert wird. Die Sequenznummer kann von den Empfängern ebenfalls verwendet werden, um abzuschätzen, wieviele Pakete verlorengegangen sind. In jedem RTP-Paket ist Information darüber enthalten, welche Audio-Codierung (wie PCM, ADPCM oder GSM) verwendet wurde, so daß es einem Sender während der Konferenz möglich ist die Codierung zu wechseln, um sich etwa einem neuen Teilnehmer anzupassen, der über ein weniger leistungsfähigeres Netz angebunden ist. Da Mitglieder der Arbeitsgruppe während der Konferenz zustoßen oder sich abmelden, ist es nützlich zu wissen, welches Mitglied zu einem gegebenen Zeitpunkt gerade an der Konfererenz teilnimmt und wie gut es die Audiodaten empfängt. Zu diesem Zweck versendet jede Audioanwendung in der Konferenz periodisch per Multicast den Namen ihres Benutzers inklusive eines Empfangsberichts über den RTCP-(Kontroll-) Port. Die Email-Adresse und andere Benutzerinformationen können ebenfalls eingeschlossen werden. Eine Partei sendet eine RTCP-BYE-Nachricht, wenn sie die Konferenz verläßt. Der RTCP-Empfangsbericht zeigt an, wie gut der aktuelle Sprecher empfangen wird.

5. RTP und RTCP

5.1 Das RTP-Paketformat

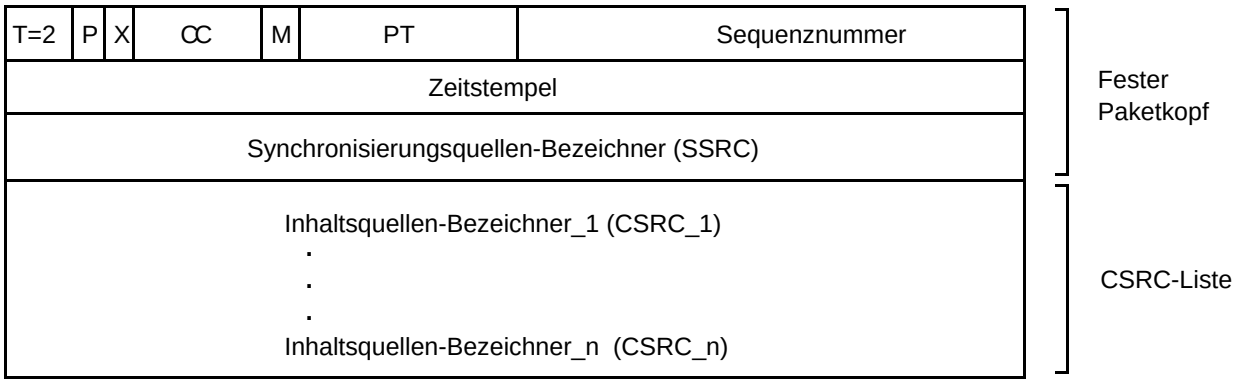


Abb. 4: RTP-Datenpaket-Kopf

Ein RTP-Paket besteht aus dem festen RTP-Paketkopf, einer möglicherweise leeren Liste von Inhaltsquellen (CSRC-Liste) und den Nutzdaten, falls vorhanden. Dabei sind RTP-Nutzdaten die Daten, die auf den festen RTP-Paketkopf und die CSRC-Liste folgen (siehe Abb. 4). Das Format und die Interpretation der Nutzdaten sollen hier nicht betrachtet werden. Beispiele von Nutzdaten sind Audio- und Videodaten. Die ersten Zwölf Oktette des in Abb.4 gezeigten Paketkopfes sind in jedem RTP-Paket enthalten (sie bilden den festen RTP Paketkopf), wobei die Liste der CSRC-Bezeichner nur dann vorhanden ist, falls sie von einem Mixer eingefügt wurde.

Die Felder haben die folgende Bedeutung:

type (T): 2 Bits
Gibt den Typ des RTP-Pakets an. Der Typ des oben gezeigten Pakets ist zwei (2). Dieser Wert wurde gewählt, um die Pakete von denen einer früheren RTP-Version zu unterscheiden.

padding (P): 1 Bit
Falls dieses Bit den Wert 1 besitzt, beinhaltet das Paket am Ende ein oder mehrere Oktette, die nicht Teil der Nutzdaten sind. Das letzte Oktett des Pakets ist dann die Zahl der Oktette, die ignoriert werden sollen. Dieses Padding (Polsterung) kann benötigt werden, falls ein Verschlüsselungsalgorithmus feste Blockgrößen erwartet, oder falls mehrere RTP-Pakete in einer Dateneinheit des unterliegenden Protokolls transportiert werden sollen.

extension (X): 1 Bit
Dieses Bit zeigt an, daß dem festen Paketkopf genau eine Paketkopf-Erweiterung eines bestimmten Formats folgt. Es wird zwar davon ausgegangen, daß der in Abb. 4 gezeigte feste Paketkopf für die üblichen Anwendungen ausreichend ist. Mit der Erweiterung des Paketkopfes wird aber die Möglichkeit offen gehalten weitere Funktionalität hinzuzufügen ohne gleich eine neue Version von RTP veröffentlichen zu müssen. Ausserdem kann eine Erweiterung auch zu Testzwecken nützlich sein.

CSRC count (CC): 4 Bits
Das CC-Feld enthält die Anzahl der CSRC-Bezeichner die auf den festen Paketkopf folgen.

marker (M): 1 Bit
Die Interpretation dieses Feldes wird durch ein Profil spezifiziert.

payload type (PT): 7 Bits
Dieses Feld bestimmt das Format der Nutzdaten und ihre Interpretation durch die Anwendung.

Sequenznummer: 16 Bits
Die Sequenznummer zählt RTP-Pakete, sie wird mit jedem gesendeten Paket um eins erhöht.

Zeitstempel: 32 Bits
Der Zeitstempel wird mit einer Taktrate erhöht, die durch das Format der Nutzdaten bestimmt ist. Beispielsweise wird bei Audiodaten der Zeitstempel nach jedem Sampling um eins erhöht.

SSRC: 32 Bit
Dies ist der Synchronisationsquellen-Bezeichner, er wird per Zufall gewählt und sollte innerhalb einer Sitzung eindeutig sein.

CSRC: bis zu 15 Einträge á 32 Bits
CSRC ist die Abkürzung für Inhaltsquellen-Bezeichner. Die Gesamtzahl aller Bezeichner, die in der CSRC-Liste stehen ist im Feld CC (CSRC count, siehe oben) eingetragen. Es können in der Liste maximal 15 Inhaltsquellen angegeben werden. Z.B. werden bei Audio-Paketen alle Inhaltsquellen, deren Pakete zusammengemixt wurden, um ein neues Paket zu erzeugen in die Inhaltsquellen-Liste des neuen Pakets eingetragen, um dem Empfänger die Möglichkeit zu geben, die Sprecher zu identifizieren. Ein CC-Wert von 0 zeigt an, daß die Synchronisierungsquelle die Inhaltsquelle ist. Falls CC ungleich 0 ist, so ist die Synchronisierungsquelle ein Mixer und nicht die Inhaltsquelle. Ein Mixer, der gleichzeitig

eine Inhaltsquelle ist, muß sich selbst in die CSRC-Liste für dieses Paket eintragen. Die CSRC-Liste wird durch Übersetzer nicht verändert.

5.2 Die RTCP-Pakete

Wie schon angesprochen stellt RTCP zwei Funktionen zur Verfügung:

- Überwachung von Qualitätsparametern
- Übermittlung von Teilnehmerinformation

Für die erste Funktion sind die Sender- bzw. Receiver-Report Pakete zuständig, die unten beschrieben werden. Die zweite Funktion unterstützt lose kontrollierte Sitzungen, das sind solche bei denen Teilnehmer während der Sitzung ohne explizite Kontrolle und Parameterverhandlungen einer Konferenz beitreten oder eine Konferenz verlassen. RTCP Pakete werden an alle Teilnehmer einer Sitzung verschickt und zwar mit dem gleichen Verteilermechanismus der auch für Datenpakete verwendet wird. Das unterliegende Protokoll muß für die Daten- und Kontrollpakete einen Multiplexmechanismus unterstützen (Beispiel: Verwendung von verschiedenen Ports in UDP). Es soll an dieser Stelle keine detaillierte Beschreibung der einzelnen RTCP-Paketformate gegeben werden, stattdessen folgt eine Übersicht über die in RTCP vorgesehenen Pakettypen.

- SR: Sender Report (Senderbericht)
  Dieses Paket wird von einer Quelle verschickt, die vor kurzem RTP-Daten gesendet hat. Das Sender Report-Paket besteht aus zwei Teilen. Der erste Teil, der eigentliche Senderbericht, ist 24 Oktette lang und in jedem Sender Report-Paket enthalten. Der zweite Teil beinhaltet null oder mehr Empfangsberichte, abhängig von der Anzahl der Quellen von denen seit dem letzten Senderbericht RTP-Pakete empfangen wurden. Jeder dieser Empfangsberichte enthält Statistiken über den Empfang von RTP-Paketen bezogen auf eine einzelne Synchronisierungsquelle.
- RR: Receiver Report
  Das Receiver Report-Paket wird nur dann von einer Quelle anstatt eines SR-Pakets verschickt, wenn die Anwendung nicht vor kurzem RTP-Daten gesendet hat. Der Aufbau eines RR-Pakets ist ähnlich dem eines SR-Pakets.
- SDES: Source description
  Das SDES-Paket besteht aus einem Paketkopf und null oder mehr Attributen zur Beschreibung von Quellen. Folgende Attribute zur Quellenbeschreibung sind derzeit vorgesehen (weitere können durch Profile definiert werden):
  - CNAME:
    User- und Domain-Name.
  - NAME:
    Zeichenkette zur Bezeichnung der Quelle (z.B. Name, Titel, etc.)
  - EMAIL:
    Die E-Mail Adresse des Benutzers
  - PHONE:
    Telefonnummer des Benutzers.

    - LOC:
      Z.B. Land, Ort, Straße, Hausnummer, usw.
    - TXT:
      Eine Nachricht, die den momentanen Status der Quelle beschreibt (z.B. "kann momentan nicht sprechen, bin beim Essen").
- TOOL:
  Eine Zeichenkette, die den Namen und evtl. die Version der Anwendung angibt (kann bei der Fehlersuche nützlich sein).
- PRIV: Private extensions
  Dieser Typ dient dazu experimentelle oder anwendungsspezifische SDES-Erweiterungen zu definieren.
- BYE: Goodbye
  Eine Anwendung sendet ein BYE-Kontrollpaket, wenn sie die Konferenz verläßt.
- APP: Application-defined
  Dient zu Experimentierungszwecken.

Zusammenfassung

Abschließend läßt sich wohl sagen, daß mit RTP und RTCP eine Lücke in der Familie der Internetprotokolle geschlossen wird. Als positiv hervorzuheben ist dabei deren weitreichende Verwendbarkeit innerhalb der Klasse der Realzeitanwendungen, was auf eine breite Akzeptanz hoffen läßt. Jedoch können Anwendungen die Dienstleistungsgarantien erwarten nicht von diesen Protokollen profitieren. Zudem bleibt abzuwarten inwieweit das Internet in Zukunft dem ständig steigenden Datenaufkommen gewachsen ist.

Literatur

/RTP 1/ Schulzrinne; Casner; Frederick; Jacobson
RTP: A Transport Protocol for Real-Time Applications
Internet Engineering Task Force
Internet-Draft
draft-ietf-avt-rtp-06.txt

/RTP 2/ Schulzrinne, Henning;
Issues in Designing a Transport Protocol for Audio and Video Conferences and other Multiparticipant Real-Time Applications
Internet Engineering Task Force
Internet-Draft
draft-ietf-avt-issues-01.txt

/RTP 3/ Turletti, Thierry; Huitema, Christian;
Packetization of H.261 video streams
Internet Engineering Task Force
Internet-Draft
draft-ietf-avt-video-packet-03.txt

/RTP 4/ Speer, Michael F.; Hoffman, Don
RTP Encapsulation of CellB Video Encoding
Internet Engineering Task Force
Internet-Draft
draft-ietf-avt-cellb-profile-01.txt

/RTP 5/ Casner, Steve;
Minutes of the Audio/Video Transport Working Group (AVT)
(30.08.94)

Der QoS-Broker — Handel mit Dienstgüte?

Jochen Mayer

Multimedia-Anwendungen bilden durch ihre vielfältigen Ansprüche an den Kommunikationsdienst (z.B. gleichzeitige Übertragung mehrerer Datenströme mit unterschiedlichen Anforderungen an die Dienstqualität) eine Herausforderung für heutige Kommunikationssysteme. Sie verlangen nicht nur fortgeschrittene Dienstmodelle mit Fähigkeiten zur Aushandlung von Qualitätsparametern, sondern auch Mechanismen innerhalb der Kommunikationssysteme, die Dienstqualitäten unterstützen.
In diesem Beitrag wird ein an der University of Pennsylvania entwickeltes Rahmenwerk zur Unterstützung von Dienstqualitäten vorgestellt. Das Konzept beruht auf einem QoS-Broker, der das Verbindungsglied zwischen Anwendungen einerseits und Kommunikationssystem und Betriebssystem andererseits bildet.

1 Einleitung und Motivation

Multimedia-Anwendungen zeichnen sich durch die Übertragung unterschiedlicher Medien wie Audio- und Videodaten aus. Dabei müssen große Datenmengen auf unterschiedliche Art und Weise verarbeitet werden. Um die Übertragung in einer sinnvollen Weise, die der Anwendung angepaßt ist, zu gewährleisten, sind Forderungen nach sehr unterschiedlichen Dienstqualitäten mit der Übertragung verbunden. Diese Dienstqualitäten umfassen quantitative Merkmale, wie Datendurchsatz und Übertragungszeiten, und qualitative Merkmale, wie Reihenfolgetreue, Fehlerkorrektur und Synchronisation verschiedener Datenströme. Die Beschreibung von Diensten bestimmter Güte erfolgt durch *QoS-Parameter* (Quality of Service-Parameter, Dienstgüteparameter).
Diese Dienste sollen von fortschrittlichen Hochgeschwindigkeitsnetzwerken erbracht werden, die in Hardware, beispielsweise auf Basis von Glasfaserkabeln, und Software, durch spezielle Hochgeschwindigkeitskommunikationssysteme, speziell an diese Anforderungen angepaßt sind. Man spricht auch von speziellen Multimedia-Netzwerken (MMN).
Beispiele für Hochgeschwindigkeits-Kommunikationssysteme, die die Grundlage für Multimedia-Kommunikation bieten, sind das Heidelberg Transport System *HeiTS* des IBM Networking Research Center Heidelberg und das Funktionale Kommunikationssubsystem *FuKSS* des Instituts für Telematik an der Universität Karlsruhe.
In herkömmlichen Netzwerksystemen war es bislang üblich, jedem Teilnehmer das gesamte augenblicklich verfügbare Potential zur Verfügung zu stellen und die Kommunikationsdienste so gut wie möglich zu erbringen (*best-effortt*).
MMNe stellen nun jedoch sehr hohe Potentiale an Ressourcen zur Erbringung von Diensten zur Verfügung. Es besteht die Gefahr, daß diese großen Potentiale bei unkontrolliertem Zugriff bereits durch wenige Teilnehmer ausgeschöpft werden, wodurch andere Teilnehmer ungerechtfertigt behindert werden. Dieses Problem wird als *misbehaved user* (unfairer Benutzer) bezeichnet.
Um vor Mißbrauch und Engpässen zu schützen werden die Netzwerkressourcen durch Ressourcenverwalter administriert. Diese besitzen Wissen über vorhandene und belegte Ressourcen und teilen sie den anfragenden Teilnehmern zu. Dadurch kann das Potential verteilt und insgesamt die Verfügbarkeit des Netzwerks erhöht werden. Die Teilnehmer wiederum erhalten vom Kommunikationssystem garantierte Dienste (*guaranteed services*), mit denen sie ihre Anwendungen vernünftig abwickeln können. Ressourcenverwalter wachen auch ständig über die Verfügbarkeit der Ressourcen und sorgen gegebenenfalls für eine Anpassung im laufenden Betrieb.
An der Ausführung zugesicherter Dienste ist aber nicht nur das Multimedia-Netzwerk beteiligt, sondern auch

- das Betriebssystem,
- das Multimedia-System, das Multimedia-Geräte wie Kameras und Mikrofone sowie entsprechende Hard- und Software zur Bedienung der Geräte zur Verfügung stellt, und
- der Multimedia-Dienstanbieter der Gegenstelle, der zum Beispiel Videobilder beschafft und überträgt.

Diese Dienstgeber müssen ebenfalls in der Lage sein Dienste in bestimmter Dienstgüte zu erbringen. Das Multimediasystem, das im allgemeinen in das Betriebssystem über Treiber integriert ist, muß in der Lage sein, die Mediadaten innerhalb der spezifizierten QoS-Parameter zu verarbeiten und die Gegenstelle muß den gewünschten Multimedia-Dienstanbieter sowie Betriebssystem-Dienste für dessen Betrieb zur Verfügung stellen.
Wenn Ressourcen nicht im gewünschten Umfang verfügbar sind ist es von Vorteil, wenn der ablehnende Partner ein alternatives Angebot macht. Die Anwendung kann dann entscheiden, ob sie mit den so geminderten QoS-Parametern betrieben werden kann oder ob der Betrieb

abgelehnt werden muß. Dies erfordert eine Verhandlungsphase zwischen den beteiligten Instanzen.
Zusammengefaßt bedeutet dies: um einer verteilten Multimedia-Anwendung garantierte Dienste zur Verfügung zu stellen, müssen QoS-Parameter zwischen verschiedenen Instanzen ausgehandelt werden und Ressourcen zur Erbringung dieser Parameter reserviert und aufrechterhalten werden.
Um die Kooperation von Anwendung, Betriebssystem und Kommunikationssystem, auch zwischen dem Dienstnehmer und dem Diensterbringer, zu ermöglichen ist es sinnvoll eine Vermittlungsinstanz zwischen diesen Instanzen zu integrieren. Eine solche Vermittlungsinstanz stellt der *QoS-Broker* (Dienstgütemakler) dar, der vom Distributed Systems Laboratory an der University of Pennsilvania entwickelt wurde.

2 Ressourcen einer verteilten Multimedia-Anwendung

Um eine Multimedia-Anwendung sinnvoll zu betreiben, müssen den einzelnen beteiligten Instanzen Ressourcen zugestanden werden, die sie benötigen um garantierte Dienste zu erbringen.
Beispiel Bildtelefon:
Eine Bildtelefonanwendung auf einer Arbeitsstation in einem Multimedia-Netzwerk benötigt Ressourcen, die von verschiedenen Instanzen in unterschiedlicher Qualität zur Verfügung gestellt werden. Das lokale Multimediasystem stellt spezielle Geräte, wie zum Beispiel Mikrofon, Lautsprecher und Kamera. Dies sind aktive Ressourcen, die exklusiv genutzt werden.
Das Betriebssystem auf der lokalen Maschine bietet Speicher, CPU-Zeit und ein grafisches Interface zur Darstellung der Videobilder an. Diese Ressourcen werden mit anderen Anwendungen des lokalen Systems geteilt. Ebenfalls als Ressource kann der Speichertransfer von den Multimedia-Geräten innerhalb einer bestimmten Zeit in den Hauptspeicher und von dort in das Kommunikationssystem betrachtet werden.
Auf der Gegenseite befindet sich der Multimedia-Dienstanbieter, in diesem Fall ebenfalls ebenfalls eine Bildtelefonanwendung, der die Daten der Gegenstelle in einer bestimmten Qualität zur Verfügung stellt und andererseits die Weiterverarbeitung der gesendeten Daten in festgelegten Qualitätsstufen garantiert. Die Dienstanbieter der Gegenstelle ist dabei in der gleichen Weise auf Ressourcen des Betriebssystems und des Kommunikationssystems angewiesen wie die initiierende Anwendung. Schließlich garantiert das Kommunikationssystem den Transport der Daten über eine Verbindung zur Gegenstelle, ebenfalls in einem definierten Zeit- und Qualitätsrahmen.
Wie aus dem Beispiel ersichtlich ist, werden der Anwendung von verschiedenen Instanzen des Netzwerks unterschiedliche Ressourcen und Dienste zur Verfügung gestellt, die sich in Qualität und Aussehen grundlegend unterscheiden.

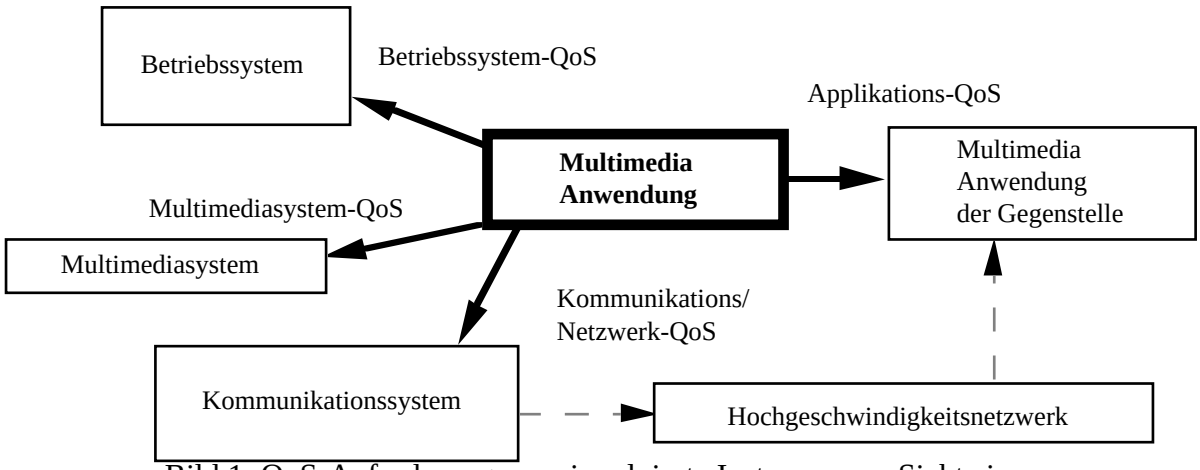


Bild 1: QoS-Anforderungen an involvierte Instanzen aus Sicht einer Multimedia-Anwendung

Um die Qualität der Dienste bestimmen zu können, werden von der initiierenden Anwendung QoS-Parameter festgelegt. Die QoS-Parameter definieren sich lokal im Kontext der zu erbringenden Leistung, bzw. der vorhandenen Ressourcen. Das heißt, jede Instanz hat ihre eigenen Definitionen von Quality-of-Service.
Für die folgende Arbeit wird vereinfacht das Multimediasystem als in das Betriebssystem integriert angenommen. Dies ist in der Realität durch die Integration mit Treibersoftware und Nutzung gemeinsamer Ressourcen wie CPU und Hauptspeicher gegeben.

2.2 Ressourcenverwaltung

Ressourcen sind Systemeinheiten, die benutzt werden um Daten zu bearbeiten. Aktive Ressourcen, beispielsweise die CPU oder ein Netzwerkadapter, bieten Dienste an, die beansprucht werden können. Passive Ressourcen, wie etwa Speicher oder Bandbreite, werden belegt und von aktiven Ressourcen zur Verarbeitung benötigt. Außerdem können Ressourcen exklusiv genutzt werden (eine Kamera) oder mit anderen Instanzen und Anwendungen geteilt werden (Speicher). Drittens können Ressourcen in einem System einfach oder mehrfach existieren. Auch Dienste, die zur Verfügung gestellt werden, können als Ressourcen betrachtet werden, die beliebig oft duplizierbar sind.
Ressourcen werden üblicherweise Teilsystemen zugeordnet, im Falle verteilter Multimedia-Anwendungen

(1) dem Betriebssystem (mit integriertem Multimediasystem),
(2) dem Kommunikationssystem (mit Netzwerk) und
(3) dem Dienstbringer der Gegenstelle.

Da das Multimediasystem über die Schnittstellen und die Treibersoftware in der Regel eng mit dem Betriebssystem verknüpft ist, wird es üblicherweise vereinfacht als Teil des Betriebssystems betrachtet.
Das Betriebssystem stellt als Ressourcen CPU-Zeit und CPU-Verfügbarkeit[1] . Unter Verfügbarkeit ist die Zusicherung an einen Prozeß zu verstehen, regelmäßig bearbeitet zu werden, z.B. in Form einer hohen Priorität. Eine weitere wichtige Betriebssystem-ressource ist der Speicher, einerseits in Form von Systemspeicher, der zum Puffern und zum Transport von Daten von und zu anderen Subsystemen zur Verfügung gestellt wird, wie auch als Anwendungsspeicher für die Anwendung. Für die Übertragung von Daten innerhalb

[1] CPU-Zeit als Merkmal stellt nur einen Gesamtbetrag an Zeit dar, die der Bearbeitung einer Anwendung zur Verfügung steht. Sie macht aber keine Aussage darüber, innerhalb welcher realen Zeitschranken die CPU einer Anwendung garantiert zur Verfügung gestellt wird.

bestimmter Zeitbedingungen ist ebenfalls die Speichertransferzeit[2] von Bedeutung. Hinzu kommen exklusiv genutzte Betriebsmittel wie Kamera, Mikrofon oder Lautsprecher.
Die Dienstgüteparameter des Kommunikationssystems schlagen sich beispielsweise in Datenpaketgröße, Paketlaufzeiten (*End-to-End-Delay*) oder garantierten Fehlertoleranzwerten nieder.
Eine verteilte Multimedia-Anwendung definiert eigene Ansprüche an Ressourcen der Subsysteme, die sich in den QoS-Parametern ausdrücken. Das Betriebssystem hat beispielsweise zu gewährleisten, daß die benötigten Mediadaten innerhalb definierter Zeitgrenzen in akzeptabler Qualität beschafft und weiterverarbeitet werden können. In diesen Bereich fallen das Einlesen von Daten und der Transport vom Systemspeicher in den Anwendungsspeicher, bzw. umgekehrt. Das Kommunikationssubsystem garantiert im Rahmen seiner Möglichkeit den mediengerechten Transport der Daten vom Sender zum Empfänger. Die Ressourcen des Netzwerks werden dabei hinter dem Kommunikationssubsystem versteckt.
Zusätzlich werden von der Applikation Dienstgütegarantien der Applikation der Gegenstelle verlangt, die im symmetrischen Fall die gleiche Applikation wie auf der Initiatorseite sein kann oder eine spezielle Serverapplikation für eine Client-Server-Anwendung.
Die Ressourcen werden im Allgemeinen während der Initialisierung der Anwendung von Ressourcenverwaltern reserviert. Die jeweiligen Ressourcenverwalter der Teilsysteme erhalten über definierte Schnittstellen die angeforderten QoS-Parameter mitgeteilt. Mittels eines Zugangskontrolltests wird die Verfügbarkeit der Ressourcen überprüft. Ist diese gegeben, werden die Ressourcen im nächsten Schritt für die Anwendung reserviert.
Sind die Ressourcen jedoch nicht im geforderten Umfang vorhanden, so besteht die Möglichkeit, innerhalb gewisser Grenzen, geringere oder alternative Ressourcen zur Verfügung zu stellen. Diese werden ebenfalls reserviert, der Anfragende jedoch zur Änderung seiner QoS-Parameter aufgefordert. Dieser kann dann entscheiden, ob die alternativen Ressourcen seinen Zwecken ausreichen.
Oft ist die Verwendung von Ressourcen nur dann hinlänglich sinnvoll, wenn mehrere beteiligte Teilsysteme, untereinander abgestimmt, Ressourcen stellen können. Zum Beispiel können auf einer Verbindungsstrecke innerhalb eines Multimedia-Netzwerks mehrere Ressourcenverwalter in einzelnen Verbindungsknoten involviert sein. Die Dienstgüte der gesamten Verbindung definiert sich somit nach der schwächsten Dienstgüte in der Verbindungsstrecke. Die übrigen Ressourcenverwalter müssen somit ebenfalls auf die schwächeren QoS-Parameter abgestimmt werden.

2.3 Ressourcenverwaltung im Netzwerk

Die Ressourcenverwaltung in einem MMN gliedert sich hauptsächlich in zwei Aufgaben, der Reservierung von Ressourcen während des Verbindungsaufbaus und der Zuteilung der Ressourcen während der Übertragung.
Bei den QoS-Verhandlungen werden die Ressourcenverwalter der beteiligten Knoten mittels eines Reservierungsprotokolls über die angeforderten QoS-Parameter benachrichtigt. Die übermittelten QoS-Parameter müssen in einem ersten Schritt auf die systemrelevanten Parameter und Ressourcenanforderungen umgerechnet werden. So muß der Ressourcenverwalter im Netzwerk aus den geforderten Paketübertragungszeiten die von ihm benötigten Parameter wie Pufferspeicherbedarf (*Queuing Buffer*)und Paketbehandlungsstrategie (*Packet Queuing*) errechnen. Dann wird mittels eines Zugangskontrolltests entschieden, inwiefern die QoS-Vorgaben erfüllt werden können. Ist der Test positiv, werden die Ressourcen reserviert und die Anfrage weitergeleitet. Falls der Test Werte ergibt, die noch im Rahmen der zulässigen QoS-Anforderungen liegen und eine Änderung der angeforderten QoS-Parameter zulässig ist, werden die Ressourcen ebenfalls reserviert und die veränderten

[2] Speichertransferzeit ist die Zeit, innerhalb derer Daten vom Systemspeicher in den Anwendungsspeicher und umgekehrt befördert werden.

QoS-Parameter weitergeleitet. Haben die nun geänderten QoS-Parameter den Endknoten erreicht, so werden sie quittiert zurückgesendet. Auf dem Rückweg werden die geänderten Parameter von den involvierten Knoten entgegengenommen, die nun ihre Ressourcen gegebenenfalls noch einmal anpassen. Diese Anpassung erfolgt einfacher, da die Anforderungen geringer sind, die reservierten Ressourcen jedoch schon den höheren Anforderungen genügen.
Bei diesem Vorgang sind mehrere Strategien möglich. Bei der bilateralen End-zu-End-Verhandlung können die angeforderten QoS-Parameter lediglich von der Gegenstelle, nicht aber von der Netzwerkschicht beeinflußt werden. Diese Verhandlungen finden zwischen der initiierenden und der Anwendung der Gegenstelle statt.
Hingegen sind bei der triangulären Verhandlungsstrategie (*Triangular Neotiation*) auch Änderungen durch die Netzwerkknoten erlaubt. In beiden Fällen müssen die Reservierungsparameter jedoch an den Initiator (*Caller*) zurückgeleitet werden, wobei die beteiligten Instanzen auf dem Rückweg über die endgültigen QoS-Parameter informiert werden.
Wenn die Änderung quantitativer Parameter durch beteiligte Instanzen erlaubt ist, so muß auch der Rahmen der Änderung definiert werden. Die angeforderten Parameter können zum Beispiel als akzeptabler Mittelwert definiert werden, der den Anforderungen gemäß angepaßt werden kann, oder aus einem angestrebten Wert und einem Grenzwert (*Bounded Target*), innerhalb derer die veränderten Werte sich befinden können.
Damit die beteiligten Instanzen Ressourcen[3] reservieren können, müssen diese zunächst auf Verfügbarkeit geprüft werden. Dazu werden Tabellen über vorhandene Ressourcen und Belegungstabellen benötigt. Ist die Verfügbarkeit gegeben, so werden die Ressourcen für die Übertragungsdauer anschließend belegt.
Falls sich die Situation im Netzwerk ändert, zum Beispiel durch ansteigende oder auch sinkende Last, können betroffene Knoten die Neuverhandlung der Verbindung verlangen, die möglicherweise zu einer Neuaushandlung der QoS-Parameter führen kann. Der Bedarf zur Neuverhandlung kann vorausplanend durch die Verteilung der Ressourcen ersehen werden oder durch eine erhöhte Fehlerquote festgestellt werden, je nachdem ob eine optimistische oder pessimistische Reservierungsstrategie angewendet wurde. Pessimistisch bedeutet, daß die Ressourcen entsprechend der maximalen Last reserviert werden, bei optimistischer Reservierung werden sie entsprechend der erwarteten mittleren Last vergeben.
Die Anfrage zur Neuverhandlung wird entlang der Verbindung verteilt und führt zum Anhalten der Übertragung, woraufhin mittels unterschiedlicher Strategien neue QoS-Parameter ausgehandelt werden können. Entsprechende Mechanismen zur Rekonfiguration sind in Hochgeschwindigkeits-Kommunikationssystemen vorgesehen.

3 Der QoS-Broker

Der QoS-Broker stellt die Vermittlungsinstanz zwischen den, an der mulitimedialen Netzwerkkommunikation beteiligten, Teilinstanzen dar. Seine Aufgaben können in drei Bereiche gegliedert werden.

(1) Am Endpunkt des Initiators handelt der QoS-Broker die durch Parameter bestimmten lokal benötigten Ressourcen des Betriebssystems und des Multimedia-Systems aus.
(2) Außerdem ist der QoS-Broker für die Verhandlung zwischen den Applikationen der Endpunkte (*Peer-to-Peer-Negotiation*), wobei je ein QoS-Broker in jedem Endpunkt an den Verhandlungen beteiligt ist.
(3) Schließlich werden noch die Verhandlungen zwischen den Schichten (*Layer-to-Layer-Negotiation*) in den Endpunkten vom QoS-Broker durchgeführt.

Diese Verhandlungsschritte werden in Abbildung 2 nochmals verdeutlicht.

[3] Prinzipiell ist es auch möglich, Ressourcen im Voraus reservieren zu lassen.

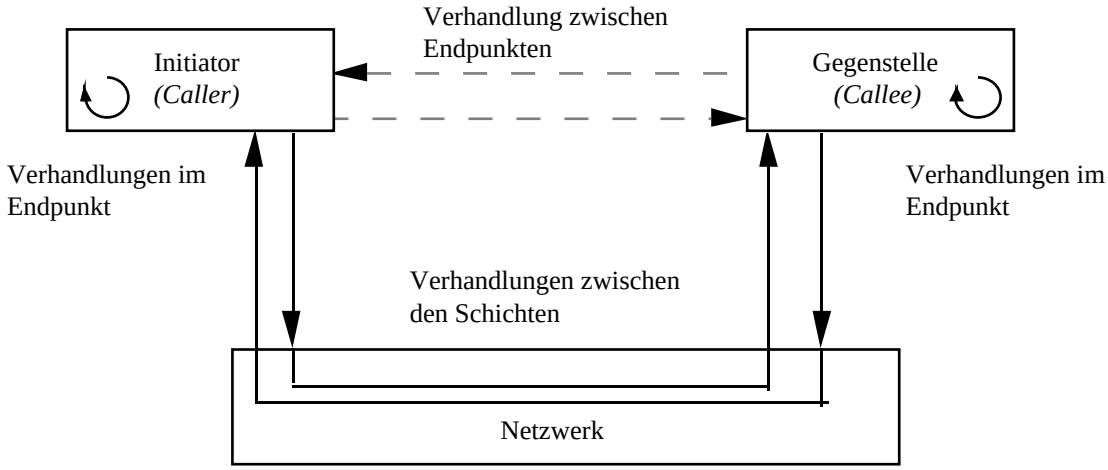


Bild 2: Verhandlungsschritte bei der QoS-Anforderung einer verteilten Multimedia-Anwendung

3.1 QoS-Verhandlung der Anwendungen

Aus Sicht der verteilten Multimedia-Anwendung sind bei der Initiierung der Multimedia-Übertragung mehrere Instanzen involviert. Zunächst fordert ein *Initiator* (*Caller*) von einer *Gegenstelle* (*Callee*) einen Dienst an, zum Beispiel die Übertragung eines Videoclips in einem Video-on-Demand-Service. Um nun die Form des so angeforderten Dienstes zu bestimmen findet eine Verhandlung auf gleicher Ebene statt (*Peer-to-Peer-Negotiation*). Die Aufgabe zwischen diesen beiden Instanzen zu vermitteln übernimmt der QoS-Broker. Auf Initiatorseite übernimmt er dazu die Rolle des *QoS-Buyer* (Dienstgütekäufer), der die Verhandlungen mit der Ermittlung der Vorgaben initiiert, während er auf der Seite des Angerufenen als *QoS-Seller* (Dienstgüteverkäufer) die Verfügbarkeit angeforderter Dienste anbietet.
Der QoS-Buyer ermittelt zunächst die Dienstgüteparameter, wie sie von der Anwendung verlangt werden. In der ersten Phase versucht er die Ressourcen zu beschaffen, die lokal im Endpunkt vom Betriebssystem benötigt werden (Verhandlungen im Endpunkt). Dazu bedient er sich einer *Zugangskontrolle* (*Admission Service*), die feststellt ob das Betriebssystem in der Lage ist, die benötigten Daten innerhalb der vorgegebenen Zeitschranken von den Multimedia-Geräten zu beschaffen und ob diese Daten innerhalb der maximalen Datenpaketlaufzeit aus dem Benutzerspeicher transferiert werden können. Schlagen die Verhandlungen fehl, wird versucht die QoS-Parameter mit der Applikation neu abzustimmen. Wenn die lokalen Ressourcen reserviert sind, werden die Verhandlungen mit dem QoS-Seller des Angerufenen aufgenommen. Antwortet der QoS-Seller mit "*accept*" oder "*modify*", werden die QoS-Parameter, sofern nötig, angepaßt und die QoS-Verhandlungen des Kommunikationssystems können beginnen.
Der QoS-Seller verfährt dabei ähnlich wie der Buyer. Er nimmt die gewünschten QoS-Parameter entgegen und prüft seinerseits ob die angeforderten Dienste erbracht werden können. Anschließend beginnt er mit lokalen Verhandlungen im Endpunkt. War das Ergebnis jeweils positiv, so sendet er dem Buyer ein entsprechendes Signal, bzw. die modifizierten QoS-Parameter, und erwartet die Rückmeldung des Kommunikationssystems über die erfolgte oder nicht erfolgte Reservierung der Netzwerkressourcen.

3.2 QoS Verhandlung zwischen den Schichten

Nachdem die Verhandlungen zwischen den Endpunkten positiv beendet sind, initiiert die Anwendung die Verhandlungen mit dem Kommunikationssystem. Dazu müssen zunächst die Anwendungs-QoS-Parameter übersetzt werden, wie an dem folgenden Beispiel deutlich wird.
Beispiel:
Zur Übertragung von Videobildern fordert eine Applikation die folgenden Dienstqualitäten. Jedes Bild hat eine Auflösung von 640 x 480 Pixeln in Echtfarben, d.h. jedes Pixel mit 24 Bit kodiert mit einem Speicherbedarf M_A von folglich 900 KByte. Die Übertragungsrate R_A beträgt 30 Bilder/Sekunde. Für das Kommunikationssystem sind diese Werte jedoch nur Anhaltspunkte, sie müssen übersetzt und ergänzt werden.
Das Kommunikationssystem benutzt zur Kodierung von bewegten Videobildern ein MPEG2-Verfahren, das die Mediadatenpaketgröße wesentlich verringert. Außerdem werden Transportdatenpakete konstanter Länge M_N benutzt, in die die komprimierten Mediadaten eingepaßt, bzw. auf die sie verteilt werden müssen. Die von der Applikation benötigte Paketrate beträgt also $R_N = Cod(M_A)/M_N*R_A$. Weitere Beispiele finden sich in Abbildung 3.
Zusätzlich kann der QoS-Broker, in dem Wissen, daß es sich um eine Videoanwendung handelt, qualitative Dienste, wie Reihenfolgetreue verlangen und auf Neuversenden verzichten, da ein verlorenes Videobild einfacher zu verschmerzen ist, als der zusätzliche Speicher- und Zeitaufwand.
Wichtig bei der Übersetzung der QoS-Parameter von Applikations-QoS nach Netzwerk-QoS ist die Bidirektionalität der Übersetzung im Zuge der Modifikation und Aushandlung der Parameter. Im Beispiel der Übertragung bewegter Videobilder könnte ein Kommunikationssubsystem zum Beispiel die geforderte Übertragungsrate nicht erfüllen. Eine alternative, geringere Übertragungsrate muß nun vom QoS-Broker zurücktransformiert werden um die neuen Werte mit der Applikation abzustimmen. Die niedrigere Übertragungsrate kann sich nun jedoch in mehreren Formen des Qualitätsverlustes niederschlagen. So hat der QoS-Broker die Möglichkeit, die Auflösung der Bilder, die Verringerung der Farbtiefe oder die Verringerung der Übertragungsrate vorzuschlagen. Dies hängt von der Applikation ab. Um dieses Dilemma aufzulösen muß dem QoS-Broker Zusatzwissen über Präferenzen der Applikationen zur Verfügung gestellt werden, zum Beispiel in Form eines Regelwerks.
Die Verhandlung der QoS-Parameter im Netzwerk setzt sich vom Kommunikationssystem aus durch das Netzwerk bis in den Endpunkt der Gegenstelle fort und wird von dort bestätigt, modifiziert oder abgelehnt wieder zur Quittierung und Information zurückgesendet.

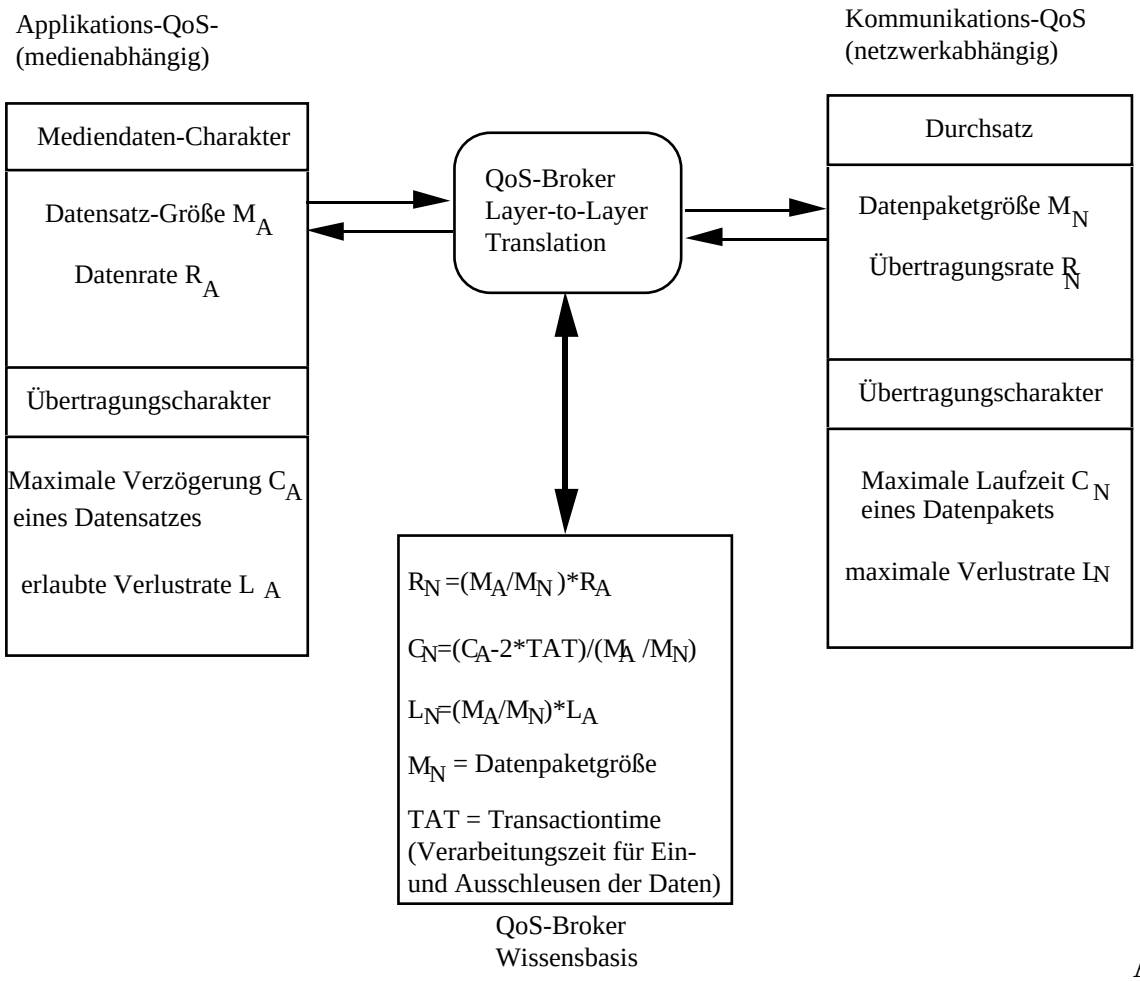


Abbildung 3: Bidirektionale Layer-To-Layer Übersetzung von QoS-Parametern

4 Der QoS-Broker im Einsatz

Ein QoS-Broker wurde im Rahmen einer Telerobotik-Anwendung des Distributed Systems Laboratory an der University of Pennsylvania experimentell implementiert. Im Versuch sollte ein Roboter über ein ATM-Netzwerk von einer Operationsstation aus ferngesteuert werden. Der ferngesteuerte Roboter versendet Videobilder einer Kamera und Audiodaten, die über ein Mikrofon erfaßt werden und sendet diese über ein ATM-Netzwerk zum Steuerungsrechner. Der Steuerungsrechner wird von einem Benutzer bedient und ist mit einem zweiten Roboter zur Kontrolle versehen (Abbildung 4).
Zu diesem Zweck werden Steuerungsdaten von der Steuerungsstation zum Roboter gesendet und Video-, Ton- und Sensordaten des Roboters zurück übermittelt. Die unterschiedlichen Merkmale der Sensor- und Videodaten bieten eine breite Abdeckung unterschiedlicher Medien-Anforderungen. Die Sensordaten benötigen zum Beispiel eine maximale Datengesamtlaufzeit von 20 ms, es dürfen keine zwei aufeinanderfolgenden Pakete verloren gehen bei einer Datenrate von 50 Abtastungen pro Sekunde mit allerdings lediglich 64 Bit pro Abtastung. Die Laufzeit der Videodaten darf bis zu 200 ms betragen, die Datenlast ist jedoch trotz Kompression vergleichsweise hoch.

Auf der Operatorseite wird ein Roboter mittels eines IBM-PC mit JIFFE-Echtzeit-Prozessorsystem kontrolliert, das über einen Bus-Connector mit einer IBM RS/6000 Workstation verbunden ist. Auf der Gegenseite wird der ferngesteuerte Roboter von einer SUN-4-Station gesteuert, die ebenfalls per Buskopplung an eine RS/6000-Station angeschlossen ist. Der Datentransport findet zwischen den RS/6000-Stationen statt, zwischen denen ein Hochgeschwindigkeits-ATM-Switch in einem ATM-LAN eingefügt ist. Der QoS-Broker, das Kommunikationssubsystem sowie die Multimedia-Unterstützung für Video und Ton sind jeweils auf den IBM-Workstations implementiert.

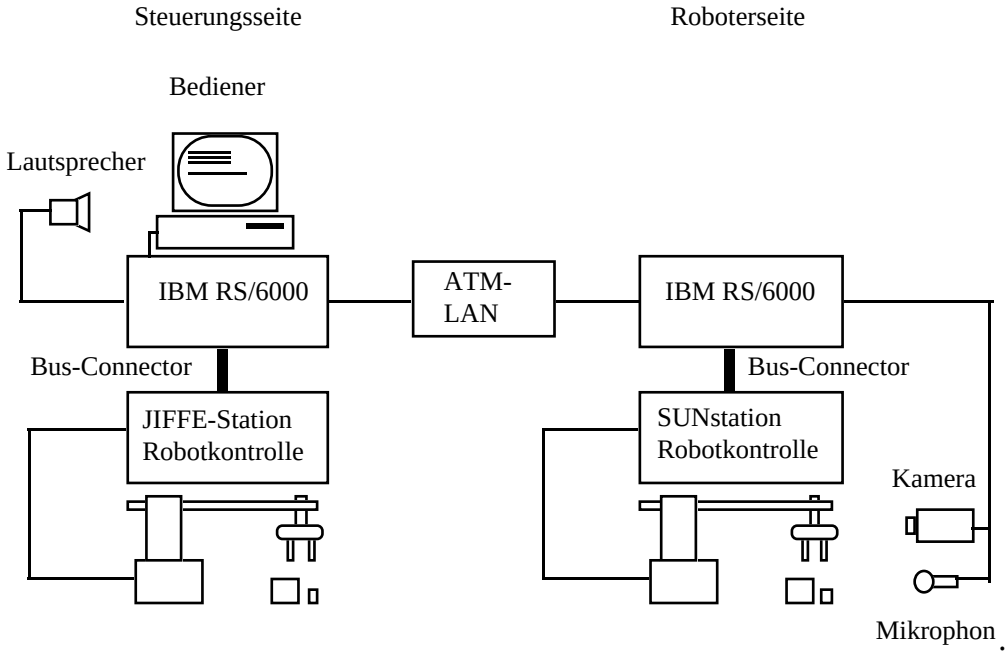


Bild 4: Experimenteller Aufbau einer Roboter-Steuerung mittels ATM-LAN

Der QoS-Broker ist als Dienstleistungs-Anwendungsprozeß implementiert. Er bietet Dienste zur Verwaltung und Verhandlung von Anwendungs- und Kommunikations-QoS-Parametern. Die Verhandlung der Anwendungs-QoS-Parameter geschieht durch Eintrag in die Broker-Tabelle, woraufhin die Zulassung der Parameter im Betriebssystem und der Gegenstelle überprüft wird. Das Resultat der Verhandlungen wird zurück übergeben.
Die Kommunikations-QoS-Parameter können richtungsabhängig zwischen den Schichten von oben nach unten, also von der Applikation zum Kommunikationssubsystem, wie auch umgekehrt verhandelt werden. Der erste Weg wird vom QoS-Buyer, der zweite vom QoS-Seller genutzt. Der Aufruf des QoS-Brokers geschieht über ein API (*Application Programmers Interface*) mit folgendem Aussehen:

```
QoSBroker( QOS-Parameter, Zusätzliche Parameter, Bemerkung, Seite, Richtung )
```

Die *QoS-Parameter* können sowohl Anwendungs- wie auch Kommunikations-QoS-Parameter enthalten, da der Broker sowohl von der Anwendungsseite (dem QoS-Buyer) als auch der Kommunikationsseite aufgerufen wird. Die Parameter werden in Tabellen gespeichert und vom Broker in der Verhandlung zwischen den Schichten umgesetzt. Die *zusätzlichen Parameter* können weitere nicht QoS-relevante Informationen enthalten, die zwischen Bediener- und Roboterseite ausgetauscht werden. Der Parameter *Bemerkung* enthält das Ergebnis der Verhandlungen und kann als *accept* (akzeptiert), *modified* (verändert) oder *rejected* (abgelehnt) bewertet werden. *Seite* kennzeichnet die beiden Teilnehmer der Telerobotik-Applikation. Diese Unterscheidung ist nötig, da die Endteilnehmer und die Weiterverarbeitung der zusätzlichen Parameter auf beiden Seiten auf unterschiedliche Weise

geschieht. So wird zum Beispiel während der Verhandlungen Informationen über die Position des Roboterarms ausgetauscht. Der Parameter *Richtung* kann die Werte*up* oder *down* annehmen und gibt an, welche Art von QoS-Parameter von wem eingegeben wurden. *Up* bedeutet, daß Anwendungs-QoS-Parameter von der Applikation in den QoS-Buyer übergeben wurden, bei *down* wurden Kommunikations-QoS-Parameter vom Kommunikationssubsystem an den QoS-Seller übergeben.
Die Aushandlung der verschiedenen QoS-Parameter geschieht über unterschiedliche Kanäle und dauerte nur wenige Millisekunden.
Da AIX im Augenblick keine garantierten Dienste anbietet und somit die Prozeßverwaltung mehr oder weniger dem Zufall überlassen ist mußten Einschränkungen im Versuchsaufbau gemacht werden. Es darf nur ein Benutzer im System arbeiten, nur eine Multimedia-Anwendung von der RS/6000 bedient werden. Durch diese Einschränkungen kann das Zeitverhalten wenn nicht garantiert so doch wenigstens einigermaßen vorhergesagt werden.
Da der Ressourcenverwalter im ATM-Netzwerk zum Versuchszeitpunkt noch nicht implementiert war, war der QoS-Broker nicht auf die Verfügbarkeit von Kommunikations-Ressourcen angewiesen. Diese wurden in dem sonst nur minimal ausgelasteten ATM-Netzwerk des Versuchs immer allokiert.
So blieben die Endpunkt zu Endpunkt-Verhandlungen, die im Versuch getestet wurden. Der Bediener fragte bei der Robotereinheit um die Übermittlung von Video- und Audiodaten an. Dabei wurden die lokalen Ressourcen jeweils geprüft, sowohl auf Initiatorseite als auch in der Gegenstelle, reserviert und schließlich koordiniert.

5 Zusammenfassung

Der QoS-Broker ist eine Instanz zur Vermittlung von garantierten Diensten zwischen Instanzen, die an einer verteilten Anwendung in einem Hochgeschwindigkeits- Kommunikationsnetzwerk beteiligt sind.
Seine Aufgaben gliedern sich in drei Breiche:

(1) Aushandlung von QoS-Parametern zwischen den beteiligten Instanzen,
(2) Übersetzung von QoS-Parametern zwischen den Schichten und
(3) Benachrichtigungen der Instanzen über geänderte QoS-Parameter.

Die Instanzen, die in Interaktion mit dem QoS-Broker stehen sind dabei:

(1) die initiierende Multimedia-Anwendung,
(2) das lokale Betriebssystem mit dem integrierten Multimediasystem,
(3) die Multimedia-Anwendung der Gegenstelle und
(4) das Kommunikationssystem als Schnittstelle zum Multimedia-Netzwerk.

Die zur Verfügung gestellten Dienste und Ressourcen werden dabei auf Grundlage von QoS-Parametern reserviert und zur Verfügung gestellt.
Die Verhandlungen über Betriebssystem-QoS finden dabei lokal statt, die zwischen Initiator-Anwendung und Anwendung in der Gegenstelle in einer Endpunkt-zu-Endpunkt-Sicht innerhalb einer Kommunikationsschicht und die zwischen Anwendung und Kommunikationssystem zwischen den Schichten mit Übersetzung der QoS-Parameter.
Die Implementierung des QoS-Brokers des Distributed Services Laboratory der Universität von Pennsilvania ist bislang stark eingeschränkt und verhandelt QoS-Parameter nur zwischen den Anwendungen in den Endpunkten.
In Zukunft wäre es wünschenswert, wenn Betriebssysteme Echtzeiteigenschaften und ein universelles Ressourcenverwaltungssystem besäßen, um so garantierte Dienste auch lokal zu ermöglichen.
Es besteht ein direkter Zusammenhang zwischen den angeforderten Dienstqualitäten und den entstehenden Kosten. Falls die Dienste kostenfrei sind, wird eine Anwendung immer mit dem größtmöglichen Aufwand betrieben werden. Deshalb sollte ein Zusammenhang zwischen der

Dienstgüte und den entstehenden Kosten bestehen, der ebenfalls durch den QoS-Broker berücksichtigt werden sollte.
Durch die getrennte Aushandlung von QoS-Parametern zwischen den Endpunkten und anschließend mit dem Kommunikationssystem könnte ein beträchtlicher Verhandlungsaufwand entstehen, wenn bereits zwischen den Endpunkten festgelegte QoS-Parameter vom Kommunikationssystem abgelehnt werden. Dann müssen die teilnehmenden Anwendungen möglicherweise ihre QoS-Parameter neu aushandeln. Dies ließe sich durch die Integration beider Verhandlungen in einem Schritt umgehen.

VERGLEICH ZWEIER SICHERHEITSARCHITEKTUREN FÜR OFFENE, VERTEILTE SYSTEME

Weihua Zhang

Diese Seminararbeit vergleicht zwei verschiedene Sicherheitsarchitekturen. Hierzu werden zunächst die Anforderungen an die Netzwerksicherheit, wie sie von der ISO definiert worden sind, vorgestellt. Hieran anschließend wird zunächst die von Muftic und Sloman erarbeitete Architektur vorgestellt. Es handelt sich hierbei um eine flexible Architektur, deren Komponenten international standardisiert sind. Durch verschiedene Kombinationen der Komponenten können alle von der ISO geforderten Funktionen erbracht werden. Die Architektur von Mirhakkak ist eine sehr einfache Architektur, welche aus nur zwei Komponenten besteht und Aspekte der Netzwerksicherheit allein in der Transportschicht behandelt.

0. Einleitung

Mit der zunehmenden Verbreitung offener verteilter Systeme spielt die Netzwerksicherheit eine immer größere Rolle und hat sich ein starkes Interesse an sicherer Kommunikation entwickelt. Da ein Angriff auf den Nachrichtenaustausch in Computernetzwerken durch Abhören, Modifizieren oder Wiederholen von Nachrichten in der Regel mit geringem Aufwand möglich ist, werden Mechanismen zur Gewährleistung einer sicheren Kommunikation benötigt. Aus diesem Grund sind eine Reihe von Ansätzen für die Integration von Sicherheitsaspekten in Netzwerkarchitekturen entwickelt worden. Diese Seminararbeit vergleicht zwei verschiedene Sicherheitsarchitekturen. Zunächst werden die Sicherheitsanforderungen, welche von der ISO identifiziert worden sind, kurz eingeführt. Im Anschluß hieran werden die beiden Architekturen vorgestellt und miteinander verglichen. Eine zusammenfassende Bewertung schließt dann diese Arbeit.

1. Anforderungen an die Netzwerksicherheit

Um Angriffe auf die Sicherheit in offenen verteilten Systemen zu verhindern werden eine Reihe verschiedener Sicherheitsdienste benötigt. ISO hat die folgenden identifiziert: /MuSl 94/

- Instanzauthentisierung (Entity authentication)
  Dieser Dienst prüft, ob die Identität einer Instanz tatsächlich diejenige ist, die vorgegeben wird.
- Überwachung der Zugriffsrechte (Access control)
  Hier wird die Entscheidung getroffen, ob ein Benutzer eine bestimmte Resource des offenen, verteilten Systems benutzen darf.
- Vertraulichkeit von Daten (Data confidentiality)
  Hiermit ist gemeint, daß nur vereinbarte Benutzer die Nachrichten, Dateien etc. lesen bzw. interpretieren können.
- Datenintegrität (Data integrity)
  Maßnahmen der Datenintegrität überwachen, ob Daten mit Absicht (illegale Benutzung) oder ohne Absicht (technische Fehler, z.B. Störung) verändert werden.
- Verbindlichkeit von Nachrichten
  Dieser Dienst kann in zwei Ausprägungen verstanden werden, welche beliebig miteinander kombiniert werden können:
  a) Der Empfänger muß nachweisen können, daß eine Nachricht tatsächlich von einem bestimmten Sender stammt.
  b) Der Sender muß nachweisen können, daß eine Nachricht tatsächlich von einem bestimmten Empfänger empfangen wurde.

2. Die Architektur von Muftic und Sloman (AMS)

2.1. Die Architektur und ihre Komponenten

Muftic und Sloman haben die folgende Architektur vorgeschlagen: /MuSl 94/

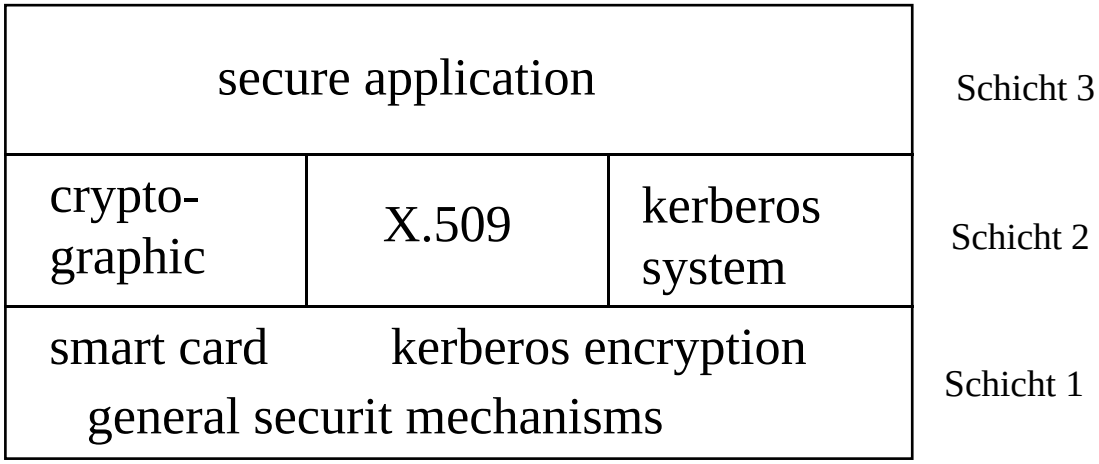


Bild 2-1 Die Architektur von Muftic und Sloman

Diese Architektur basiert auf drei Schichten. Die erste Schicht ist wird auch "Supporting cryptographic modules" genannt. Sie enthält drei Komponenten:

1) General security mechanisms enthält alle Algorithmen, die die darauf aufbauenden Sicherheitsdienste brauchen. Dies sind:
   - Methoden zur Erzeugung von großen Zufalls- und Primzahlen.
   - Symmetrische krytpographische Algorithmen (z.B. DES (Data Encryption Standard)).
   - Asymmetrische krytpographische Algorithmen (z.B. RSA (Rivest-Shamir-Adleman)).
   - Verschiedene Codes zur Gewährleistung der Nachrichtenintegrität (message integrity codes).

2) Smart-Card-Modules enthält
   - Funktionen für die Manipulation von Benutzer- und Smart-Card-Daten:
     - Aktivieren / Deaktivieren der Smart-Cards
     - Authentisierung von Benutzern
     - Veränderung von PIN´s
   - Funktionen des auf öffentlichen Schlüsseln beruhenden RSA-Verfahrens:
     - Erzeugen oder Verifizieren der auf RSA basierenden digitalen Unterschriften.
   - Funktionen des auf geheimen Schlüsseln basierenden DES-Verfahrens:
     - Verschlüsselung / Entschlüsselung
     - Erzeugen von geheimen Schlüsseln

- Erzeugen und Verifizieren von sogenannten "MAC-Codes" (message authentication certificates), welche als Bescheinigungen für die Authentisierung der Nachrichten dienen.

• Funktionen für die Smart-Card Verwaltung

3) Kerberos Encryption Module enthält die Funktionen, die das Kerberos System benötigt.
Diese drei Komponenten können nach verschiedenen Anforderungen flexibel ausgewählt werden. Selbstverständlich müssen die "general security mechanisms" immer vorhanden sein, da sie grundlegende Fähigkeiten implementieren, welche von allen anderen Modulen verwendet werden.
Die zweite Schicht enthält ebenfalls drei Komponenten. Kerberos System ist für die Instanzauthentisierung und die Überwachung der Zugriffsrechte zuständig. Cryptographic implementiert die restlichen drei Anforderungen der ISO. Falls domänenübergreifende Zugriffe erforderlich werden, so kann X.509 eingesetzt werden
Die dritte Schicht ist die Anwendungsschicht. Sie enthält die um Sicherheitsaspekte erweiterten Anwendungen.

2.2. Kerberos System /MuSl 94/,/Stal 94/,/Stah 93/

Kerberos ist eine am Massachusetts Institute of Technology (MIT) entwickelte Software. Die Hauptaufgaben des Kerberos Systems sind die Instanzauthentisierung und die Überwachung der Zugriffsrechte. Es besteht aus einem relativ zentralem Kerberos Authentisierungsserver (KAS Kerberos authentication server), in dem alle Benutzernamen und die zugehorigen Authentisierungsinformationen im Netzwerk gespeichert werden, einem Verwalter, dem sogenannten Ticket Granting Server (TGS), in welchem alle Zugriffsrechte des ganzen Netzwerk gespeichert werden und einem bei jedem Teilnehmer des Netzwerks vorhandenen Kerberos User Agent (KUA). Das ganze System wird anschaulich im Bild 2-2 dargestellt.

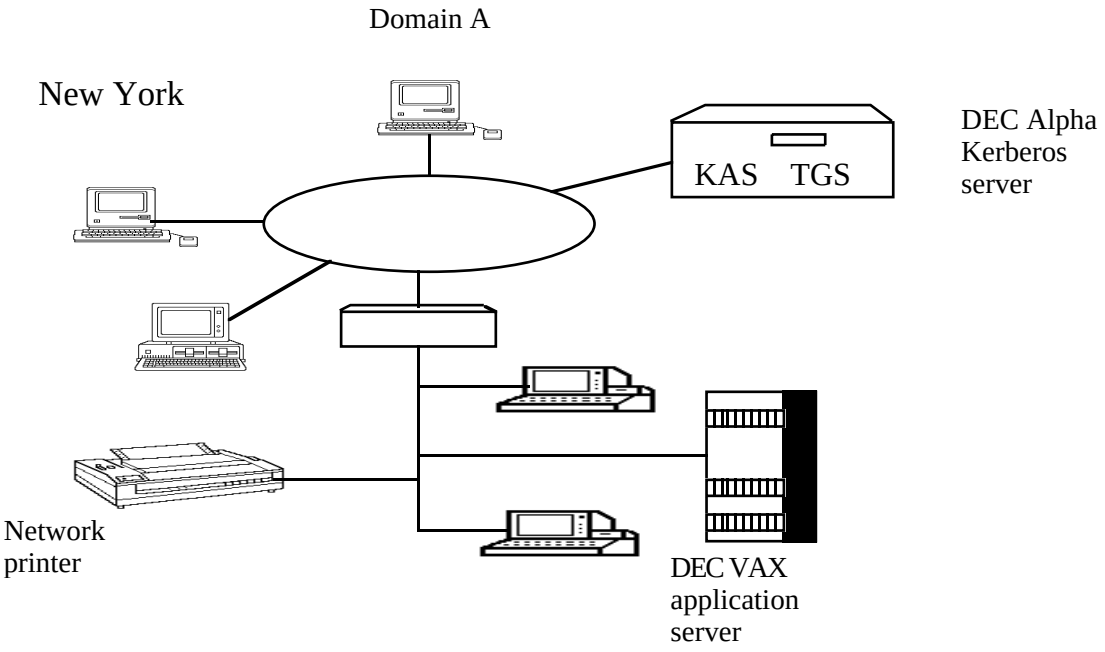
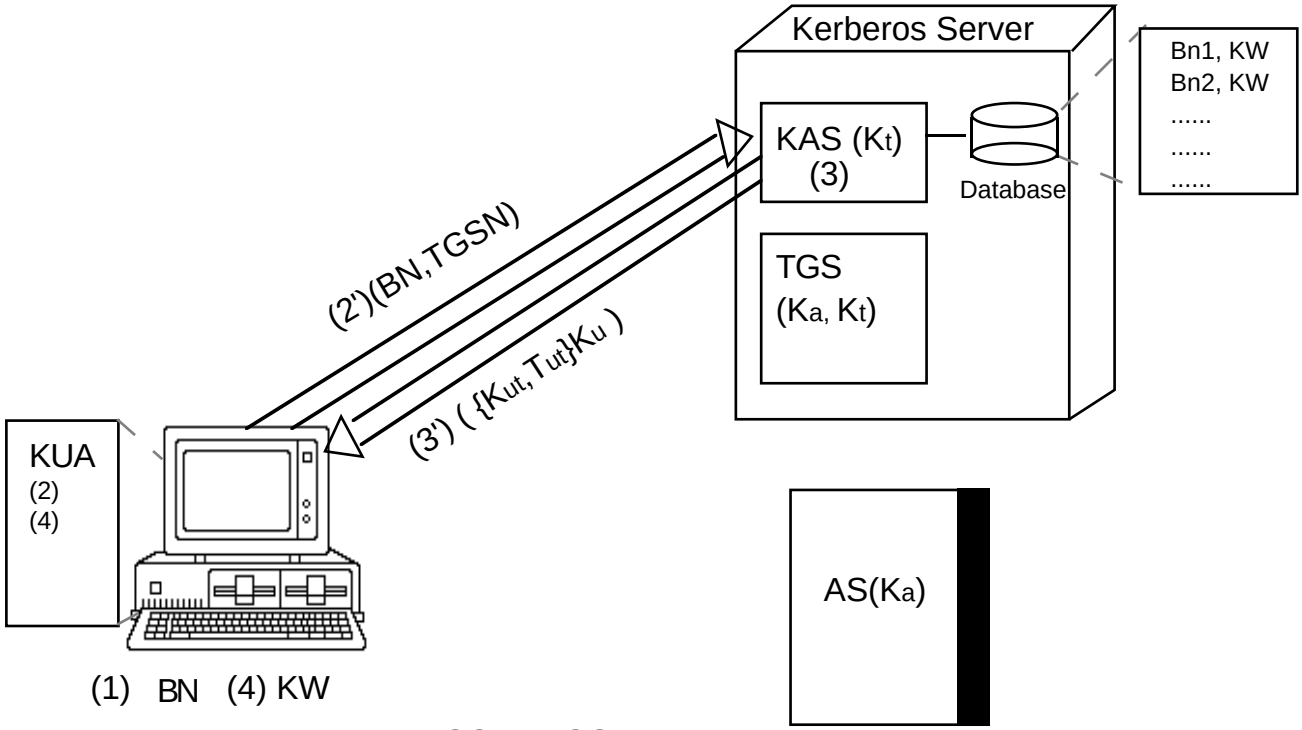


Bild 2-2 Kerberos System

Das Kerberos System basiert auf dem DES Algorithmus. Zwischen zwei kommunizierenden Partnern muß ein geheimer Schlüssel vereinbart werden, bevor diese sicher miteinander kommunizieren können. Zuerst muß sich die eine Kommunikation initiierende Instanz anmelden, z. B. durch Präsentation einer Idenentitätskennung und dem zugehörigen Paßwort. Diese Informationen sind in der Datenbank des KAS gespeichert. Änderungen können durch einen Verwalter durchgeführt werden. Die prinzipielle Interaktionen zwischen Kerberos Server und KUA bzw. Benutzer ist in den Bildern 2-3 und 2-4 dargestellt.



(1) BN (4) KW

BN: Benutzernamen; TGSN : TGS Name.

K_u : Schlüssel vom Benutzer, der durch Benutzerkennwort erzeugt wird.

K_t : Schlüssel vom TGS; K_a : Schlüssel vom AS.

K_{ut}: Schlüssel zwischen Benutzer und TGS

T_{ut}: Ticket zwischen Benutzer und TGS.

$T_{ut}=\{BN,TGSN,Ad_{BN},AZ, LDT, K_{ut}\}K_t$.

Ad_{BN} : Adresse des Benutzers.

AZ: Aktuele Zeit; LDT: Lebensdauer des Ticket

(Info): Info senden

{Info}K: mit K verschlüsseln

Bild 2-3 Login-Interaktion von Kerberos

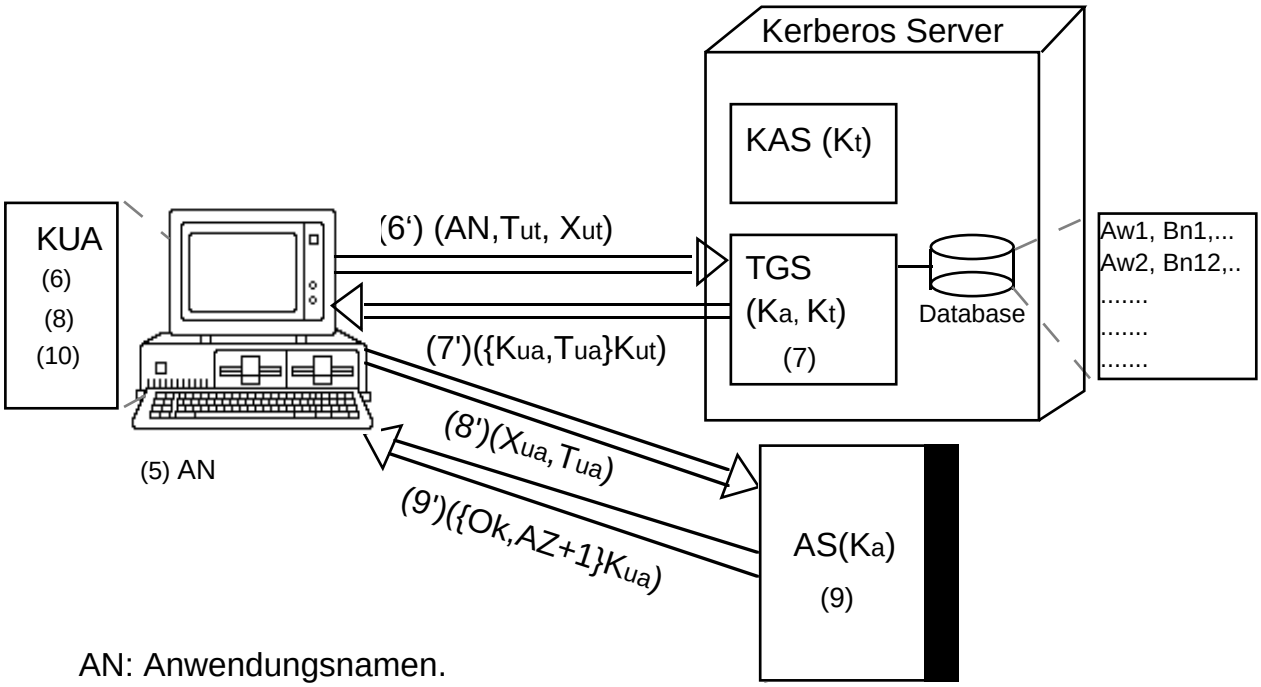
Im Bild 2-3 ist das Login-Verfahren dargestellt. Die Schritte hierbei sind:

1) Der Benutzer (BN) gibt seinen Namen ein.

2) Der KUA schickt den Namen (2') zum KAS.

3) Der KAS prüft, ob der BN eingetragen ist oder nicht. Falls ja, erzeugt der KAS mit BN's Kennwort einen Schlüssel K_u, zufällig einen sogenannten Sitzungsschlüssel K_{ut} und ein Ticket T_{ut}. Diese sendet er in der mit K_u verschlüsselten Nachricht (3') zu KUA zurück.

4) Der KUA verlangt das Kennwort des Benutzers. Nachdem der Benutzer sein Kennwort eingegeben hat, generiert der KUA mit diesem Kennwort wie auch bereits der KAS den Schlüssel K_u und entschlüsselt (3') mit diesem. Hiermit bekommt KUA den Konversationsschlüssel K_{ut} und das Ticket T_{ut}. KUA speichert beide Informationen und "vergißt" das Kennwort.



AN: Anwendungsnamen.
BN, Ad_{BN}, AZ, LDT, K_t, K_a, T_{ut} und K_{ut}: siehe Bild 2-3.
X_{ut}: Zeugnis des Benutzers für TGS.
$X_{ut}=\{BN,Ad_{BN},AZ\}K_{ut}$.
K_{ua}: Schlüssel zwischen Benutzer und AS.
T_{ua}: Ticket zwischen Benutzer und AS.
$T_{ua}=\{BN,ASN,Ad_{BN},AZ, LDT, K_{ua}\}K_a$.
ASN: AS Name.
X_{ua}: Zeugnis des Benutzers für AS.
$X_{ua}=\{BN,Ad_{BN},AZ\}K_{ua}$.
(Info): Info senden
{Info}K: mit K verschlüsseln

Bild 2-4 Anwendungsbenutzungs-Interaktion von Kerberos

Damit ist das Login-Verfahren abgeschlossen und das System kann bisher noch nicht feststellen, ob das Paßwort falsch ist. Für den Fall, daß das Kennwort falsch ist, sind K_{ut} und T_{ut} falsch und weitere Operation können nicht durchgeführt werden, da diese auf K_{ut} und T_{ut} beruhen.
Bild 2-4 stellt das Verfahren dar, mit dem der Benutzer eine Anwendung benutzen kann.

5) Wenn ein Benutzer eine Anwendung benutzen möchte, gibt er den Anwendungs–namen ein.

6) KUA erzeugt das Zeugnis X_{ut} und schickt die Nachricht (6') zu TGS

7) TGS entschlüssel T_{ut} mit K_t und erhält so Informationen über den Benutzer und den Konversationsschlüssel K_{ut}. TGS prüft diese Information mit X_{ut}. Falls die Überprüfung erfolgreich verläuft, prüft er, ob der Benutzer das Recht hat, die gewünschte Anwendung zu benutzen. Ist dies der Fall, so erzeugt er noch einmal zufällig einen weiteren Konversationsschlüssel K_{ua} und ein Ticket T_{ua}. Dann schickt er die Nachricht (7') zu KUA zurück.

8) KUA entschlüsselt die Nachricht (7') mit K_{ut} und erhält so K_{ua} und T_{ua}. Dann erzeugt er ein Zeugnis X_{ua}, und sendet (8') zu AS.

9) Der Anwendungsserver (AS) entschlüsselt T_{ua} mit K_a wie (7) und vergleicht die Informationen in T_{ua} mit denen in X_{ua}. Bei Übereinstimmung sendet er eine positive Antwort (9') zum KUA zurück.

10) Der Benutzer kann nun die gewünschte Anwendung benutzen.

Damit ist die Überwachung des Zugriffsrechts abgeschlossen. Möchte der Benutzer weitere Anwendungen benutzen, wiederholen sich die Schritte ab (5).

2.3. Cryptographic /MuSl 94/,/X.509 88/

Dieser Dienst dient zur Realisierung der restlichen drei Anforderungen der ISO:

1) Datengeheimnis:
   Benutzer und Anwendung tauschen Daten nur mit K_{ua} verschlüsselt aus.

2) Datenintegrität:
   Dieser Dienst überwacht bei einem Nachrichtenaustauch, ob die ausgetauschten Nachrichten mit oder ohne Absicht verändert wurden. Hierzu werden sogenannte digitale Unterschriften verwendet.
   Digitale Unterschriften basieren auf dem RSA-Algorithmus. Für den RSA-Algorithmus benötigt man zwei Schlüssel, von denen einer geheim und ein anderer öffentlich ist. Sind die Informationen mit dem geheimen Schlüssel verschlüsselt worden, so kann die Information nur mit dem öffentlichen Schlüssel entschlüsselt werden. Analoges gilt für die umgekehrte Richtung. Bei einer digitalen Unterschrift wird folgende Information gesendet:
   (Auftrag/Ergebnis, K_s{MAC}), mit MAC = h(Auftrag/Ergebnis).
   wobei K_s der Geheimschlüssel desjenigen ist, der die digitale Unterschrift leistet.Mit h wird eine Funktion (z.B. eine Hashfunktion) bezeichnet, mit welcher die Prüfsumme (MAC) berechnet wird. Auf der Empfängerseite wird K_s{MAC} mit dem zugehörigen öffentlichen Schlüssel K_p entschlüsselt und ebenfalls MAC' = h(Auftrag/Ergebnis) berechnet. Falls MAC = MAC' ist, wird angenommen, daß Auftrag und Ergebnis zuverlässig sind.

3) Verbindlichkeit von Nachrichten:
   Auftraggeber und Auftragnehmer werden mit A bzw. B bezeichnet.

a) A sendet mit $K_{sA}\{N\}$ => hier kan jeder, der den öffentlichen Schlüssel K_{pA} hat, die Nachricht N lesen und beweisen, daß N von A gesendet wurde, da nur A den Schlüssel K_{sA} besitzt.

b) A sendet mit $K_{pB}\{N\}$ => Nur B kann die Nachricht entschlüsseln, da nur B den zugehörigen Schlüssel K_{sB} besitzt.

Nach Empfang entschlüsselt B die Nachricht N mit K_{sB} ($N=K_{sB}\{K_{pB}\{N\}\}$) und quittiert N mit $K_{sB}(Q_n)$. => A kann beweisen, daß B die Nachricht empfangen hat.

c) A->B: $K_{pB}\{ K_{sA}\{N\}, A \}$

=> Nur B kann die Nachricht lesen und beweisen, daß sie von A kommt.

B->A: $K_{pA}\{ K_{sB}\{N\}, B \}$

=> Nur A kann die Nachricht lesen und beweisen, daß sie von B kommt.

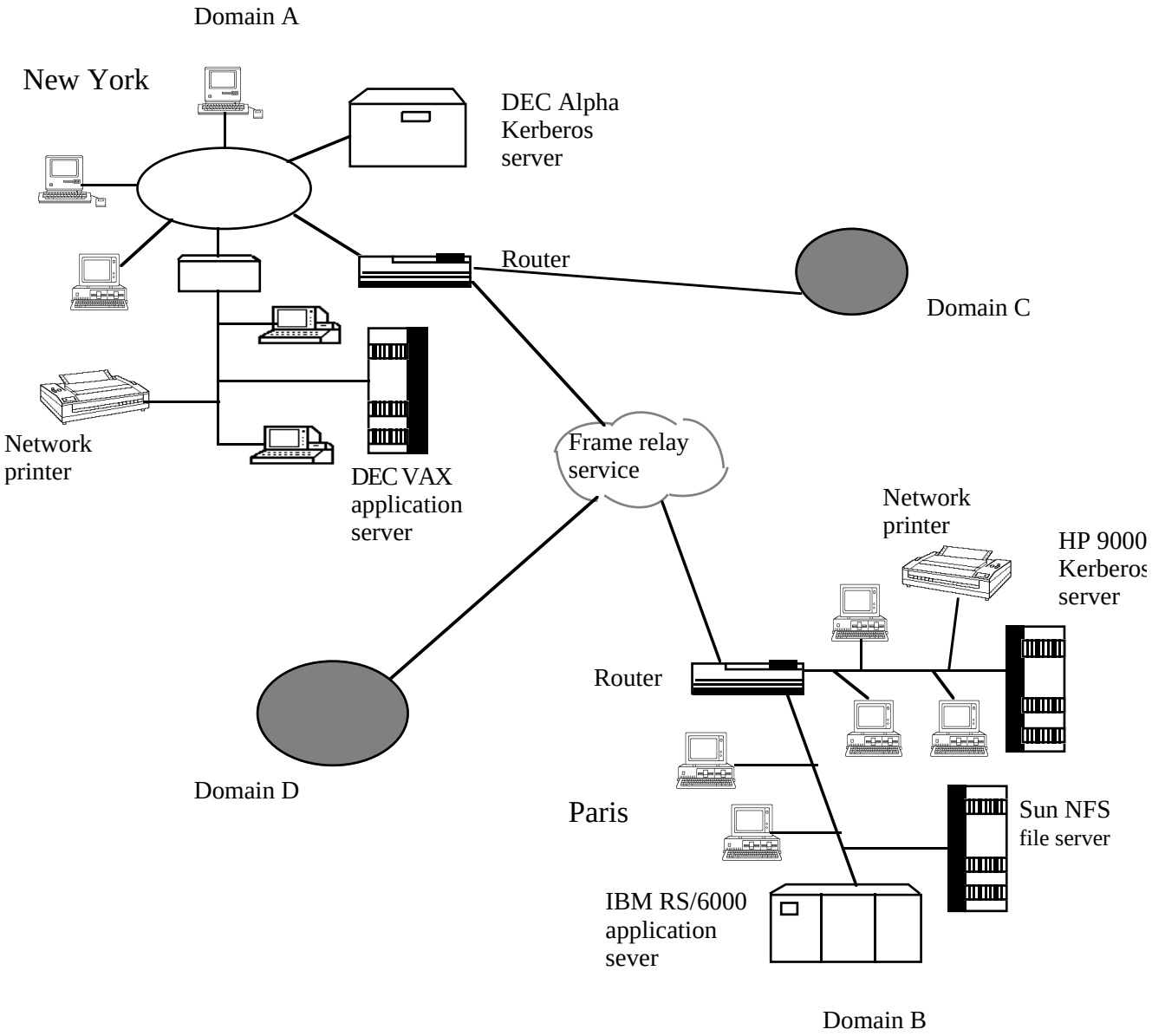


Bild 2-5 Interaktion iüber Domänengrenzen hinweg bei Kerberos

2.4. X.509 /MuSl 94/, /X.509 88/

Bild 2-5 zeigt ein großes Netzwerk, das in mehrere Domänen unterteilt ist. Wenn ein Benutzer, welcher in Domäne A (New York) registriert ist, eine Anwendung in Domäne B (Paris) benutzen möchte, erfordert das einen Sicherheitsmechanismus, welcher über Domänengrenzen hinweg operiert (cross-domain operation). Es gibt in Kerberos einen Mechanismus für dieses Problem. Dieser funktioniert folgendermaßen: der TGS einer jeden Domäne hat mit jedem TGS der anderen Domänen einen geheimen Schlüssel vereinbart. Möchte ein Benutzer eine Anwendung in anderer Domäne benutzen, verlangt er zuerst ein Ticket von seinem eigenen TGS um mit dem TGS der betreffenden anderen Domäne in Kontakt zu treten. Mit diesem Ticket, bewirbt sich der Benutzer bei dem entfernten TGS um die Benutzungsgenehmigung der gewünschten Anwendung zu erhalten. Dieses Ticket, das der Benutzer zum fernen TGS schickt, beweist die Identität des Benutzers in Domäne A. Der TGS in der Domäne B entscheidet nun, ob der Benutzer der Domäne A das gewünschte Programm benutzen darf oder nicht. Bei dieser Entscheidung muß der TGS in Domäne B dem TGS in Domäne A vertrauen.
Dieses Verfahren hat einen Nachteil, welcher darin begründet ist, daß Kerberos auf dem DES und somit auf symetrischer Kryptografie basiert. Besteht ein großes verteiltes System aus n Domänen so werden insgesamt n*(n-1)/2 Schlüssel benötigt, damit zwischen jedem Paar von Ticket Granting Servern ein privater Schlüssel vereinbart werden kann. Im Falle n=1000 werden beispielsweise ca. eine halbe Millionen Schlüssel benötigt. Das impliziert eine große Netzbelastung, wenn alle Schlüssel regelmäßig aus Sicherheitsgründen durch neue Schlüssel ersetzt werden.
Für Interaktionen über Domänengrenzen hinweg empfiehlt sich daher der Einsatz von Verfahren, die auf asymetrischer Kryptografie beruhen. X.509 ist ein internationaler Standard der CCITT, welcher auf dem asymetrischen RSA-Algorithmus beruht. X.509 benutzt sogenannte Bescheinigungen (Certificates). Ein Bescheinigung ist eine spezielle Datenstruktur, welche die Benutzeridentität und den zugehörigen öffentlichen Schlüssel enthält und durch eine sogenannte Certification Authority zertifiziert wird. Die Certification Authority hat ähnliche Aufgaben wie der KAS bei Kerberos.
Die Bescheinigungsverwaltung bei X.509 besteht aus der CA und den Protokollen, welche die Funktionen für die Verwaltung ausführen. Die CA sind in einer Hierarchie organisiert.
Ein Bescheinigungsverwaltungssystem besteht aus zwei Arten von Funktionen: zum einen aus den Benutzerfunktionen, die in den Workstations implementiert werden müssen, wie z.B. Erzeugen von Bescheinigungen, Anträge auf Zertifizierung bei der CA, Empfangen der Bescheinigungen und Verifikation der Bescheinigungen von Partnern. Diese Funktionen könnten von um Smart-Card-Funktionalität erweiterten Kerberos User Agents erbracht werden.
Die andere ist die Menge aller Protokolle, welche die Bescheinigungsverwaltung auf CA-Servern implementieren. Sie ermöglichen allen Servern, ihre eigenen Bescheinigungen zu erzeugen, die Bescheinigungen der ihnen untergebenen Ebene zu unterschreiben und miteinander sowie mit den ihnen auf unterster Ebene unterstehenden Benutzern zu kommunizieren. Bei X.509 müssen bei n Domänen nur n öffentliche Schlüssel ausgetauscht werden.

2.5 Sichere Anwendungen

Die oberste Schicht der Architektur umfaßt die um Sicherheitsaspekte ergänzten Anwendungen. In /MuSl 94/ werden zwei Beispielanwendungen, Privacy Enhanced Mail (PEM) und Secure EDIFACT vorgestellt. Beide Anwendungen sind basieren auf dem X.509 Standard, da sie Interaktionen über Domänengrenzen hinweg erfordern. Secure EDIFACT verwendet zusätzlich noch Smart-Card-Funktionen.

3. Die Architektur von Mirhakkak (AM)

Die Hauptkomponenten der Architektur von Mirhakkak /Mirh 93/ sind der sogenannte "Reference Monitor" (RM) und das Transport Layer Security Protocol (TLSP). Sie sollen im folgenden näher beschrieben werden.

3.1. Reference Monitor

Mit dem Begriff "Reference Monitor" wird die Menge aller Sicherheitsmechanismen in einem einzelnen Computer bezeichnet, welche dafür zuständig sind die Sicherheits–richtlinien des Computers durchzusetzen. In einem einzelnen Computer ist der RM dafür zuständig die Interaktionen zwischen den drei grundlegenden Arten von Instanzen, nämlich Benutzern, Prozessen und Anwendungsdaten zu kontrollieren. Benutzer und Prozesse sind aktive Einheiten und werden daher Subjekte genannt. Anwendungsdaten, wie z.B. Dateien oder Datenobjekte im Hauptspeicher werden Objekte genannt. Zusammenfassend kann man die Funktion des RM als Kontrolle der Zugriffe von Subjekten auf Objekte bezeichnen.
In einem verteilten System kooperieren die RM´s der einzelnen Computer und konstituieren auf diese Weise den RM des ganzen verteilten Systems.
Der RM eines verteilten System ist für die gleichen Funktionen verantwortlich wie der RM eines Einzelsystems, aber mit dem Unterschied, daß die drei Klassen von Instanzen irgendwo im verteilten System angesiedelt sein können. Interaktionen in einem verteilten System sind nur zwischen Prozessen möglich. Bild 3-1 illustriert die in einem verteilten System möglichen Interaktionen.

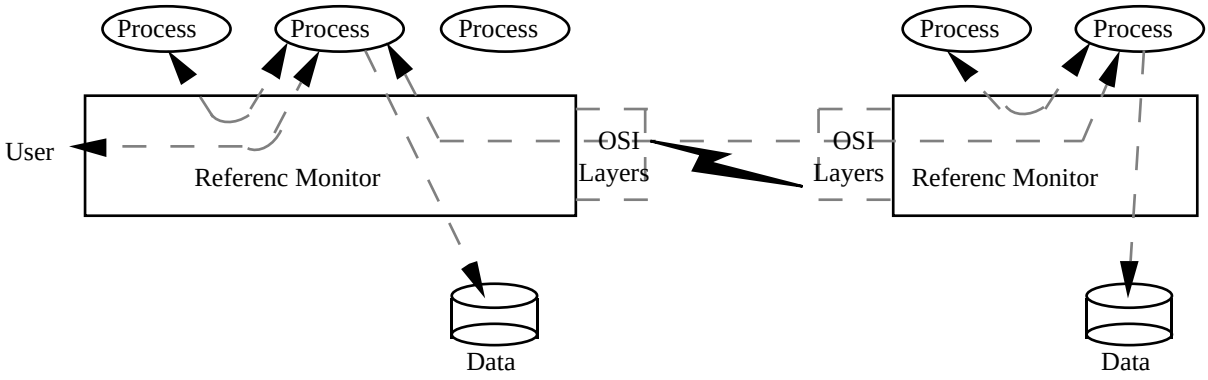


Bild 3-1 Verteiltes System Modell

Der RM einer jeden Komponente hat einen Autorisierungsdienst, welcher die zu lokalen und entfernten Instanzen gehörenden Sicherheitsiinformationen und -attribute speichert. Diese Informationen dienen dazu, zu bestimmen, ob ein entfernter oder lokaler Zugriff erlaubt werden soll oder nicht. Die Autorisierungsdienste der einzelnen Komponenten kooperieren miteinander und bilden so den Autorsierungsdienst des verteilten Systems.
Die Sicherheitsattribute müssen aber nicht in jedem Rechner statisch gespeichert werden, sondern können bei Bedarf dynamisch von den Autorisierungsdiensten anderer Rechner erfragt werden.

3.2. Transport Layer Security Protocol

Das Transport Layer Security Protocol ist im Dezember 1992 als internationaler Standard 10736 von der ISO/IEC verabschiedet worden. Es ist zwischen dem oberen Teil des Transportprotokolls, welcher für die Konstruktion von TPDUs für Protokolloperationen zuständig ist, und dem unteren Teil des Transportprotokolls angesiedelt, welcher die Aufgabe hat, TPDUs unter Benutzung der Schnittstelle zur Vermittlungsschicht und ihrer Dienstprimitive zu senden und zu empfangen. Das deutet darauf hin, daß das Transport-protokoll unabhängig von dem TLSP ist und daß das TLSP auch unabhängig von den Dienstprimitiven der Vermittlungsschicht ist. Darum kann für das verbindungsorientierte und das verbindungslose Transportprotokoll das gleiche TLSP verwendet werden.
Die grundlegende Funktion des TLSP ist es, für jede TPDU einen geschützten und einen ungeschützten Kopf hinzuzufügen (siehe Bild 3-3). Der geschützte Kopf und die eigentliche TPDU bilden den geschützten Teil während der Übertragung. Je nach dem, welcher Sicherheitsdienst gewünscht wird, wird der geschützte Teil der TPDU durch eine Prüfsumme ergänzt oder auch mit einem geeigneten Algorithmus verschlüsselt. Müssen mehrere TPDU´s mit gleichen Sicherheitsanforderungen zwischen den gleichen Dienstzugangspunkten in Quell- und Zielrechner transportiert werden, so können sie gemeinsam eingekapselt und übertragen werden.

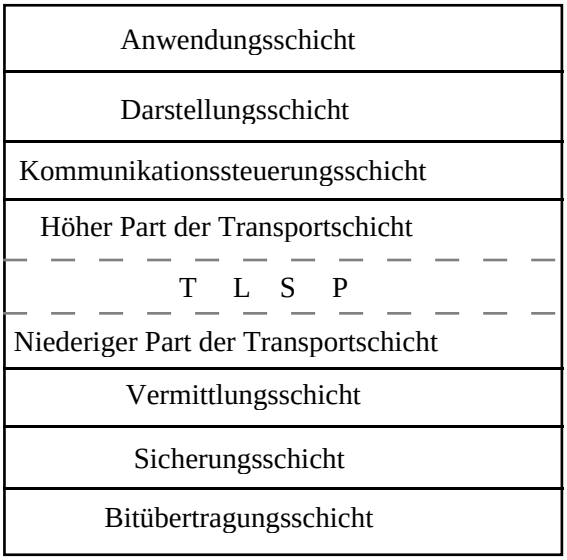


Bild 3-2 Lage des TLSP im ISO 7-Schichtenmodell

Der geschützte Kopf enthält Informationen wie z.B. eine Sicherheitskennung, in der die Sicherheitsklassifikation und die zuständige Autorisierungsinstanz, welche mit der TPDU assoziiert wird, kodiert ist. Soll die Vertraulichkeit der zu übertragenen Daten gewährleistet werden, so wird der geschützte Teil der TPDU verschlüsselt. Soll lediglich die Unversehrtheit der Daten garantiert werden, so ist es ausreichend an den geschützten Teil der TPDU eine Prüfsumme, Integrity Check Value (ICV) genannt, welche aus der TPDU und einem geheimen Schlüssel generiert wird, anzuhängen. Eine Änderung der Daten während der Übertragung kann entdeckt werden, weil der beim Ziel berechnete ICV-Wert von dem übertragenen Wert abweichen wird. Müssen sowohl Unversehrtheit als auch Vertraulichkeit der zu übertragenen Daten garantiert werden, so werden beide Verfahren miteinander kombiniert. Die Prüfsumme kann hierbei aber auch ohne geheimen Schlüssel generiert werden, da sie bereits durch die anschliessende Verschlüsselung geschützt wird.
Einige Informationen müssen die TLSP-Instanzen ungeschützt austauschen können. Diese werden in dem ungeschützten Teil der TPDU übertragen. Zu diesen Informationen gehören

beispielsweise die Versionsnummer des TLSP-Protokolls und eine sogenannte Sicherheits-Assoziations-Kennung (Security Association Identifier, SA-Id). Diese Kennung referenziert eine vorher etablierte Sicherheitsassoziation (SA).
Bevor nämlich zwei TLSP-Instanzen sicher miteinander kommunizieren können, müssen einige Parameter, wie z.B. Verschlüsselungsalgorithmen und verwendete Schlüssel, vereinbart werden. Dies geschieht durch den Aufbau einer Security-Association. Der Aufbau von solchen Sicherheitsassoziationen muß aber nicht unbedingt durch das TLSP selbst bewerkstelligt werden, sondern kann unter Umständen auch den höheren Schichten überlassen bleiben. Dies ist eine Frage des Sicherheitsmanagement, welches hier nicht weiter betrachtet wird.

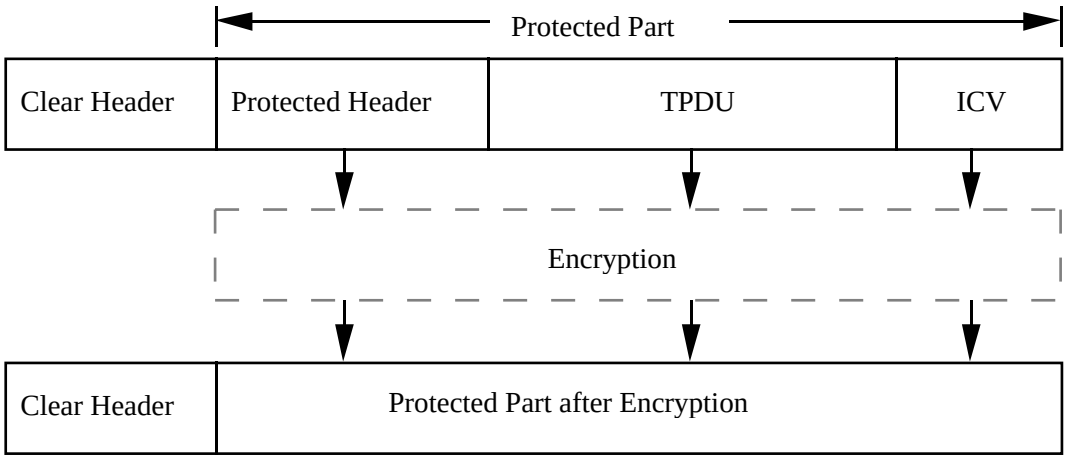


Bild 3-3 Kapselung und Verschlüsselung von TPDUs

Nicht alle von der ISO geforderten Sicherheitsdienste können durch das TLSP realisiert werden. Es ist nicht möglich die Verbindlichkeit von Nachrichten mit dem TLSP zu garantieren. Die folgenden Dienste können aber mit dem TLSP realisiert werden:

- Datengeheimnis
  wird von dem TLSP durch Verschlüsselung der eingekapselten Daten garantiert.
- Datenintegrität
  kann durch das ICV Feld gewährleistet werden.
- Instanzauthentisierung
  kann von dem TLSP durch die gleichen Mechanismen, mit denen auch das Datengeheimnis und die Datenintegrität gewährleistet werden, realisiert werden. Hierfür werden die Instanzadressen dem geschützten Teil zugeschlagen. Es gibt zwei Formen dieses Dienstes. Zum einen muß die Datenherkunft (data-origin) bei dem verbindungslosen Transportdienst authentisiert werden; im verbindungsorientierten Fall müssen beide Partnerinstanzen (peer-entity) authentisiert werden. Wird nur die Datenherkunft authentisiert, so ist es ausreichend in die Prüfsummenbildung oder die Verschlüsselung das Quelladreßfeld einzubeziehen. Im verbindungsorientierten Fall werden beide Adressen entsprechend geschützt.
- Zugriffsrechtüberwachung
  ist eigentlich die Aufgabe des RMs. Das TLSP kann nur die Verbindung zwischen Prozessen überwachen.

4. Vergleich beider Architekturen

Die von Muftic und Sloman vorgeschlagene Architektur (AMS) ist eine sehr komplexe Architektur. Sie hat drei Schichten. Jede Schicht hat mehrere Komponenten. Diese Komponenten können nach verschiedenen Anforderungen flexibel ausgewählt werden. Obwohl jede der Komponenten isoliert nur einen Teil der von der ISO definierten Anforderungen erfüllen kann, kann die gesamte Architektur alle geforderten Dienste erbringen. Innerhalb einer Domäne kann beispielsweise Kerberos, ergänzt um einige kryptografische Funktionen, das volle Leistungsspektrum erbringen. In größeren Netzwerken kann für die Authentisierung und Authorisierung domänenübergreifender Interaktionsbeziehungen X.509 gewählt werden. Die AMS ist völlig unabhängig von dem verwendeten Kommunikationsprotokoll. Alle Funktionen der AMS können außerhalb des 7-Schichten-Modells der ISO implementiert werden. Das heißt, es ist für diese Sicherheitsarchitektur unerheblich, ob das verwendete Kommunikationssystem nach dem 7 Schichtenmodell aufgebaut ist. Daher eignet sie sich sowohl für größere Netze, in denen alle 7 Schichten des Referenzmodells implementiert sind, als auch für schlankere Ansätze, wie z.B. Feldbussysteme, welche nicht alle Schichten des Referenzmodells implementieren. An dieser Stelle darf aber nicht unerwähnt bleiben, daß die Sicherheitsheitsarchitektur selber nicht gerade einen schlanken Ansatz darstellt.
Kerberos basiert auf zentralen Verwaltern, dem KAS und dem TGS, welche in jeder Domäne vorhanden sein müssen. In diesen Verwaltern werden alle Benutzer und Anwendungen einer Domäne verwaltet. Ihre Zuverlässigkeit und Sicherheit spielen eine sehr große Rolle für die Sicherheit und Verfügbarkeit des ganzen Systems. Fallen sie aus, so kann das ganze System (in einer Domäne) nicht mehr funktionieren. Je größer das System (die Domäne) ist, desto größer ist der Schaden bei einem Ausfall. Außerdem können die beiden Verwalter zu einem Leistungsengpaß werden. In diesem Sinne ist Kerberos nur für kleinere Domänen geeignet. Größere Netzwerke müssen daher in Domänen unterteilt werden. Für domänenübergreifende Authentisierung eignet sich X.509, da es auf asymetrischen Verschlüsselungsverfahren beruht und keinen zentralen Schlüsselverwalter benötigt. Die Architektur von Muftic und Sloman kann also für Netzwerke beliebiger Größe verwendet werden.
Obwohl die einzelnen Komponenten bereits existieren haben die Autoren noch nicht genau spezifiziert, wie die Komponenten als Gesamtsystem zusammenwirken sollen. Ein wesentlicher Punkt ist, wie Kerberos und X.509 miteinander kombiert werden können.
Die Architektur von Mirhakkak (AM) ist dagegen eine sehr einfache Architektur. Sie hat nur zwei Komponenten: den Reference Monitor (RM) und das Transport Layer Security Protocol (TLSP). Auch bei dieser Architektur erfüllen die einzelnen Komponenten nur einen Teil der Forderungen der ISO. Z.B. kann das TLSP zwischen zwei Computer–systemen das Datengeheimnis, die Datenintegrität und einen Teil der Authentifizierung leisten, während der RM in jedem isolierten Computersystem die Sicherheits–anforderungen garantieren muß. Die Verbindlichkeit von Nachrichten kann von der Architektur in der bisher vorgeschlagenen Form noch nicht gewährleistet werden. Darüberhinaus ist auch nicht näher spezifiziert worden, wie die Schlüssel in einem Netzwerk verwaltet werden sollen. Diesbezüglich ist lediglich auf ein außerhalb der Architektur liegendes Sicherheitsmanagement verwiesen worden. Die Architektur erscheint daher noch erweiterungsbedürftig.
Obwohl die Architektur von Mirhakkak prinzipiell unabhängig von dem verwendeten Kommunikationsprotokoll ist, so hängt sie doch von dem Kommunikationsmodell ab. Da TLSP in der Schicht 4 angesiedelt ist, muß das Kommunikationssystem mindestens eine Transportschicht haben. In kleineren Systemen, die keine Transportschicht haben, wird diese Architektur daher höchstens in modifizierter Form einsetzbar sein.
In /Mirh 93/ ist kein zentraler Verwalter erwähnt worden. Ob sich die Architektur auch für größere Netze eignet, ist abhängig vom Verschlüsselungsverfahren in dem TLSP. Hier ist es denkbar, daß für Interaktionen innerhalb einer Domäne ein symetrisches und für domänenübergreifende Interaktionen ein asymetrisches Verschlüsselungsverfahren eingesetzt wird. Welches Verfahren bei einer konkreten Verbindung verwendet wird, kann dann beim

Aufbau der Sicherheitsassoziation festgelegt werden. Zusammenfassend erscheint daher auch diese Architektur für größere Netzwerke geeignet.

5. Zusammenfassung

In diesem Aufsatz wurden zwei Sicherheitsarchitekturen für offene, verteilte Systeme miteinander verglichen. Die Architektur von Muftic und Sloman ist eine sehr umfassende und flexible Architektur. Ihre Komponenten sind international standardisiert. Mit der Architektur können alle von der ISO gefordeten Sicherheitsdienste erbracht werden. Sie eignet sich sowohl für größere als auch kleinere Netzwerke. Die von Mirhakkak vorgeschlagene Architektur stellt einen sehr einfachen Ansatz dar. Sie besteht aus nur zwei Komponenten. An einigen Stellen ist sie noch erweiterungsbedürftig, da mit ihr die Verbindlichkeit von Nachrichten nicht garantiert werden kann und auch noch nicht definiert worden ist, wie Sicherheitsassoziationen aufgebaut werden können. Die eigentliche Authentisierung über Rechnergrenzen hinweg wird bei dieser Architektur nämlich beim Aufbau der Sicherheitsassoziation stattfinden. Auch diese Architektur eignet sich sowohl für kleinere als auch für größere Netzwerke, da das Verschlüsselungs–verfahren flexibel gewählt werden kann.

Literatur:

/Mirh 93/ Mohammad Mirhakkak: " A Distributed System Security Architecture: Applying the Transport Layer Security Protocol", ACM SIGCOMM Computer Communication Review, Vol 23, Number 5, 1993

/MuSl 94/ Sead Muftic and Morris Sloman: " Security Architecture for Distributed Systems ", Computer Communications, Vol. 17, No. 7, 1994

/Stah 93/ Dr. Stanley H. Stahl: " Information Security in Workstation Enviroments", Computer & Security, December 1993

/Stal 94/ William Stallings: "Kerberos Keeps the Enterprise Secure", Data Communications, Oct. 1994

/X.509 88/ CCITT X.509, "The Directory Authentication Framework", Melbourne, 1988

SICHERHEITSMANAGEMENT IN OFFENEN SYSTEMEN

Daniel Müller

Für viele Anwendungen in offenen Systemen ist die Sicherheit des Systems eine wichtige Voraussetzung. Dies wird in der vorliegenden Arbeit zunächst begründet, indem Werte und Bedrohungen in offenen Systemen identifiziert werden. Geeignete Schutzmaßnahmen werden vorgestellt und es zeigt sich, daß zu deren Koordination ein Sicherheitsmanagement notwendig ist. Auf die Struktur eines solchen wird anhand des entsprechenden OSI-Entwurfs genauer eingegangen. Die Vereinheitlichung des Sicherheitsmanagements für die in offenen Systemen vorhandene Vielfalt von verschiedenen Sicherheitsarchitekturen ist das Ziel des SAMSON-Projekts, welches im zweiten Teil der Arbeit kurz vorgestellt wird.

Einleitung

Offene Systeme sind Netzwerke von Rechnern verschiedenster Art, die auf eine Weise miteinander verbunden sind, die den freien Informationsaustausch zwischen beliebigen Teilnehmern zu den unterschiedlichsten Zwecken erlaubt. Die zugrundeliegende Netzstruktur ist häufig so komplex, daß keine Komponente eine vollständige Übersicht über das gesamte Netz haben kann. Eine zentrale Kontrolle, die für alles verantwortlich wäre und auf die sich alle verlassen könnten, kann es in einem solchen Netzwerk nicht geben.
Die Daten, die innerhalb eines offenen Systems übertragen werden oder auch in Datenbanken im System abrufbar sind, haben bestimmte Werte, also Eigenschaften, deren Aufrechterhaltung wichtig ist, wenn die Daten ihre Zweckbestimmung erfüllen sollen. Für kommerzielle Daten beispielsweise ist es häufig wesentlich, daß sie vertraulich behandelt werden. Ein anderes Beispiel sind die Verwaltungsdaten des Netzes selbst, an denen vor allem ihre Unversehrtheit für den ungestörten Netzbetrieb entscheidend ist. Der Nutzwert eines offenen Systems wird stark reduziert, wenn es nicht Sicherheit in Bezug auf Echtheit, Vertraulichkeit und Unversehrtheit von Daten und auch die Verfügbarkeit der Systemkomponenten garantieren kann.
Gerade die Offenheit des Systems erleichtert böswillige Angriffe auf wertvolle Informationen und Systemkomponenten, da der Zugang zum System kaum geschützt werden kann. Die Anzahl der Teilnehmer ist unübersichtlich groß, und jeder Teilnehmer könnte ein Angreifer sein. Die zu schützenden Daten sind an jedem Punkt auf den unter Umständen sehr langen Wegen durch das System potentiellen Bedrohungen ausgesetzt. Durch die gegebene Heterogenität von Hardware und Betriebssytemen können ebenso Sicherheitslücken entstehen, wie durch Uneinheitlichkeiten im Management.
Der Sicherheitsaspekt ist aus diesen Gründen bei offenen Systemen ein nicht zu vernachlässigendes Problem. Deshalb liegt es nahe, das Management der sicherheitsrelevanten Systemeigenschaften neben den übrigen Management-Aufgaben als eigenes Gebiet zu betrachten. Die Aufgabe des Sicherheitsmanagents ist es also, Schutzmöglichkeiten gegen Sicherheitsbedrohungen anzubieten, diese zu pflegen und innerhalb des offenen Systems aufeinander abzustimmen.
Der nächste Abschnitt soll einführend zunächst einen Überblick über die in einem offenen System auftretenden Gefahren und über gebräuchliche Gegenmaßnahmen geben. Anschließend wird die im OSI-Standard gegebene Organisation des Sicherheitsmanagements genauer dargelegt und die dort eingeführten vier Aufgabenbereiche beschrieben. Schließlich wird noch ein Ansatz zur Vereinheitlichung bestehender Management-Architekturen, das SAMSON-Projekt, vorgestellt.

Sicherheit in offenen Systemen

Als Grundlage für die Behandlung des Sicherheitsmanagements sollen in diesem Abschnitt einige wichtige Begriffe aus der Sicherheitstechnik eingeführt und erläutert werden.

Werte

Um auftretende Gefahren lokalisieren zu können, muß man sich zunächst über die zu schützenden Güter in einem offenen System klarwerden und die an diesen Gütern schützenswerten Eigenschaften bestimmen.
Schützenswerte Güter sind nicht nur die im System bewegten und gelagerten Daten, sondern auch die Kommunikations- und Datenverarbeitungsdienste selbst, sowie die zugrundeliegende Hardware. Zu den schützenswerten Eigenschaften, auch als abstrakte Werte bezeichnet, zählt man die folgenden:

- Authentizität
  Die grundlegende Forderung, daß alle beteiligten Komponenten "auch sind, was sie zu sein vorgeben", ist in allen bestehenden Sicherheits-Architekturen enthalten, da ohne sie andere Werte kaum schützbar wären.
- Vertraulichkeit
  Sowohl für im System abgelegte, wie auch für durch das System bewegte Daten ist es oft von essentieller Wichtigkeit, daß sie vor unbefugter Einsichtnahme geschützt werden.
- Integrität
  Von so gut wie jeder Art von Information, die dem System übergeben wird, erwartet man, daß ihre Integrität erhalten wird, d.h. daß sie nicht auf unvorhergesehene Weise verändert wird. Manche Daten sind in dieser Hinsicht besonders empfindlich und gefährdet, beispielsweise die Verwaltungsdaten des Systems. Auch für Systemressourcen ist die Eigenschaft ihrer Unversehrtheit wichtig.
- Verfügbarkeit
  Der Wert eines offenen Systems basiert auf der Verfügbarkeit der Informationsdienste. Auch die Verfügbarkeit von Daten ist zu schützen, nämlich vor unvorhergesehener Löschung.
- Nicht-Abstreitbarkeit
  Insbesondere in offenen Systemen mit ihrer Teilnehmer- und Verwendungsvielfalt ist es für manche Zwecke nötig, vergangene Kommunikationsaktivitäten unbestreitbar nachweisen zu können.

Bedrohungen

Genau entsprechend zu den eben beschriebenen Werten lassen sich nun potentielle Bedrohungen ausmachen, denen diese Werte in offenen Systemen ausgesetzt sein können:

- Bedrohung der Authentizität

  Die Bedrohung gegen die Authentizität von Benutzern durch Vorgeben einer falschen Identität bezeichnet man als Maskerade. Sie kann von Benutzern gegenüber dem System, oder auch gegenüber anderen Benutzern verübt werden. Eine andere Form eines Angriff auf die Authetizität eines Benutzers stellt die Wiedereinspielung eines beoachteten und aufgenommenen Authentisierungsvorgangs dar.

  Andersherum könnte auch die Authentizität von Systemkomponenten bedroht werden, beispielsweise durch fehlerhafte oder gefälschte Systemprogramme (z.B. Trojanische Pferde). Die meisten Sicherheitsarchitekturen verlangen aber nur die sichere Identifizierung der Benutzer gegenüber dem System, nicht auch die der Funktionsträger dem Benutzer gegenüber.

- Verlust der Vertraulichkeit

  Jede unbefugte Einsichtnahme in Daten oder laufende Kommunikation bedroht deren Vertraulichkeit. Bei Daten, die im System bewegt werden, kann eine solche Einsichtnahme überall auf dem - möglicherweise recht langen und inhomogenen - Weg zwischen den Kommunikationsendpunkten stattfinden.

  Eine Bedrohung der Vertraulichkeit ist z.B. auch das Ableiten persönlicher aus statistischen Daten durch entsprechend gestellte Anfragen an Datenbanken im offenen System.

  Auch die sogenannte Verkehrsanalyse bedroht eine spezielle Art von Vertraulichkeit: Durch die unbefugte Beobachtung von Datenpaketen - selbst wenn ihr Inhalt für den Eindringling unlesbar ist - kann zumindest die Identität und der Standort der Kommunikationsteilnehmer ermittelt werden, eventuell auch noch mehr durch Untersuchung von Nachrichtenlänge, Auslieferungszeitpunkten und Übertragungsfrequenz.

  Eventuell kann auch einfach die Wahrung einer gewissen Anonymität eine Form schon schützenswerter Vertraulichkeit sein: Wenn ein Benutzer unter verschiedenen Kennungen mit zwei verschiedenen Partner kommuniziert, sollten diese nicht unbedingt feststellen können, daß es sich um ein und dieselbe Person handelt.

- Verlust der Integrität

  Für viele Arten von Informationen ist weniger ihre Vertraulichkeit von Bedeutung, als ihre Unversehrtheit. Bedroht wird sie durch jede Form der unbefugten Modifikation.

  Die Integrität von Ressourcen kann ebenfalls gefährdet werden, z.B. durch unbefugten Zugriff auf Systemfunktionen oder durch Trojanische Pferde.

- Bedrohung der Verfügbarkeit

  Die Verfügbarkeit von Diensten kann auf vielerlei Weisen beeinträchtigt werden, sei es durch Beschädigung der Hardware, Benutzung durch unbefugte Benutzer oder durch befugte Benutzer zu unbefugten Zwecken, durch böswillige Veränderung von Systemprogrammen, Mißbrauch von Manager-Funktionen oder durch Computerviren.

- Bedrohung der Nicht-Abstreitbarkeit

  Eine solche kann z.B. auftreten, wenn über große Entfernungen Verträge abgeschlossen wurden und eine der Seiten diese Tatsache abstreiten will. In offenen Systemen ist die Nicht-Abstreitbarkeit gewisser Kommunikationsaktivitäten aus zwei Gründen von besonderer und zunehmender Bedeutung: Erstens ist es häufig der Fall, daß nicht zwei einander vertrauende Benutzer in einer nicht vertrauenswürdigen Umgebung in Verbindung treten wollen, sondern daß sich eben die kommunizierenden Benutzer gegenseitig mißtrauen. Zweitens gibt es wegen der heterogenen Struktur eines offenen

Systems keine zentrale Stelle, die das Verhalten aller Partner global kontrollieren könnte.

Nicht ganz eindeutig zuordnen lassen sich die absichtliche unerlaubte Erzeugung und Zerstörung von Daten. Die Zerstörung von Daten kann sicherlich einfach deren Verfügbarkeit bedrohen, kann aber auch als Angriff auf Integrität oder Nicht-Abstreitbarkeit verwendet werden. Mit der Erzeugung von Daten können Authentizität, Verfügbarkeit von Systemkomponenten (durch übermäßige Belastung mit unerlaubt erzeugten Daten) oder ebenfalls die Nicht-Abstreitbarkeit bedroht werden.

Dienste

Den ausgemachten Bedrohungen kann man nun wiederum zu ihrer Vermeidung geeignete Sicherheitsmaßnahmen zuordnen. Von den meisten Sicherheitsarchitekturen werden die folgenden sogenannten Sicherheitsdienste angeboten:

- Authentisierung
- Vertraulichkeit von Daten
- Integrität von Daten
- Zugriffskontrolle (zum Schutz der Verfügbarkeit)
- Nicht-Abstreitbarkeit

Je nach Sicherheitsarchitektur sind diese verschieden implementiert, und basieren jeweils auf einem oder mehreren sogenannten Sicherheitsmechanismen. Gängige Sicherheitsmechanismen sind zum Beispiel:

- Verschlüsselung
  Um die Vertraulichkeit jeglicher Art von Information zu schützen, existieren die verschiedensten Ausprägungen von Verschlüsselungsmechanismen. Man unterscheidet reversible und irreversible Algorithmen. Erstere benutzen entweder symmetrische oder asymmetrische Verschlüsselung, d.h. zum Entschlüsseln wird entweder der selbe Schlüssel benutzt wie zum Verschlüsseln, oder ein anderer. Irreversible Algorithmen können - müssen aber nicht - ebenfalls Schlüssel benutzen, die je nach Verfahren öffentlich oder privat sein können.
- Digitale Unterschriften
  Wird ein Dokument mit einer digitalen Unterschrift versehen, so kann der Empfänger einerseits den Absender eindeutig identifizieren und ihm die Absendung nachweisen, andererseits ist auch die Integrität der signierten Informationen sichergestellt. Mechanismen zur Erstellung und Verifikation digitaler Unterschriften basieren meist auf asymmetrischen Verschlüsselungsverfahren.
- Zugriffskontrolle
  In den Bereich der Zugriffskontrolle fallen Mechanismen, die vor unbefugter Benutzung von Ressourcen schützen. Realisiert wird dies beispielsweise durch Zugriffskontrollisten (Access Control Lists, ACLs) und Passwörter, Kontrolle von Zugriffszeit, -weg oder -dauer. Manche Sicherheitsarchitekturen unterscheiden weiter zwischen vom System vorgegebener (mandatory) und vom Benutzer innerhalb gewisser Grenzen frei einstellbarer (discretionary) Zugriffkontrolle.

- Integritätssicherung
  Zur Sicherstellung der Integrität von Daten kann beim Senden bestimmte Zusatzinformation - beispielsweise eine Prüfsumme - als Funktion der gesendeten Daten erzeugt und mit übertragen werden. Beim Empfang der Daten muß dann dieselbe Zusatzinformation wieder berechnet und mit der empfangenen verglichen werden. Zur Erhöhung der Sicherheit kann diese Information natürlich zusätzlich noch verschlüsselt werden. Bei Datenströmen werden außerdem Folgenummern oder Zeitstempel zur Integritätssicherung verwendet.

- Partnerauthentisierung
  Zur sicheren Identifizierung eines Teilnehmers nicht dem System, sondern seinem Kommunikationspartner gegenüber dienen Mechanismen zur Partnerauthentisierung, die auf dem Austausch identifizierender Information (Passwörter, Schlüssel) basieren.

- Verhinderung von Verkehrskontrolle
  Durch Einfügen von "Stopfpaketen" oder Auswählen immer anderer Verkehrswege können entsprechende Mechanismen Regelmässigkeiten im Verkehrsfluß verschleiern und so eine Verkehrskontrolle unterbinden.

- Routingkontrolle
  Aufgabe von Routingkontrollmechanismen ist es, bestimmte Daten nur über Teilsysteme und Verbindungsstrecken weiterzuleiten, die für diese Daten am sichersten sind, oder die Daten so über verschiedene Verkehrswege zu verteilen, daß sie unterwegs nicht mehr zusammenhängend "abgehört" werden können.

- Notar-Mechanismen
  Zwei oder mehr kommunizierende Benutzer können insbesondere zur Sicherung der Nicht-Abstreitbarkeit einen unabhängigen Notardienst im System beispielsweise zum Abschliessen von Verträgen benutzen.

Management und Sicherheitspolitik

Die genannten Sicherheitsdienste können alle in gewisser Weise zur Sicherheit eines offenen Systems beitragen, wenn sie richtig eingesetzt werden. Die Aufgabe, solche Dienste für das System und seine Benutzer zur Verfügung zu stellen und zu pflegen, d.h. sie sinnvoll zu konfigurieren, zu überwachen, Sicherheitslücken zu entdecken und sie durch Anpassung der Maßnahmen zu schließen ist von beträchtlichem Umfang. Deshalb ordnet man diese Aufgabe einem eigenen Teilbereich des Systemmanagements zu: dem Sicherheits-Management. Da ein offenes System aus vielen Teilsystemen besteht und es damit kein allen übergeordnetes Sicherheits-Management geben kann, gehört zu den Aufgaben des Sicherheits-Managements eines bestimmten Teilsystems auch die Abstimmung der Sicherheitsmaßnahmen mit denen der anderen, insbesondere angrenzender, Teilsysteme.
Eine Grundvoraussetzung für erfolgreiches Sicherheits-Management ist, daß klare Vorgaben darüber bestehen, welche Bedrohungen in welchem Außmaß und mit welchen Mitteln bekämpft werden sollen und welche Ressourcen zu schützen sind. Ein Dokument, das diese Vorgaben festlegt, ist die sogenannte Sicherheitspolitik (- eine etwas ungeschickte aber übliche Übersetzung des englischen "security policy". Besser wäre vielleicht "Sicherheitsrichtlinien" oder "Sicherheitsziele"). Wie spezifisch die Vorgaben einer Sicherheitspolitik sind, ist je nach Sicherheits-Architektur verschieden: Das Spektrum reicht von allgemeinen Kriterien bis zu recht genauen Festlegungen von Mechanismen und Algorithmen für die einzelnen Dienste.

Das OSI-Sicherheitsmanagement

Der detaillierten Beschreibung der Aufgaben des Sicherheitsmanagements eines offenen Systems, welches im letzten Abschnitt eingeführt wurde, erfolgt im zweiten Teil des Entwurfs zum OSI-Basisreferenzmodell (/ISO89/).
Dieser Entwurf soll nun genauer betrachtet werden.

Kategorien

Die OSI-Sicherheits-Architektur unterteilt die Tätigkeiten des Sicherheitsmanagements in vier Kategorien:

- Systemsicherheits-Management (system security management)
- Sicherheitsdienst-Management (security service management)
- Sicherheitsmechanismen-Management (security mechanism management)
- Sicherheit des OSI-Managements

Die Aufgaben dieser einzelnen Kategorien werden im folgenden beschrieben.

Systemsicherheits-Management

Das Systemsicherheits-Management befaßt sich mit der Sicherheit des gesamten offenen Systems. Dazu gehören typischerweise folgende Aufgaben:

- Sicherheitspolitik
  Die Notwendigkeit einer Sicherheitspolitik wurde bereits beschrieben. Die Aufgabe des Systemsicherheits-Managements ist es, eine solche zu etablieren und zu pflegen. Wenn nötig sind Anpassungen vorzunehmen. Es ist für die Konsistenz der Sicherheitspolitik zu sorgen, wozu auch die Abstimmung unter verschiedenen Teilsystemen gehört.
- Interaktion mit anderen OSI-Management-Funktionen
  Sicherheit ist nur ein Aspekt unter mehreren eines offenen Systems, die ein Management nötig haben. Andere Management-Bereiche sind meist auf Sicherheitsdienste angewiesen, um ihre empfindlichen Daten zu schützen.
- Interaktion mit Sicherheitsdienst-Management und Sicherheitsmechanismen-Management
  Sie ist z.B. nötig, um die Richtlinien der Sicherheitspolitik zu verwirklichen. Außerdem verwenden auch die Funktionen des Systemsicherheits-Managements selbst zu ihrem Schutz Sicherheitsdienste.
- Ereignisbehandlung
  Dazu gehört das Berichten über Angriffe gegen die Systemsicherheit und das Festlegen von Schwellwerten, die bestimmen, wann solche Ereignisse berichtenswert sind.

- Protokollierung
  Hier muß eine Auswahl getroffen werden, welche Ereignisse protokolliert werden sollen und das Mitloggen bestimmter Vorkommnisse ein- und ausgeschaltet werden. Die Überwachung sicherheitsrelevanter Ereignisse informiert den Sicherheits-Manager über Nutzen und Effektivität der verwendeten Sicherheits-Funktionen und gibt ihm damit die Möglichkeit, die Notwendigkeit von Anpassungen zu erkennen.
- Angriffe
  Es müssen Regeln festgelegt werden, wie auf tatsächliche oder vermutete Angriffe auf die Sicherheit zu reagieren ist. Eventuell muß automatisch an anderer Stelle Bericht erstattet werden. Aktionen zur Wiederherstellung der Sicherheit müssen ausgelöst werden können. Eventuell müssen Parameter der Sicherheitsdienste verändert werden.
- Vergabe von Zugriffsrechten
  Der Zugriff auf Management-Funktionen und -Informationen muß zur Gewährleistung eines ordnungsgemäßen Betriebs beschränkt sein. Auch dies fällt in den Aufgabenbereich des Systemsicherheits-Managements.

Sicherheitsdienst-Management

Sicherheitsdienst-Management ist das Management der einzelnen Sicherheitsdienste. Dies bedeutet zum Beispiel:

- Festlegung der Ziele der einzelnen Sicherheitsdienste
  Die Ziele eines Sicherheitsdienstes können abhängig von den Umständen unterschiedlich sein. Für jeden Dienst muß entsprechend den Vorgaben der Sicherheitspolitik vom Sicherheitsdienst-Management ein genaues Ziel bestimmt und ihm zugeordnet werden.
- Auswahl geeigneter Mechanismen
  Bei manchen Sicherheitsdiensten gibt es mehrere Möglichkeiten, geeignete Mechanismen für diese auszuwählen. Bei einigen, wie z.B. den Vertraulichkeits- und Integritätsdiensten, gibt es sogar viele unterschiedliche Möglichkeiten (verschiedene symmetrische oder asymmetrische Verschlüsselungsverfahren), die unter bestimmten Umständen bestimmte Vor- und Nachteile haben. Einen bestimmten Mechanismus auszuwählen ist die Aufgabe des Sicherheitsdienst-Managements.
- Aushandlungsprozedur zwischen Diensten im eigenen und in fremden Teilsystemen
  Da nicht jedes Teilsystem jeden Sicherheitsdienst und -mechanismus in gleicher Weise unterstützt, muß bei Kommunikationsvorgängen mit anderen Teilsystemen sichergestellt werden, daß kompatible Dienste und Mechanismen benutzt werden, die von beiden Seiten akzeptiert werden.
- Aufruf von Sicherheitsmechanismen
  Einige Sicherheitsmechanismen besitzen eigene Management-Funktionen (vgl. nächster Abschnitt), die aufgerufen werden müssen, wenn der Mechanismus benutzt werden soll. Zum Beispiel müssen bei Verwendung kryptographischer Mechanismen die verwendeten Schlüssel verwaltet werden.
- Interaktion mit anderen Sicherheitsdienst- und Sicherheitsmechanismen-Management-Funktionen

Dies ist wichtig, damit innerhalb eines Teilsystems vergleichbare Dienste angeboten werden und ein vergleichbares Sicherheitsniveau gewährleistet ist. Damit soll z.B. verhindert werden, daß mangelnde Sicherheitsmaßnahmen in einem Teilsystem die durch teure Maßnahmen erreichte Sicherheit eines angrenzenden Teilsystems untergraben.

Mechanismen-Management

Manche Mechanismen benötigen eigene Management-Funktionen, weil sie je nach benutzender Anwendung mit verschiedenen Managementinformationen versorgt werden müssen. Einige Beispiele für Mechanismen, die ein eigenes Mechanismen-Management brauchen sind die folgenden:

- Schlüsselverwaltung
  Schlüssel müssen - abhängig von den Vorgaben der Sicherheitspolitik - ab und zu neu generiert und dann im System an bestimmte Komponenten auf sichere Weise verteilt werden. Auch die Generierung und Vergabe von Schlüsseln, die auf Protokollebene zur Sicherung für die Laufzeit einer bestimmten Verbindung benötigt werden, kann durch das Management der Schlüsselverwaltung geregelt werden.
- Verschlüsselung
  Zur Verschlüsselung mit kryptographischen Funktionen ist eine Vielzahl an Parametern einzustellen; hierbei ist auch eine Zusammenarbeit mit der Schlüsselverwaltung nötig.
- Digitale Unterschriften
  Um Daten mit einer digitalen Unterschrift zu versehen, müssen Verschlüsselungsmechanismen benutzt werden. Zur Handhabung der Schlüssel wird die Schlüsselverwaltung verwendet. Management-Funktionen des Mechanismus für digitale Unterschriften müssen also auf Management-Funktionen von Verschlüsselungs- und Schlüsselverwaltungsmechanismen zugreifen. Außerdem muß ein bestimmtes Protokoll festgelegt, und eventuell die Verwendung vertrauenswürdiger dritter Parteien geregelt werden, über die beispielweise ein authentischer öffentlicher Schlüssel des Kommunikationspartners bezogen werden kann.
- Zugriffskontrolle
  Hier besteht Management-Bedarf bei der Verwaltung und Verteilung von Attributen, wie z.B. Paßwörtern, und der Führung von Zugriffskontrolllisten.
- Datenintegrität
  Wieder wird Interaktion mit der Schlüsselverwaltung, Einstellung von kryptographischen Parametern, und Benutzung eines Protokolls zwischen den kommunizierenden Instanzen benötigt.
- Authentisierung
  Beschreibende Informationen, Paßwörter oder Schlüssel müssen an die authentisierenden Instanzen verteilt werden. Auch ein Protokoll wird zur Durchführung der Authentisierung benötigt.

- Verhindern von Verkehrsanalysen
  Regeln für die Kommunikation müssen verwaltet werden, z.B. vorgegebene Datenraten. Es müssen zufällige Datenraten und Nachrichtencharakteristiken, wie z.B Nachrichtenlängen, erzeugt werden.
- Routingkontrolle
  Hier sind Informationen über vertrauenswürdige Verbindungen und Unternetze zu verwalten.
- Notarmechanismen
  Information über Notarinstanzen im offenen System sind zu verwalten und ein Protokoll zur Kommunikation zwischen Kommunikationspartnern und Notar oder zwischen Notaren ist einzuhalten.

Management-Sicherheit

Die Sicherheit aller Netzwerkmanagement-Funktionen und -Parametern, sowie der übertragenen Management-Informationen ist unverzichtbar für die Sicherheit des Gesamtsystems. Es ist die Aufgabe der Management-Sicherheit, durch Verwendung geeigneter Sicherheitsdienste und -mechanismen für angemessenen Schutz der anderen Management-Funktionen zu sorgen, sowie auch für den eigenen. Der OSI-Sicherheitsstandard geht dabei allerdings davon aus, daß die Endsysteme selbst hinreichend sicher sind. Entsprechender Schutz der Endsysteme kann also noch zusätzlich nötig werden.

Sicherheitsmanagement-Datenbasis

Die beim Sicherheitsmanagement anfallenden Einstellungen, Parameter und sonstigen Informationen müssen auf sichere Weise gespeichert werden. Hierfür wird eine spezielle Sicherheitsmanagement-Datenbasis verwendet, die der Natur von offenen Systemen entsprechend natürlich verteilt beschaffen sein muß und teilweise in der allgemeinen Management-Datenbasis des Systems integriert sein kann.
Im OSI-Entwurf des Sicherheits-Managements wird die Datenbasis mittels des X.500-Verzeichnisdienstes realisiert, der selbst geeignete Sicherheitsmaßnahmen zum Schutz der Daten anbietet.

Das SAMSON-Projekt

Die Entwicklung einer Vielzahl verschiedener Sicherheitsarchitekturen und einzelner Sicherheitsdienste hat dazu geführt, daß ein Sicherheits-Manager sich mit ebensovielen jeweils eigenen Benutzerschnittstellen auskennen muß, was leicht zu Inkonsistenzen und damit zu Sicherheitsproblemen führen kann.
Das Ziel des SAMSON-Projekts (/LEC94/) (Security And Management Services in Open Networks), eines Unterprojekts des RACE-Projekts der Europäischen Kommission, an dem sich einige europäische Firmen beteiligen, ist es, eine einheitliche Sicherheitsmanagement-Schnittstelle für möglichst viele verschiedene Sicherheitsarchitekturen zu entwickeln und zu implementieren. Erreicht werden soll dies dadurch, daß möglichst viele bestehende Implementationen von Sicherheitsfunktionen untersucht und ihre gemeinsamen Eigenschaften

extrahiert werden. Nach der Aufstellung eines abstrakten Modells für jede Sicherheitsfunktion wird eine einheitliche Management-Funktion für diese entworfen. Sogenannte "Sponsor-Funktionen" übernehmen dann die Vermittlung zwischen der Manager-Schnittstelle und den Schnittstellen der bestehenden Implementationen. Eine große Zahl generischer Sicherheitsmanagement-Funktionen ist bereits entworfen, Sponsor-Funktionen befinden sich in Entwicklung.

Unterstützte Sicherheitsdienste

Um den Nutzen des Verfahrens zu demonstrieren wurden zunächst die Sicherheitsdienste Authentisierung, Zugriffskontrolle, Protokollierung und Schlüsselverwaltung analysiert, die die Grundbedürfnisse der Sicherheitsanforderungen in einem Netzwerk abdecken. Vertraulichkeit und Unversehrtheit von Daten wird dabei über die Schlüsselverwaltung erreicht, die sichere Assoziationen ermöglicht.
Für jeden der genannten Dienste gibt es schon mehrere Implementierungen, von denen so viele wie möglich durch die Vereinheitlichung erfaßt werden sollen.
Das grundsätzliche Vorgehen ist wie folgt: Zunächst wird ein abstraktes Modell für einen Sicherheitsdienst entworfen, indem Datenklassen bestimmt werden, die für jede Implementation des Dienstes nötig sind. Dieses Modell wird dann darauf hin untersucht, welche Management-Funktionen zu seiner Handhabung gebraucht werden.

Authentisierung

Das ISO-Rahmendokument zur Authentisierung (/ISO91a/), an das sich die meisten bestehenden Implementierungen halten, verwendet bei der Authentisierung zwei Klassen von Daten: die Anwärterdaten (claimant authentication information) und die Verifikationsdaten (verification authentication information). Zur Verwaltung sind Funktionen zum Hinzufügen (neue Benutzer), Ändern (z.B. zeitweiliges De-/Reaktivieren eines Benutzers), Löschen und Anzeigen solcher Daten nötig - unabhängig von ihrer jeweiligen genauen Form. Eine einheitliche Schnittstelle zur Authentisierung muß also diese Management-Funktionen in generischer Form zur Verfügung stellen.

Zugriffskontrolle

Hier unterscheidet man zwischen identitäts- und regelbasierten Verfahren. Identitätsbasierte Verfahren verwenden Zugriffskontrollisten, regelbasierte verwenden Mengen von Regeln, die sich nicht auf Identitätsabfragen beschränken müssen, sondern beliebige andere Kriterien wie Tageszeit oder Status des Benutzers miteinbeziehen können. In beiden Fällen sind jedoch wieder Verwaltungsfunktionen zum Einfügen, Verändern, Löschen und Anzeigen von ACL-Einträgen bzw. Regeln nötig, die generisch implemetiert werden können.

Schlüsselmanagement

Die zu verwaltende Datenbasis besteht beim Schlüsselmanagement z.B. aus Schüsseln, Schlüsselzertifikaten und Gültigkeitszeiträumen.

Protokollierung

Da es zur Gewährleistung der Sicherheit eines Netzwerkes unerläßlich ist, alle sicherheitsrelevanten Ereignisse zu überwachen, ist die entstehende Datenmenge unüberschaubar groß. Das Protokollierungs-Management muß Möglichkeiten anbieten, hier sinnvolle Einstellungen bezüglich Auswahl und Filterung solcher Daten, sowie Reaktionen auf Verletzungen zu wählen und laufend geeignet anzupassen. Die einheitliche Schnittstelle dient also zur Einstellung solcher Parameter.

Sicherheitsrichtlinien

Damit die verschiedenen in einem System eingesetzten Sicherheitsdienste sich optimal ergänzen und nicht etwa sehr niedrige Sicherheitsanforderungen bei einem Dienst die hohen Anforderungen eines anderen sinnlos machen, ist auch für die Festlegung von allgemeinen Sicherheitsrichtlinien eine einheitliche Schnittstelle wünschenswert. Sie muß die Möglichkeit bieten, die einzelnen Parameter der verschiedenen Dienste auf Konsistenz zu überprüfen.

Beispiele

Im folgenden soll beispielhaft an einigen Sicherheits-Implementierungen gezeigt werden, inwieweit sie unter die einheitliche Management-Schnittstelle des SAMSON-Projekts integrierbar sind.

OSF DCE

Diese von der Open Software Foundation entwickelte Umgebung für verteilte Anwendungen (DCE: Distributed Computing Environment) besteht aus einer Reihe von Diensten, die auf dem entfernten Prozeduraufruf (Remote Procedure Call) basieren.
Die Authentisierung erfolgt in DCE mittels Paßwörtern. Die benötigten Authentisierungsdaten werden vom zentralen Registrierungsdienst der Umgebung gespeichert. Da dieser eine eigene Schnittstelle hat, können Sicherheitsmanagement-Funktionen die Daten problemlos manipulieren.
Ähnlich einfach gestaltet sich das Management der Zugriffskontrolle: DCE verwaltet hier Zugriffskontrollisten mit Namen von Benutzern oder Anwendungen.
Probleme ergeben sich beim Schlüsselmanagement: Zur gesicherten Datenübertragung bietet DCE den sogenannten authentisierten Prozeduraufruf an, der symmetrische Verschlüsselung mit Hilfe eines benutzereigenen Schlüssels verwendet. Die Schlüssel werden jedoch von den Authentisierungs- und Zugriffskontrollmechanismen implizit vergeben, damit kann also von außen kaum eine Beeinflussung stattfinden.
Auch die Möglichkeiten der Protokollierung sind sehr beschränkt. Es gibt keine zentrale Sammelstelle für Protokollierungsdaten, was die sinnvolle Überwachung sicherheitsrelevanter Ereignisse sehr erschwert.
DCE unterstützt eine gewisse Menge von Sicherheitsrichtlinien-Parametern, wie z.B. Gültigkeitszeiträume und Mindestlängen für Paßwörter. Diese Parameter werden ebenfalls vom zentralen Registrierungsdienst verwaltet und können damit vom Sicherheitsmanagement eingestellt werden.

SESAME

Eine weitere Sicherheits-Architektur für verteilte Systeme ist die des SESAME-Projekts (Secure European System for Application in a Multivendor Environment).
Authentisierung und Zugriffskontrolle von SESAME lassen sich problemlos managen: Zur Authentisierung werden einwegverschlüsselte Paßwörter benutzt. Mit Hilfe seiner authentisierten Identität kann ein Benutzer dann über einen bestimmten Dienst Zugriffsrechte anfordern. Deren Vergabe erfolgt über Zugriffskontrollisten, die nicht nur Namen sondern auch bestimmte Rollen enthalten können.
Für das Schlüsselmanagement ist bei SESAME im Unterschied zu DCE eine eigene Schnittstelle vorgesehen, die es erlaubt, kryptographische Parameter, Algorithmen und Schlüssel für die zum Schutz der Unversehrheit von Datenelementen eingesetzten digitalen Unterschriften einzustellen.
Auch ein eigener Protokollierungsdienst, der Daten von allen anderen Sicherheitsdiensten erhält, wird von SESAME angeboten. Dieser bildet eine Grundlage für sinnvolle Überwachung, wenn auch die Filterung der Daten noch extern vorgenommen werden muß.

X.500-Verzeichnisse

Zum CCITT/ISO-Standard X.500 (/ISO92a/) für einen Allzweck-Verzeichnisdienst existiert ein eigener Entwurf (X.509) für Authorisierung und Zugriffskontrolle.
Dabei werden zwei Varianten der Authentisierung angeboten: einfache und starke. Die letztere basiert auf Schlüsselzertifikaten mit dem öffentlichen Schlüssel eines Benutzers, deren Unversehrtheit durch die digitale Unterschrift der Zertifikations-Autorität garantiert wird.
Aufgaben für das Sicherheits-Management sind hier z.B. die Erzeugung solcher Zertifikate für neue Benutzer, deren Rücknahme oder die Änderung des Schlüssels für die digitale Unterschrift der Zertifikations-Autorität.
Zur Zugriffskontrolle werden wieder Zugriffskontrollisten verwendet, deren Pflege vom Sicherheitsmanagement übernommen werden kann.
Zum X.500-Verzeichnisdienst gibt es bereits eine standardisierte Schnittstelle, das X Open XDS interface, das alle zur Verwaltung von Authorisierung und Zugriffskontrolle nötigen Operationen zur Verfügung stellt.

Sicherheitsdienste auf tieferen Schichten

Als Beispiele für Sicherheitsdienste, die nicht in der Anwendungsschicht des OSI-Referenzmodells angesiedelt sind, seien noch zwei Protokolle für Transport- und Vermittlungsschicht genannt: TLSP (/ISO92b/) und NLSP (/ISO92c) (Transport bzw. Network Layer Security Protocoll).
Sie gewährleisten Vertraulichkeit und Unversehrtheit, indem sie die Daten eines jeden Pakets symmetrisch verschlüsseln oder mit einer digitalen Unterschrift versehen. Der unbedingt erforderliche Schlüsselverwaltungsdienst erfordert wieder Management-Funktionen zum Einstellen von Parametern.

Zusammenfassung

Es hat sich gezeigt, daß die Aufgabe, in offenen Systemen für Sicherheit zu sorgen, so wichtig und komplex ist, daß ein eigenes Sicherheitsmanagement als Teil des Systemmanagements erforderlich ist.
Der vorgestellte ISO-Rahmenentwurf zum Sicherheitsmanagement in OSI-Systemen teilt das Gebiet des Sicherheitsmanagements weiter in vier Kategorien auf und beschreibt recht detailliert die Aufgaben der einzelnen Kategorien.
Allerdings bezieht sich der Standard eben nur auf die OSI-Umgebung und läßt z.B. das Gebiet des Open Distributed Processing aus, welches jedoch teilweise zum Sysemmanagement eingesetzt wird und damit einen besonderen Sicherheitsbedarf hat. Auch die Sicherheit in den Endsystemen, z.B. der Schutz der Management-Werkzeuge gegen Mißbrauch, wird im Standard nicht behandelt.
Das im zweiten Teil dieser Arbeit vorgestellte SAMSON-Projekt zielt darauf ab, ein Werkzeug zum Sicherheits-Management zu schaffen, das es erlaubt, unter einer einheitlichen Oberfläche die verschiedenen Implementierungen von Sicherheitsdiensten in einem heterogenen Netzwerk anzusprechen.
Von der auf diese Weise erfolgten Vereinheitlichung erwartet man zweierlei:

- Vereinfachung
  Die Überwachung und Pflege der Netzwerksicherheit vereinfacht sich für den Administrator wesentlich, da er nicht mit vielen unterschiedlichen Management-Schnittstellen vertraut sein muß. Die Wahrscheinlichkeit von Fehlern und Inkonsistenzen wird dadurch und durch die Mitverwaltung von Sicherheitsrichtlinien verringert, die Sicherheit des Netzes erhöht.

- Erweiterbarkeit
  Neu hinzukommende Sicherheitsdienste können auf einfache Weise eingebunden werden, indem eine entsprechende Sponsor-Funktion implementiert wird.

Das Vorhaben, eine gemeinsame Oberfläche für verschiedene Sicherheits-Implementierungen zu schaffen, ist sicherlich zu begrüßen. Ob die dabei nötige Allgemeinheit nicht doch wieder zu Lasten der detailgenauen Konfigurierung der einzelnen Dienste fällt, muß der sich in Arbeit befindliche Prototyp zeigen.

Literatur

/GRI94/ Rüdiger Grimm
Sicherheit für offene Kommunikation: verbindliche Kooperation
BI-Wiss.-Ver., 1994

/ISO89/ ISO
ISO 7498-2: Information Processing Systems - Open Systems Interconnection Basic Reference Model, Part 2: Security Architecture
Genf (1989)

/ISO91a/ ISO
ISO DIS 10181-2: Authentication Framework
Genf (1991)

/ISO91b/ ISO
ISO CD 10181-3: Access Control Framework
Genf (1991)

/ISO91c/ ISO
ISO CD 10181-7: Audit Framework
Genf(1991)

/ISO92a/ ISO
ISO 9594: Draft revised Recommendations: The Directory
Genf (April 1992)

/ISO92b/ ISO
ISO DIS 10376: Transport Layer Security Protocol (TLSP)
Genf (1992)

/ISO92c/ ISO
ISO DIS 11577: Network Layer Security Protocol (NLSP)
Genf (1992)

/LEC94/ Stephan Lechner
SAMSON: Management of security in open systems
computer communications vol.17, Juli 1994

/MUF92/ Sead Muftic
Sicherheitsmechanismen für Rechnernetze
Prentice-Hall Internat., 1992

/PAT94/ Ahmed Patel
Security Management for OSI networks
computer communications vol.17, Juli 1994

/SHS94/ Steven L. Shaffer, Alan R. Simon
Network Security
Academic Press Inc., 1994

OBJECT-MANAGEMENT

Juan A. Uriarte

Neue Ansätze für ein übergreifendes Netzwerk- und Systemmanagement gehen von einem objektorientierten Modell aus. Es wird hier am Beispiel von CORBA (und dem darauf aufbauenden Distributed Management Environment der Open Software Foundation) erläutert, welche Vor- und Nachteile solche Managementarchitekturen haben. Anschließend wird ein Blick in die Zukunft geworfen.

Motivation

Im Moment ist die Konnektivität zwischen heterogenen Systeme gegeben. Primitive Dienste, die auf dieser einfachen Form der Kommunikation aufbauen, wie z.B. Dateitransfer oder entferntes Einloggen, sind standardisiert und weit verbreitet. Auf einer höheren Ebene gibt es auch Standards für „Remote Procedure Call“–Systeme , wie z.B. ONC-RPC oder DCE-RPC. RPC Systeme bilden idealerweise die Bausteine für verteilte Anwendungen, doch leider gibt es wegen mangelnder eindeutiger Standardisierung auf dieser höheren Ebene keine weit verbreiteten verteilten Anwendungen. Eine Netzwerkmanagementanwendung, die ihre Arbeit über RPC erledigt, wird sehr unwahrscheinlich mit einer anderen Netzwerkmanagementanwendung von einem anderen Hersteller zusammenarbeiten, obwohl beide womöglich denselben RPC–Standard benutzen.

Object Management Architecture

Um eine Hersteller–übergreifende Lösung für dieses Problem im Rahmen der neuen Technologie zu finden, bildete sich die Object Management Group (OMG). Sie soll auf der Grundlage des objektorientierten Modells auf verteilter Basis eine transparente Interoperabilität heterogener Systeme ermöglichen.

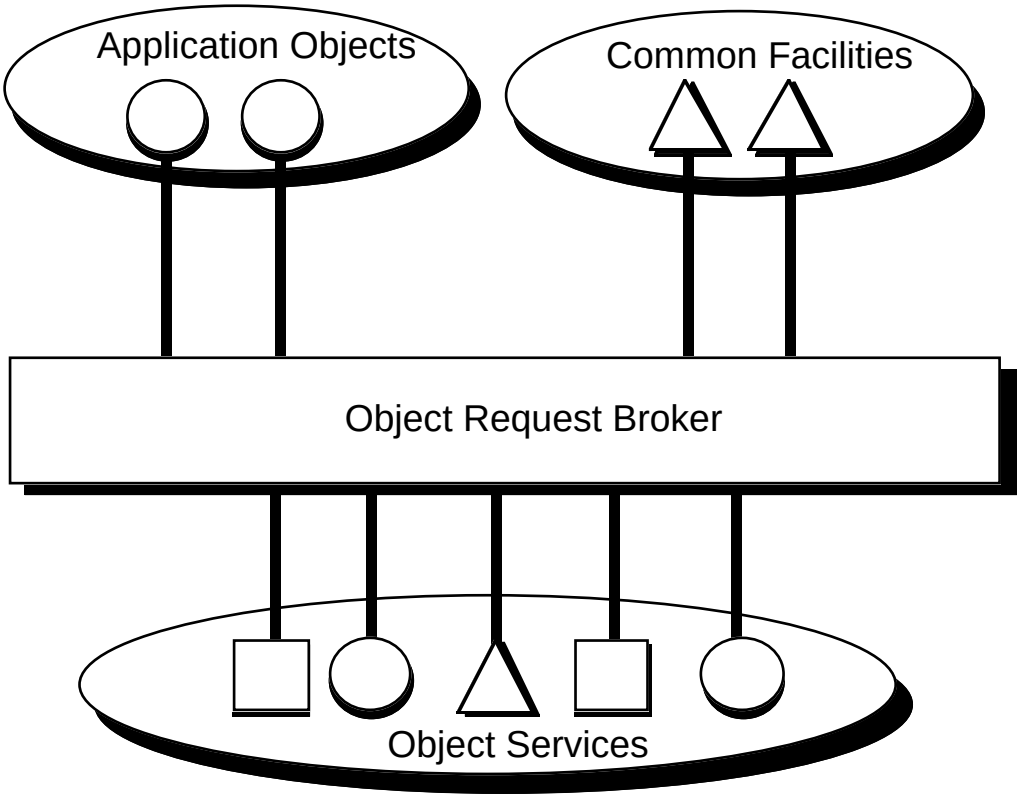


Abb. 1. Object Management Architecture

Das erstes Produkt dieser Gruppe war die Object Management Architecture (OMA) (Abb. 1.). Die OMA gliedert sich in vier Teile: die Application Objects, die Object Services, die Common Facilities und den Object Request Broker (ORB).

Die *Object Services* bieten die Basisfunktionen zur Verwaltung der Objekte im Netzwerk, insbesondere zur Erzeugung und Verwaltung von Klassen und Instanzen, und auch bei Bedarf die Möglichkeit, persistente Objekte zu erzeugen. Vereinfachend kann man sagen, sie bieten die Operationen, die objektorientierte Programmiersprachen lokal anbieten, auf eine von der Konzeption heraus verteilte Weise.

Die *Common Facilities* sind eine Art Bibliothek. Sie sollen Funktionen anbieten, die generellen Charakter haben (z.B.: drucken oder Email) und dadurch die Anwendungsentwicklung enorm erleichtern und beschleunigen. Es soll möglich sein, durch einfaches Ableiten dieser Basisklassen leistungsvolle Anwendungen zu schreiben, ohne jedesmal das Rad neuentwickeln zu müssen. Insbesondere wird sich durch die Standardisierung der Schnittstellen auf dieser hohen Ebene ein Software–Markt eröffnen. Es wird dann möglich sein, wegen der festgelegten Schnittstellen, die verschiedene Teile der OMA von verschiedenen Herstellern zu kaufen. Diese Produkte werden dann problemlos miteinander zusammenarbeiten können.

Die *Application Objects* sind die eigentliche Anwendungsobjekte, die die Funktionalität der Anwendung realisieren.

Der *Object Request Broker (ORB)* spielt die zentrale Rolle der OMA (Abbildung 2):

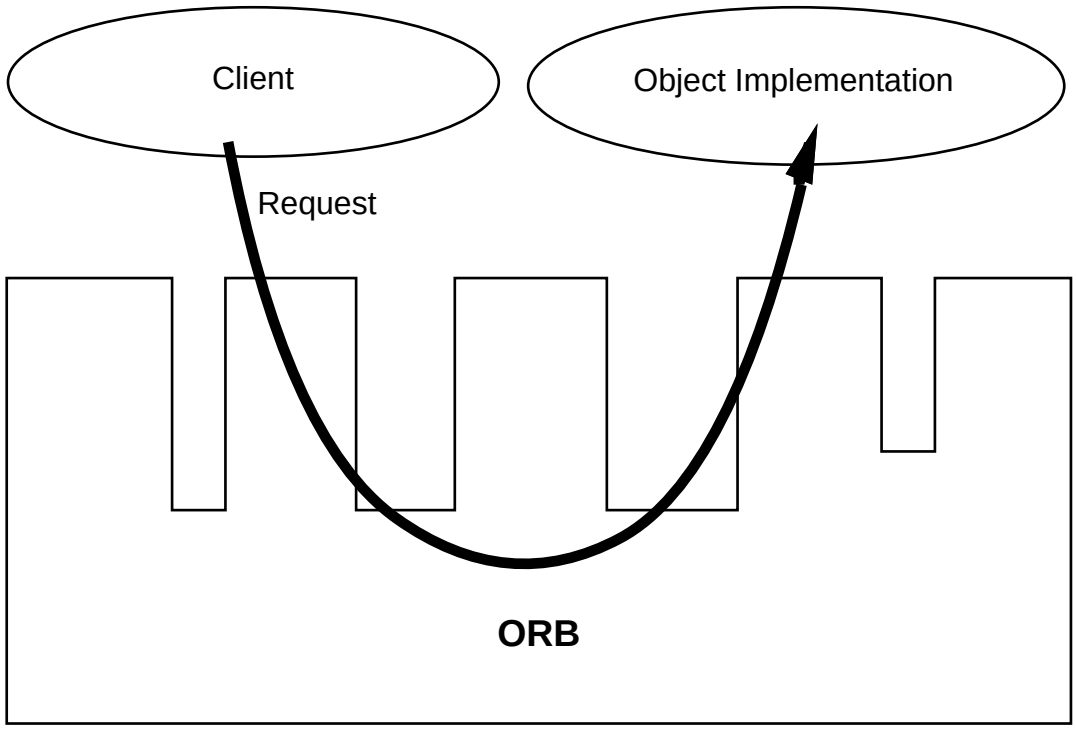


Abb. 2: Arbeitsweise des Object Request Brokers

Die Aufgabe des ORB ist es, Requests von Objekten transparent über das Netzwerk zu leiten. Transparent ist hierbei auf zwei Arten zu verstehen: erstens braucht der Aufrufer nicht zu wissen, auf welcher Maschine im Netzwerk sein Kommunikationspartner lokalisiert ist, und zweitens muß der Klient nicht unbedingt wissen, auf welchem Maschinentyp der Server läuft oder sogar, in welcher Sprache dieser implementiert ist. Die Argumentübersetzung wird automatisch durch den ORB erledigt.

Dazu bietet der ORB eine Reihe von Diensten:

- einen *Name Service*, der eine eindeutige Benennung der Objekte im Netzwerk ermöglicht.
- einen *Request Dispatch Service*, der für jedes Request die zugehörigen Objekte und Methode(n) findet.
- das *Parameter Encoding*, das, wie bei RPC-Systemen üblich, Parameter auf ein Maschinen-unabhängiges Format kodiert und dann diese auf spezifische Maschinen- und Sprachkonventionen zurückdekodiert.
- *Delivery Services* zur korrekten Zustellung von Requests und Ergebnissen.
- *Activation* für die Verwaltung von persistenten Objekten. Vor einen Aufruf muß erstmals das Objekt aktualisiert und nach dem Aufruf dessen Status gesichert werden.
- *Exception Handling*, d.h. die Dienste, die für die Behandlung von Ausnahmesituationen zuständig sind (Ausfall von einem Knoten, Ausfall des gesamten Netzwerks, Speichermangel, usw.).
- die *Security Services*, die für die Authentifikation von Server und Klient dienen und für die darauf aufbauende Authorisierung des Requests.

Interface Definition Language

Ein anderer wichtiger Bestandteil von CORBA ist die Interface Definition Language (IDL). Diese Schnittstellenbeschreibungssprache dient für eine Zielsprache-unabhängige Definition von Attributen, Ausnahmensituationen (*exceptions*), Typen, Konstanten und Methodenschnittstellen. Sie wird dann durch sogenannte ´Language Mappings´ auf eine Zielsprache eindeutig abgebildet. Im Moment sind Language Mappings für C definiert, mit der Version 2.0 von CORBA sollte dann das Language Mapping für C++ festgelegt werden. Es sind auch Language Mappings für Ada, Smalltalk und COBOL im Kommen.

IDL ist im Grunde genommen C++ mit einigen kosmetischen Änderungen und Erweiterungen, die für die Beschreibung von verteilten Schnittstellen notwendig sind:

- Es besteht die Möglichkeit, bei der Definition von Methoden die Parameter als Ein-, Aus- oder Einausgabeparameter festzulegen.
- Was in der C++ Welt „class“ heißt, wird hier „interface“ genannt.
- Es gibt das neue Schlüsselwort „attribute“, es bezeichnet einen les- und/oder schreibbaren Wert. Es ist ein rein syntaktisches Konstrukt, denn es definiert automatisch zwei Funktionen zum Setzen und Lesen des Attributs. Die letzte Zeile in Beispiel 1 hätte somit auch wie folgt heißen können:

```
rstate_t    get_router_status ();
void        set_router_status( in rstate_t state);
```

Beispiel 1:

```
interface generic_router {
// Typen, die für den Zugriff auf den Router gebraucht werden.

      enum  state_t   {     up,     down     };      //   Interface
Status
      enum  rstate_t    { up, rebooting };     // Zustand des Routers

struct      rtable_entry  {                             //  Eintrag  in
Routingtabelle
      generic_address   dest_addr;             // Zieladresse
      generic_address   src_addr;              // Quelleadresse
      generic_if_name   interface_name;   // Interface
      state_t                       link_state;            //  Status  des
Links
};

struct      statistic_entry  {                          //  Eintrag  in
Statistiktabelle
      generic_if_name   interface_name;   // Name des Interfaces
      counter_t               in_packets;            // Empfangene Pakete
      counter_t               out_packets;           // Gesendete Pakete
      counter_t               err_packets;           // Fehlerhafte Pakete
      time_t                              up_time;                    //
Zeit seit letztem Reset
};

// Funktion zum Setzen eines Routingeintrags im Router
      error_t     SetRoutingEntry {
      in    generic_if_name         interface_name;
      in    generic_address         dest_addr;
      in    generic_address         src_addr;
};

// Funktion zum Lesen eins Routingeintrags für ein Interface
      rtable_entry      GetRoutingEntry {
      in    generic_if_name   interface_name;
};

// Lesen der Statistikdaten eines Interfaces
      statistic_entry   GetStatistic {
      in    generic_if_name   interface_name;
};

// Zurücksetzen der Statistikdaten
      error_t     ResetIfStatistic {
      in    generic_if_name   interface_name;
};

// Lesen/Schreiben des Routerzustandes
```

```
      attribute   rstate_t    router_status;

// Ist nur eine syntaktische Konvention für ein
// Funktionenpaar zum Setzen und Lesen des Attributs

};
```

Der Code im Beispiel 1 ist die Beschreibung für eine einfache Schnittstelle zur Verwaltung von Routern, die deren Zustände und Operationen auf eigene, modellierte Zustände abbildet. Er bietet Funktionen zum Lesen und Setzen von Routingtabellen–Einträgen, und zum Lesen und Zurücksetzen der mitgeführten Statistik. Er soll ein Beispiel für eine exportierte Managementschnittstelle sein.

Netzwerkmanagementanwendungen können alle Router, die diese Schnittstelle benutzen, verwalten. Angenommen, Hersteller X will einen Router mit einem speziellen Feature bauen: er soll ein Lüfter haben. Er will aber, daß er weiterhin kompatibel mit bestehenden Managementanwendungen bleibt. Die folgende Definition der Schnittstelle würde diese Kompatibilität gewährleisten:

```
interface   X_Router: generic_router

      enum fan_state { on, off };

      attribute fan_state_t fan_state;
};
```

Damit kann die alte Netzwerkmanagementanwendung problemlos den neuen Router verwalten, und es ist auch möglich, daß die neuen Eigenschaften des Routers mittels des dynamischen Aufrufens (z.B. durch ein Dialogfenster mit dem Benutzer) benutzt werden.

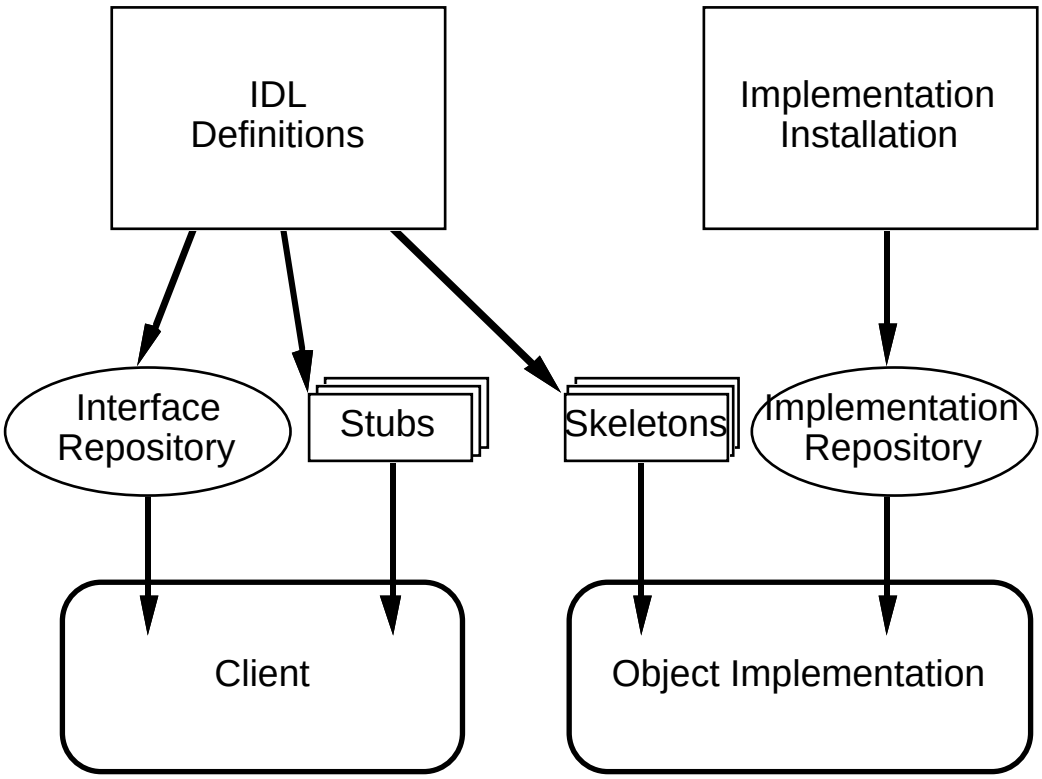


Abb. 3: IDL-Übersetzungsvorgang

Abb. 3 zeigt den IDL–Übersetzungsvorgang. Dort ist zu erkennen, wie aus der IDL-Definition folgende Komponenten erzeugt werden:

- *Stubs* in der Zielsprache zum Binden bei den Klienten. Wie aus der RPC Welt bekannt ist, sind diese Funktionen nur ´hohle´ Funktionen. Sie bieten die eigentliche Schnittstelle zum Klienten.
- *Skeletons* in der Zielsprache zum Binden bei den Servern. Diese Funktionen sind die Callbacks zur eigentlichen Methodenimplementation.
- Persistente Speicherung der Schnittstellendefinitionen im *Interface Repository*. Das ermöglicht eine Abfrage der Schnittstellen zur Laufzeit.

CORBA Schnittstellen

Noch ein wichtiger Standardisierungspunkt sind die Schnittstellen zum Object Request Broker (Abb. 4). Ein herausragendes Merkmal sind die zwei Arten, Methoden aufzurufen:

- Erstens über die mitkompilierte, statische Schnittstelle (dunkelgrau in Abb. 4), die vom IDL Compiler erzeugt wurde.
- Zweitens dynamisch zur Laufzeit durch Schnittstellenanfragen an das Interface Repository (schraffiert im Abb. 4).

Die Schnittstelle zum ORB ist auch festgelegt, obwohl die eigentlich internen Schnittstellen vom ORB zur Außenwelt (schwarz in Abb. 4) nicht näher definiert werden. Der Grund dafür ist, daß verschiedene ORB–Implementierungen unterschiedliche Zugangsmethoden brauchen

(eine verteilte objektorientierte Datenbank hat andere Ansprüche und bietet andere Dienstzugangsmethoden als ein einfacher verteilter Broker).

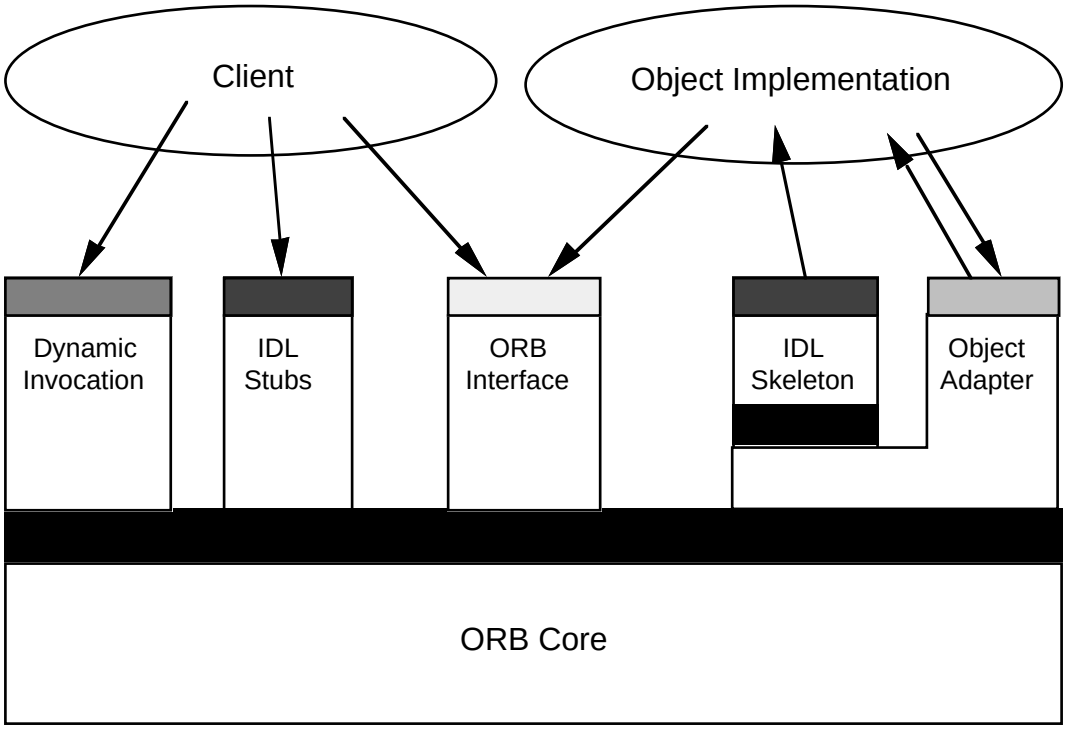


Abb. 4: Die Struktur der Object Request Broker Interfaces

Basic Object Adapter

Der Basic Object Adapter (BOA) dient primär dazu, Objekte administrativ ins CORBA-Rahmenwerk zu integrieren. Er bietet insbesondere die folgende Funktionen:

- Erzeugung und Interpretation von eindeutigen Objektreferenzen.
- Identifizierung und Authentisierung der jeweils aufrufenden Klienten.
- Aktivierung und Deaktivierung der Implementationen. Das spielt eine wichtige Rolle bei der Benutzung von persistenten Objekte.
- Und das Wichtigste: der eigentlichen Methodenaufruf mit Hilfe des vom IDL-Compiler generierten Rumpfes.

Ein Beispielrequest wäre gemäß Abb. 5 wie folgt:

- Ein Klient hat eine Referenz auf ein Objekt, in diesen Fall ein persistentes Objekt. Die Referenz wird benutzt, d.h. das Klientobjekt ruft eine Methode des referenzierten Objektes auf.
- Durch den ORB wird das Serverobjekt lokalisiert und der Request and den dortigen BOA weitergeleitet.
- In diesem Fall war der Objektserver nicht aktiv, der BOA muß ihn erstmals aktivieren (Punkt 1 in Abb. 5)
- Das Serverobjekt meldet sich beim BOA, wenn es bereit ist, Requests für eine bestimmte Schnittstelle zu akzeptieren.
- Der BOA aktiviert dann die nötige Objektinstanz, die somit bereit ist, die gewünschte Methode auszuführen.

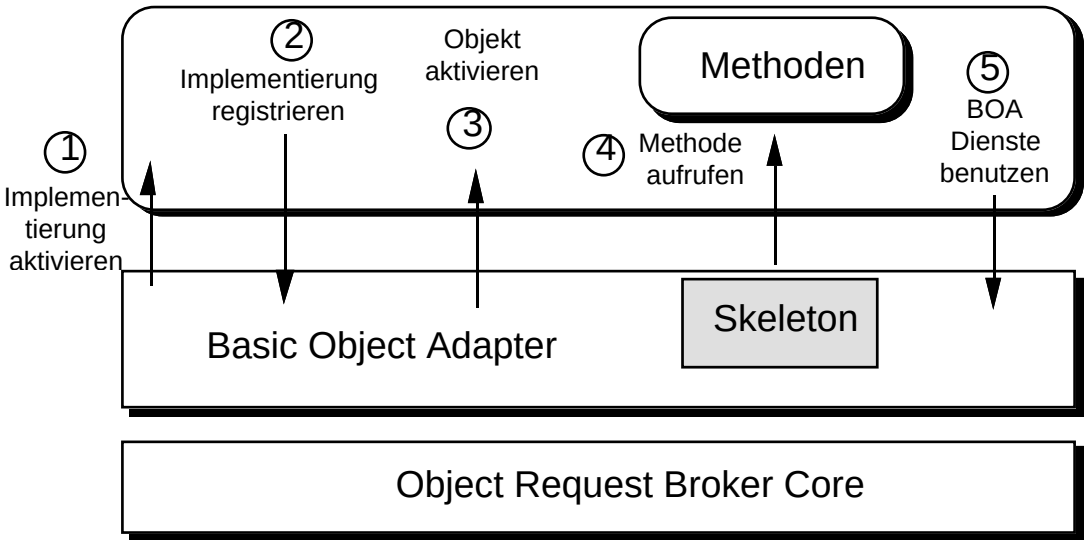


Abb. 5: Struktur und Funktionsweise eines Basic Object Adapters

Durch den BOA werden mehrere Formen der Methodenaktivierung unterstützt:

- *Mehrere Schnittstellen pro Server.*
  Ein einziger Prozeß verwaltet alle Objektinstanzen von einer bestimmten Schnittstelle.
- *Eine Schnittstelle pro Server.*
  Ein Prozeß betreut alle Instanzen einer Klasse.
- *Ein Objekt pro Server.*
  Eine Objektinstanz einer Klasse wird durch genau einen Prozeß implementiert.
- *Mehrere Server pro Objektinstanz.*
  Jede Methode einer Instanz einer Klasse wird durch einen separaten Prozeß realisiert.

Distributed Management Environment (DME)

Das DME baut auf dem Distributed Computing Environment (DCE) der Open Software Foundation auf. Es ist eine Obermenge der CORBA–Spezifikation, aber rückwärtskompatibel zu dieser. Es erweitert auch IDL um Netzwerkmanagement–Paradigmen, wie z.B. asynchrone Events, die somit konsistent ins Rahmenwerk integriert werden konnten. Die Kommunikation zwischen den ORBs wird über DCE-RPC erledigt.

Es wurde alles als Objekte modelliert, das soll eine einfachere Erweiterung ermöglichen. Insbesondere wurde auch die XMP (X/Open Management Protocols API) mitintegriert. Es wird ferner eine syntaktisch gleiche, aber semantisch unterschiedliche Enkapsulation von SNMP und CMIP definiert. So wird der Zugang zu den wichtigsten Managementprotokollen gewährleistet. Das soll auch insbesondere die Wiederverwendung von Quellcode vereinfachen.

Zur Integration von kleineren Rechnerarchitekturen (PCs) wird ein Mechanismus festgelegt, welcher durch die Verwendung von vereinfachten Klienten auf dem PC und sogenannten Proxy–Servern eine Integration dieser Systeme ermöglicht, ohne auf den betroffenen Rechnern das ganze DCE–Substrat laufen zu lassen, was eine zu große Auslastung der begrenzten Ressourcen dieser Architekturen zur Folge hätte.

Der ISO/POSIX–Standard zum Drucken ist auch Teil der DME. Somit wird ein Wandeln von lokalen, System-spezifischen Druckermechanismen in einen systemunabhängigen Netzwerk-druckdienst ermöglicht.

Es besteht auch die Möglichkeit, die Darstellungskomponente der Anwendung (Dialog-definitionen und die Aktionen, die auf diese Dialogaktionen folgen) zu speichern und mit Objekten zu verbinden. Damit wird das Einhalten einer einheitliche Schnittstelle bei der Managementumgebung vereinfacht.

Durch diese und andere Mechanismen (Management von Softwareverteilung ist auch Teil des DME, sowie globales Mitprotokollieren und Filtern von Ereignissen) wird erhofft, eine System-übergreifende Integration von *allen* Systemen in die Managementumgebung zu erreichen, von Großrechnern bis hin zu Personal Computers.

Zukunftsaussichten

CORBA ermöglicht einen Wandel des getrennten, auf einem einzigen Betriebssystem zentrierten Systemmanagements und protokollzentrierten Netzwerkmanagements hin zum integrierten heterogenen Netzwerkmanagement. Es wird auch einen Wandel von monolothischen Anwendungen zu komponentenweise zusammengesetzten Anwendungen bewirken. Das wird aber nur dann geschehen, wenn sich ein Standard für die Schnittstellen einer höheren Ebene herauskristallisiert, und dieser von viele Firmen implementiert wird. Ein Bereich, wo dieser Standard schon existiert, ist das Netzwerkmanagement, wie es im DME–„white paper“ vorgestellt wurde.

Neue Märkte für Teilkomponenten von umfassenden Managementanwendungen werden erzeugt. Das wird eine breite Öffnung im Bereich Software nach sich ziehen, und eine Senkung der Preise, da Firmen gegeneinander in diesem Bereich konkurrieren.

Literatur

/Dig93/ Digital Equipment Corporation; Hewlett-Packard Company; HyperDesk Corporation; NCR Corporation; Object Design, Inc. ; Sunsoft, Inc.
The Common Object Request Broker: Architecture and Specification,
29. Dezember 1993. Revision 1.2

/MZ93/ T. Mowbray und R. Zahavi
Distributed Computing with Object Management
ConneXions – The Interoperability Report 7(12),
Dezember 1993, Seite 18 – 24.

/Ope92/ Open Software Foundation.
OSF Distributed Management Environment (DME) Architecture,
Mai 92.

/iX93/ Torsten Beyer
Objektbörse
iX Multiuser Multitasking Magazin, Februar 1993

BREITBAND-NETZWERKMANAGEMENT

Michael Fuchs

Beim Management in Hochgeschwindigkeits-Netzwerken zeigt sich zunehmend, daß konventionelle Ansätze immer problematischer zu handhaben sind. Ein Lösungsansatz, zu dem heute sowohl die Hersteller als auch die Unternehmen, die derartige Netzwerke einsetzen, tendieren, ist das "Distributed Broadband Management" (DBM). Bei den bis heute üblicherweise eingesetzten Management-Systemen wird der zentrale Ansatz bevorzugt, d.h., die einzelnen Netzwerkelemente (NE's) werden von einem zentralen Überwachungssystem (Operation Support System, OSS oder kurz Operation System, OS) verwaltet. Der DBM-Ansatz dagegen teilt das System in mehrere Schichten ein. Struktur-Elemente einer Schicht kommunizieren so nur untereinander und über definierte und vor allem standardisierte Schnittstellen mit der Schicht darüber und darunter. Mehrere Elemente (z.B. ein komplettes Subnetzwerk) können zusammengefaßt werden und stellen dann für die darüber liegende Schicht ein einzelnes logisches Element dar. Dies bewirkt auch die Verteilung des Managements, die auf dezentralen Subnetwork-Controllern (SNC's) aufbaut.
Ein Beispiel für ein sich in der Entwicklung befindliches dezentrales Management-System wird von der amerikanischen Federal Aviation Administration (FAA) entwickelt, die zur Luftraumüberwachung eine Vielzahl heterogener, aber verbundener Netzwerke einsetzt. Um die unterschiedlichen Netztechnologien transparenter zu machen und das Management auf sogenannte Air Route Traffic Control Center zu verteilen, wird u.a. von der FAA der Telecommunications ARTCC Prototype (TAP) betrieben. Dabei werden OSI Management Standards und Objekt-orientierte Techniken eingesetzt.

1. Eigenschaften des DBM

Interessante Aspekte gibt es eine Vielzahl für DBM. Letztendlich wird aber neben der Leistung vor allem aber auch das Kosten-Nutzen Verhältnis über den Einsatz entscheiden. Es zeichnen sich jedoch gerade auch hier Vorteile des DBM gegenüber traditionellen Management-Ansätzen ab.

Kurze Innovationszyklen

Im Netzwerkbereich insgesamt bewirkt die derzeit schnelle Entwicklung beim Einsatz von Telekommunikationsnetzwerken, daß einerseits sowohl die Zahl an Herstellern, unterschiedlichen Systemen und Produkten ständig zunimmt und gleichzeitig in immer geringerem Ausmaß homogene Netzwerke von nur einem Hersteller in einer einzigen Technologie eingesetzt werden. Bei der Entwicklung von zentralen Management-Systemen hat sich gezeigt, daß mittlerweile weit mehr als die Hälfte des Entwicklungsaufwandes auf die Integration der vielen unterschiedlichen Netzwerkelemente entfällt, da bei diesem Ansatz das zentrale OS selbst alle NE's überwacht. Dies reduziert natürlich die möglichen Aufwendungen für die Integration aktueller Entwicklungen. Beispiele hierfür sind Fernsteuerung von NE-Parametern oder File Transfer von Steuersoftware. Bei DBM wird die Integration der System-spezifischen Information auf den Hersteller des NE's verlagert. Diese Information wird dann entweder vom zentralen OS nicht mehr benötigt, wenn ein Element-Manager (EM) vom gleichen Hersteller des NE selbst Management-Aufgaben übernimmt (evtl. für mehrere NE's) oder sie wird dem OS über eine standardisierte Schnittstelle dargeboten. Ein weiterer Vorteil ist, daß der Hersteller des NE natürlich sein Produkt besser kennt als der Hersteller des OS, für ihn die Implementierung von Management-Funktionen für das NE also einfacher und vollständiger hinsichtlich der Fähigkeiten des NE zu bewerkstelligen ist. Insgesamt kann sich die Entwicklung mehr auf die "produktive" Integration neuer Techniken konzentrieren.

Verringerung des Konfigurationsaufwandes

Bei derzeitigen Management-Systemen muß ein großer Teil der Konfigurationsarbeit mehrfach geleistet werden: zunächst bei der eigentlichen Installation der NE's und später bei der Einbringung der veränderten bzw. neu aufgebauten Netzstruktur in die Datenbank des Management-Systems. Durch die Standardisierung der Schnittstellen des kompletten Management-Systems besteht nun die Möglichkeit, die Erfassung der relevanten Parameter automatisiert durchzuführen. Diese beinhalten :

- Information über die Hierarchie der Komponenten des Netzwerkes, um die Alarmüberwachung zu unterstützen.
- Information über die Provisioning-Fähigkeiten der NE's (Provisioning ist die Bereitstellung von Übertragungskapazität).
- Daten über den Aufbau der Netztopologie, benötigt für Alarmüberwachung und Provisioning.
- Information über die in den NE's zur Verfügung stehenden Ressourcen.

Leistungsfähigkeit

Die Entwicklung im Netzwerkbereich bringt es mit sich, daß von einzelnen Management-Systemen überwachte Netze immer größer und weiter verteilt sein werden. Dies erhöht die Gefahr, daß lokale Probleme, die sich auf die Funktion eines gesamten Management-Systemes auswirken können, dann auch sehr große Netzwerke beeinflussen. Verteilte Management-Systeme ermöglichen mit ihrer mehrschichtigen Architektur, an Schichtenschnittstellen Probleme zu isolieren. So können beispielsweise SNC's Probleme in ihrem Subnetzwerk nach oben abschirmen.

Schulungsaufwand

Die Anwendung von ständig neuer Technologie stellt hohe Anforderungen an das Personal, und die Aufwendungen für Schulung sind beträchtlich. Erfahrungen mit SONET haben gezeigt, daß auf EM Ebene eingesetzte Workstation oder PC basierte Graphische Benutzeroberflächen (GUI's) diesen Schulungsaufwand reduzieren können.

2. Struktur und Aufgaben von DBM-Systemen

Jedes Management-System besteht aus zwei grundlegenden Komponenten:

- dem TMN/DCN (Telecommunications Management Network - Data Communications Network)
- und der Management-System Architektur.

Das TMN/DCN ist ein eigenständiges Netzwerk, das wegen der Anforderungen an die hohe Überlebenswahrscheinlichkeit möglichst unabhängig von dem zu überwachenden Transportnetzwerk sein sollte, jedoch durchaus auch dessen Transportdienste nutzen kann. Die Architektur des Management-Systems stellt besonders Anforderungen an die Erweiterbarkeit und Adaptionsfähigkeit, sowohl was die Größe betrifft als auch den Einsatz neuer Technologien.

TMN/DCN

Das Design von einem TMN/DCN beinhaltet mehrere wichtige Aspekte, die aus Kosten-Nutzen Überlegungen kaum alle voll verwirklicht werden können. Von entscheidender Bedeutung ist die Überlebenswahrscheinlichkeit. Redundante Wege zwischen den Knoten des TMN/DCN sind daher unerläßlich. Ob man sich dabei nur auf die absolut notwendigen Wege beschränkt oder einen größeren Teil des Netzes redundant auslegt, wird vom Anforderungsprofil abhängen. Wichtig in dieser Hinsicht ist auch der Einsatz vielfältiger Protokolle und Übertragungsmethoden. Eingesetzt werden heute vor allem X-25 Paket-Netzwerke mit Übertragungsraten von 9,6 und 56 kBit/s. Mit der steigenden Leistung der zu überwachenden Transportnetzwerke und neuen Diensten, die von den Management-Systemen erwartet werden, steigen auch die Anforderungen an die Leistungsfähigkeit des TMN/DCN. Zukünftig werden wohl viele Techniken, die im Transportnetzwerk eingesetzt werden, auch im TMN/DCN verwendet werden, z.B. Frame Relay oder Cell Relay. Auch die Übertragungskapazitäten des Transportnetzwerkes werden eingesetzt werden. In SONET

Netzen kann so z.B. ein SONET DCC (Data Communications Channel) mit einer Kapazität von 576 kBit/s für Management-Daten reserviert werden. Dabei bleibt jedoch zu berücksichtigen, daß Probleme im Transportnetz dann auch in das TMN/DCN durchschlagen können. Zusätzlich können natürlich auch LAN's eingesetzt werden, wenn die räumlichen Voraussetzungen dies erlauben. Die konkrete Auslegung des TMN/DCN hat sich natürlich auch daran zu orientieren, welche Dienste das Management-System darüber realisieren soll (s.u.: DMS).

DMS (Distributed Management System)

Die Auslegung des DMS richtet sich nach den Anforderungen, die an es gestellt werden.

Configuration Management

Konfigurationsmanagement bezeichnet die Verwaltung und Fernsteuerung sowohl der Parameter als auch der Steuersoftware in den NE's. Erwünscht ist, die gesamte Kontrolle der NE's von einer zentralen Stelle aus durchzuführen, z.B. einem CO (Central Office). Vermieden werden soll damit Arbeit direkt an den einzelnen Komponenten des Management-Systems, z.B. die Veränderung von Systemparametern. Damit kann vor allem auch die Benutzerschnittstelle für Konfigurationsmaßnahmen vereinheitlicht werden.

Remote Software Management

NE's selbst sind komplexe Computersysteme, für die die gleichen Voraussetzungen gelten wie für andere Computersysteme auch. Sie benötigen Steuersoftware und verwalten Daten. Eine komfortable entfernte Steuerung solcher NE's beeinhaltet also auch die Verwaltung NE spezifischer Software und Daten. Um dies zu realisieren, muß das Management-System und das TMN/DCN die Möglichkeit von File Transfer und Backup/Restore-Anwendungen bieten.

Alarm-Überwachung

Die grundlegende Aufgabe eines Management-Systems ist die Alarm-Überwachung eines Transport-Netzwerkes. Bei der Auslegung des DMS sind Abwägungen hinsichtlich des Grades der Dezentralisierung zu machen: Ein weitgehend dezentraler Ansatz ist günstig, um das TMN/DCN und die zentrale Management-Instanz zu entlasten. Alarm-Nachrichten können dann schon in niederen Schichten, z.B. in EM's gefiltert und ausgewertet werden. Um jedoch einen globalen Überblick über den Zustand des Transportnetzes zu erlangen, müssen Alarm-Meldungen korreliert werden. Dies wird vorteilhaft von zentralen Systemen durchgeführt. Bei der Entscheidung, wie das DMS arbeiten soll, muß berücksichtigt werden, daß große zentrale Systeme insgesamt teurer sind als verteilte kleine Systeme, die bei einem verteilten Management-System die erforderliche Bearbeitung der Alarm-Nachrichten praktisch nebenher durchführen können.

Performance Monitoring (PM)

PM beschreibt die Aufgaben zur Erfassung und Auswertung von Informationen über den Zustand des Transportnetzwerkes. Mit den Kenntnissen, die daraus gewonnen werden, können Maßnahmen zur Sicherung von Quality of Service (QoS) Vereinbarungen wie auch zu vorbeugender Wartung initiiert werden. Auch zu Fragen der Erweiterung des Netzes, z.B. ob und gegebenenfalls wo aufgerüstet bzw. umstrukturiert werden soll, können Aussagen gemacht werden. Um für diese Zwecke interessante Informationen zu gewinnen, müssen die anfallenden Daten zunächst in zwei Schritten verarbeitet werden. In der ersten Stufe werden die Daten noch im NE gefiltert. Dies wird mittels sogenannter Threshold Crossing Alerts (TCA's) realisiert. Ein solcher TCA wird immer dann generiert, wenn ein NE-interner Parameter einen festgelegten Grenzwert überschreitet. In einer einfachen Netztopologie können diese TCA's dann direkt an das zentrale OS zur Auswertung weitergeleitet werden. Da aber Untersuchungen gezeigt haben, daß bei Fehlfunktionen im Transportnetz die Erzeugung von TCA's ausufern kann, bietet sich eine weitere Stufe der Filterung auf dezentraler Ebene an. Dies kann in CO-Management-Systemen (CO: Central Office, zentrale Management Stelle für ein größeres Subnetzwerk) durchgeführt werden. Derartige Maßnahmen können dazu beitragen, die Last auf dem TMN/DCN deutlich zu reduzieren. In der zweiten Stufe der Verarbeitung der PM-Informationen steht deren eigentliche Auswertung. Diese wird im zentralen Management- Center (CMC) durchgeführt. Dabei wird haupsächlich versucht, aus den Daten über lange Zeitperioden (Stunden, Tage oder Wochen) Trends zu entdecken. Auch werden die PM-Informationen und eingehende Alarm-Meldungen auf Zusammenhang untersucht. Die Voraussagen über den Netzzustand reichen von Minuten bis hin zu einigen Stunden, und die daraus resultierenden Antwortzeiten (z.B. Reparaturen oder Umkonfigurationen) liegen im Bereich von Stunden bis Tagen. Der Einsatz von neueren Technologien (z.B. ATM) erfordert jedoch in Teilbereichen (z.B. bei der Sicherung von QoS-Parametern) Reaktionszeiten im Bereich von Sekunden oder weniger. Manche der Methoden, die heute für PM angewandt werden, sind dann nicht mehr einsetzbar. Darunter fällt z.B. der Download der umfangreichen PM-Daten von CO basierten File-Servern ins zentrale OS mit Hilfe von File Transfer zu Zeitpunkten, an denen das TMN/DCN nur eine geringe Last aufweist. Wie genau die Reaktionszeiten derart gesenkt werden können, ist noch nicht untersucht.

Transport Provisioning (TP)

TP erfüllt die Bereitstellung von Übertragungspfaden zwischen Endsystemen und befaßt sich damit also mit Routing-Aufgaben. Dabei bezieht sich die Terminologie "Übertragungspfade" sowohl auf verbindungsorientierte als auch auf verbindungslose Kommunikationsdienste. TP ist die Management-Anwendung, bei der der zentrale Lösungsansatz voraussichtlich am längsten Bestand haben wird. Dies hat mehrere Ursachen:

- Endsystem-Übertragungspfade werden im allg. die Grenzen innerhalb eines DMS überschreiten. Um optimale Pfade zu finden, ist daher ein globaler Überblick erforderlich.
- Heute verfügbare TP-Applikationen erfüllen die TP-Anforderungen. Für diese Anwendungen spricht darüber hinaus der gute Support, der heute geleistet wird, und der Schulungsaufwand, der bei neuen Applikationen notwendig wäre. Da die TP-Problematik komplex ist, wäre dieser Aufwand sehr hoch.

Dennoch ist es wahrscheinlich, daß auch TP-Anwendungen zukünftig verteilt realisiert werden. Der Grund hierfür liegt vor allem an den Vorteilen einer einheitlichen Struktur eines Management-Systems. Datenaustausch oder die gemeinsame Nutzung von Daten mit anderen verteilt organisierten Management-Anwendungen werden dann möglich.

3. Abstraktionsmechanismen

Beim Entwurf von verteilten Management-Systemen werden im Vergleich zu konventionellen Management-Systemen einige neue Aspekte in den Vordergrund treten, die sich durch verschiedene Abstraktionsebenen ergeben.

Management Schichten

Grundlage für Abstraktion ist die Aufteilung des Management-Systems in mehrere logische Schichten. In den ITU-T Empfehlungen werden die folgenden vier Schichten genannt:

- Service Management Layer (SML)
- Network Management Layer (NML)
- Element Management Layer (EML)
- Element Layer (EL)

Konzept der logischen NE's

Die Trennung von NML und EML ermöglicht die Einführung von logischen NE's. Diese bestehen aus mehreren physikalischen NE's. Diese Menge von NE's wird der NML als einzelnes zu verwaltendes NE-Objekt präsentiert. Intern wird das logische NE von einem eigenen Manager verwaltet, der auch die Schnittstelle für die NML bietet. Das wichtigste Anwendungsmerkmal ergibt sich beim Provisioning. Das zentrale Managment-System kennt lediglich die Zugangspunkte und deren Bandbreite. Das Provisioning innerhalb des logischen NE wird von dem internen Management-System realisiert. Andere Anwendungen benötigen jedoch detailliertere Kenntnisse des logischen NE. So müssen z.B. Alarm-Nachrichten zu Fehlzuständen mit ausführlicherer Information über die Ursache versehen werden, bevor sie vom Management-System des logischen NE an die darüberliegende NML weitergereicht werden.

Betrachtungen der Schnittstellen-Spezifikationen und der Standardisierung

Ein kritischer Punkt bei der Realisierung der Schichtenstruktur ist die Standardisierung der Schnittstellen. So ist noch nicht klar, ob alle Schnittstellen standardisiert werden sollen. Die Telekommunikationsgesellschaft US-West z.B. spricht sich gegen die Standardisierung der EM-EML Schnittstelle aus. Das bedeutet, das innerhalb eines logischen NE's keine Vorgaben zur Implementierung gemacht werden. Grundlage ist die Annahme, daß die Hersteller von NE's auch die passenden Element-Management-Systeme liefern und Standards innerhalb der logischen NE's lediglich die technische Entwicklung einschränken. Wichtig für die Standardisierung bleiben aber die Schnittstellen oberhalb der logischen NE's.

4. Der ISO/OSI Ansatz

Auch bei der ISO wurde erkannt, daß Bedarf an einer einheitlichen, integrierten Management-Umgebung besteht, und schon seit einiger Zeit existieren Ansätze, die zwar nicht speziell auf Breitband-Transportnetzwerke ausgelegt sind, die aber dank ihrer verteilten und Objekt-orientierten Architektur geeignet sind, auch dort eingesetzt zu werden.

Common Framework for Management Integration

Um ein komplexes Transport-Netzwerk einheitlich zu managen, ist es notwendig, daß sich das gesamte Management-System an Standards hält, die in einem gemeinsamen Rahmen definiert wurden. In diesen Rahmen fallen:

- Einheitliche Kommunikationsregeln für alle Instanzen im Management-System.
- Eine festgelegte Menge von Operationen, die benötigt werden, um Management-Informationen auszutauschen, z.B. Anfragen nach Daten, Initiierung von Funktionen in einer entfernten Management-Einheit oder die Aufforderung, Ereignis-Meldungen weiterzuleiten.
- Eine Festlegung, welche Management-Aufgaben (s.o.) das Management-System zu erfüllen hat.
- Eine einheitliche Syntax und Semantik für die auszutauschenden Management-Informationen.
- Eine konsistente und integrierte Benutzerschnittstelle.
- Einheitliche Hard- und Software, auf der das Management-System läuft.
- Standardisierte Schnittstellen zwischen den verschiedenen Komponenten, auf die das Management-System zugreift, z.B. zu Datenbanken oder Kommunikationsfunktionen.

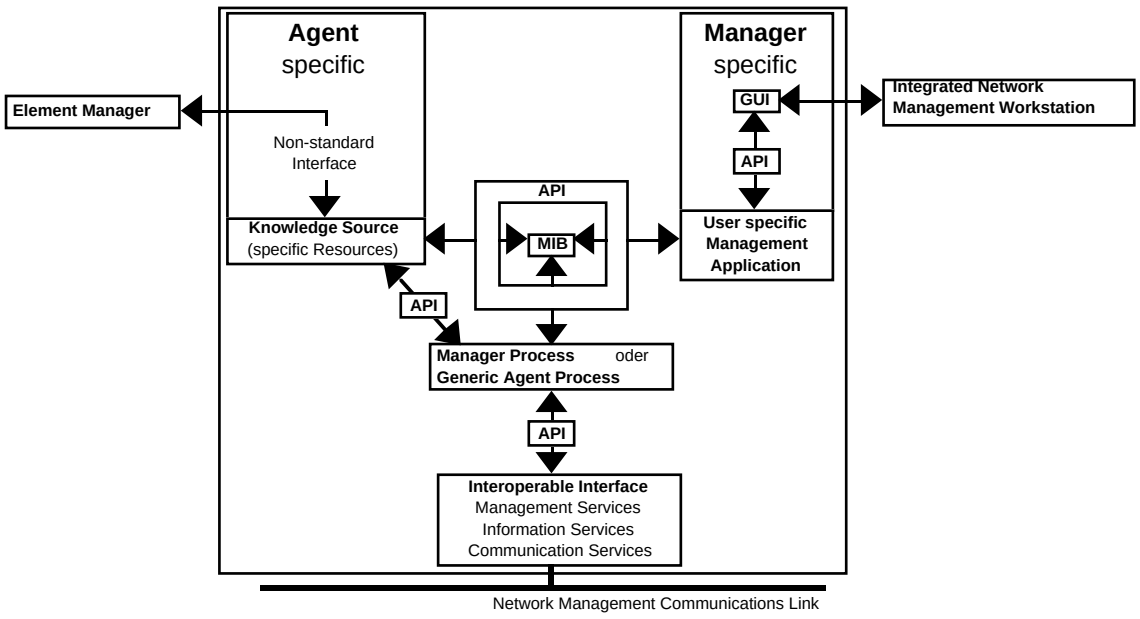
Welche dieser Anforderungen erfüllt werden, bestimmt den Grad der Integration, die das Management-System erreichen kann. Werden Anforderungen nicht erfüllt, werden spezielle Anpassungen erforderlich, z.B. spezielle Schnittstellen oder Konvertierungsroutinen, um die Gesamtfunktion zu realisieren. Daß alle diese Anforderungen in einem größeren Netzwerk erfüllt werden, ist unwahrscheinlich. Z.B. ist zu erwarten, daß in einem großen, heterogenen Netz auch die Hardware, auf der die einzelnen Management-Komponenten laufen, uneinheitlich ist.

Der Aufbau des TAP

Network Management Model

Das Modell, für das man sich für den TAP entschieden hat, basiert auf dem ISO/OSI Basis-Referenz-Model für Netzwerkmanagement und Festlegungen des "Network Management Forums" (Forum). Die bisher eingesetzten Forum Release 1 Spezifikationen definieren nicht

die Technologie, die innerhalb einer Management-Instanz verwendet werden soll, sondern lediglich, wie die Management-Instanzen miteinander kommunizieren. Der innere Aufbau bleibt dabei der einzelnen Implementierung überlassen und folgt beim TAP weitgehend den OSI Richtlinien. Die OSI Management Architektur besteht aus mindestens einem Manager-System und einem Agenten-System. Interaktion wird immer in Form von Operationsaufrufen vom Manager-System initiiert. Der interne Aufbau von Manager- und Agenten-System ist grundsätzlich gleich. Die unterschiedliche Funktionalität wird erreicht indem einer Instanz eine Rolle entweder als Agent oder Manager zugeteilt wird. Alle Instanzen verfügen über eine Schnittstelle, mit der auf das TMN/DCN zugegriffen wird. Diese besteht aus den "Communications Services", die das Protokoll definieren, mit dem die Management-Daten ausgetauscht werden sollen, den "Management Services", die die Management-Funktionalität spezifizieren, und den "Information Services", die die Namensgebung der Managed Objects festlegen. Konkret wird beim TAP als Protokoll das "Common Management Information Protocol (CMIP)" eingesetzt. Für die Communication Services wird der "Common Management Information Service (CMIS)" verwendet. Die Kommunikation zwischen zwei Systemen wird von dem Manager Process bzw. dem Agent Process kontrolliert, die über ein API auf die Kommunikationsschnittstelle zugreifen. Diese Prozesse benötigen "Managed Objects", um Ressourcen verwalten zu können. Diese Objekte werden von den Instanzen in einer "Management Information Base (MIB)" verwaltet, wobei das Manager-System Objekte für alle im Netzwerk vorhandenen Ressourcen, der Agent nur Objekte für die für ihn sichtbaren Ressourcen verwalten muß.



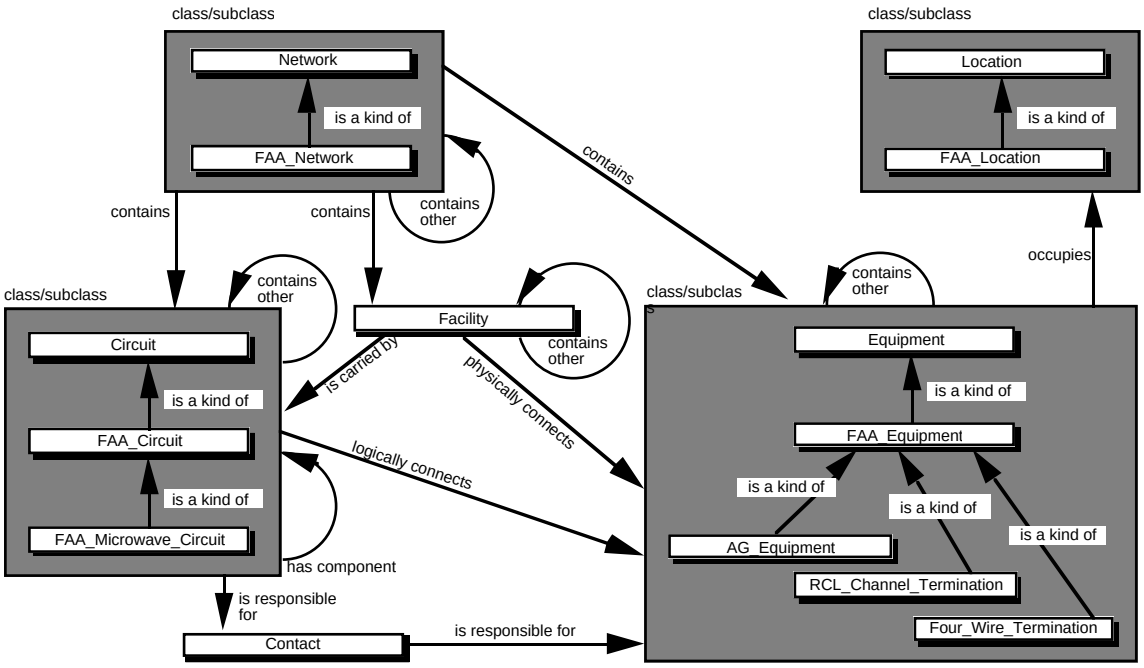
TAP

Network Management Model

Die Schnittstelle nach außen ist bei Agenten-Instanzen die "Knowledge Source", die direkt mit den zu verwaltenden Ressourcen (z.B. einem EM) verbunden ist. Die Manager-Instanzen verfügen über ein GUI.

Generic Network Data Model

Um die einzelnen Objekte in der MIB strukturiert anzuordnen, wurde beim TAP ein relativ einfaches Gegenstands/Beziehungs-Modell entworfen, das auf den verschiedenen Klassen aufbaut, die von ISO und Forum genormt wurden (s. Abbildung). Die eigentlichen Objekte sind Instanzen, die von diesen allgemein gehaltenen Klassen abgeleitet wurden.



Generic Network Data Model

Integrated Alarm and Status Monitoring Data Model

Um eine einheitliche Sicht auf Alarm- und Status-Nachrichten und -Ereignisse zu realisieren, wurde ebenfalls ein Objekt-orientiertes Gegenstands/Beziehungs-Modell entworfen. Zentral ist dabei die Klasse Log_Record mit ihren abgeleiteten Klassen Event_Log_Report, State_Change_Record und Alarm_Record, deren Instanzen die Information über eingehende Ereignisse speichern. Die anderen Gegenstandsobjekte sind die Ressourcen, auf die sich die Ereignisse beziehen, und Klassen, die für die Darstellung der Ereignisse auf dem GUI verwendet werden.

Architektur

Für die Architektur des TAP wurde vorgesehen, daß es einen Integration Manager und für jedes Subsystem (z.B. einen EM) einen Agenten geben soll. Die Agenten kommunizieren mit den ihnen zugeteilten Subsystemen mit nicht-standardisierten Nachrichten und erzeugen daraus OSI-konforme Nachrichten für den Integration Manager. In Zukunft werden die Agenten direkt in die zu verwaltenden Subsysteme integriert. Das GUI basiert auf OpenLook, der X-Window Oberfläche von Sun, und besteht aus einer Library, mit der problemlos spezielle Anwendungen erstellt werden können. Während die meisten Komponenten des TAP-Systems speziell entworfen wurden, werden für die Objekt-orientierte MIB und das TMN/DCN-Interface kommerzielle Produkte eingesetzt. Bis heute wird der TAP nur für Performance Monitoring und Alarm-Überwachung eingesetzt. Aus diesem Grund werden auch noch die alten Management-Systeme zu Konfigurations- und Kontrollaufgaben

weiterverwendet. Jedoch sollen im Zuge der Weiterentwicklung des TAP auch diese Fähigkeiten implementiert werden, so daß in absehbarer Zeit das gesamte Netz der FAA alleine durch einen Nachfolger des TAP verwaltet werden wird.

Abschließende Betrachtung des TAP

Der TAP ist ein erster Versuch, moderne Management-Methoden großflächig einzusetzen. Dabei muß jedoch berücksichtigt werden, daß die Entwicklung noch nicht abgeschlossen ist. Schließlich ist der TAP auch nur ein Prototyp. Die Nachfolger des TAP werden über weitere Anwendungsmöglichkeiten verfügen, die über die bis heute möglichen Funktionen der Alarm-Überwachung und des Performance-Monitoring hinausgehen. Ein wichtiger Aspekt des TAP ist, daß aufgrund noch fehlender Standards auf ein zukunftssicheres Common Computing Environment verzichtet werden mußte. Dies kann in der Zukunft zu Kompatibilitätsproblemen führen.

5. Einsatz von Common Computing Environment

Das gesamte Management-System des TAP wurde auf einer einheitlichen Hardwareplattform (SUN Workstations) realisiert. Auch die verwendete Software ist bis auf die Anpassungen an die unterschiedlichen zu verwaltenden Ressourcen im gesamten System gleich. Daher war man frei, spezielle Lösungen einzusetzen. Ist dies nicht mehr möglich, bietet sich der Einsatz von einem "Common Computing Environment" (CCE) an.

Die Gründe

Beim Aufbau von modernen Hochgeschwindigkeits-WAN's spielt heute SONET die tragende Rolle. Solange einzelne SONET-Ringe noch isoliert installiert werden, wird das gesamte System eines solchen Ringes meist noch aus Komponenten von nur einem Hersteller aufgebaut, so daß keine Interoperabilitätsprobleme zu erwarten sind. Jedoch ist gerade Interoperabilität eines der wichtigsten Argumente für SONET. Diese wird durch Herstellerunabhängigkeit erreicht. Die Folge wird sein, daß die SONET-Netze zusammenwachsen werden und es im Gesamtnetz Komponenten (sowohl Software wie auch Hardware) von vielen verschiedenen Herstellern geben wird. Diese Herstellervielfalt schlägt sich auch auf die Management-Komponenten nieder, wodurch dann Integrationsprobleme entstehen. Diese Probleme werden dann noch verschärft, wenn weitere Techniken (z.B. ATM), die auf den SONET-Netzen aufbauen, zur Anwendung kommen werden. Ein weiterer Grund, der für die Einigung auf ein CCE spricht, ist die Komplexität des Management-Systems. Es besteht selbst aus verschiedenen Komponenten, die an den verschiedenen Stellen im Management-System darüber hinaus noch in verschiedenem Umfang zu realisieren sind. Ein Beispiel dafür ist die MIB im OSI-Management-System: Sie muß sowohl in den Agenten wie auch in den Management- Systemen vorhanden sein. Jedoch werden von einem Agenten u.U. nur einige wenige Objekte verwaltet werden müssen, das Manager-System wird aber eine sehr große Datenbasis enthalten. Auch hier würde sich der Einsatz von Komponenten verschiedener Hersteller möglicherweise auszahlen.

Die Vorteile

Ein umfassendes CCE bietet sowohl den Herstellern als auch den Netzbetreibern Vorteile. Für die Hersteller der NE's gab es das Problem, daß jeder Netzbetreiber eine andere Schnittstelle zu seinem Management-System implementiert haben wollte, so daß an dieser Stelle der Entwicklungsaufwand mehrfach geleistet werden mußte. Wird ein CCE zum Standard für alle, entfällt für den Hersteller dieser zusätzliche Aufwand. Der Netzbetreiber hat den Vorteil, daß er Komponenten von verschiedenen Herstellern einsetzen kann.

- Dies wird den Preis der Komponenten senken.
- Die Qualität der Software wird steigen, weil zum einen nur noch für das CCE entwickelt wird und zum anderen durch die mögliche Kombination von Produkten unterschiedlicher Hersteller für jede Komponente die jeweils beste Lösung eingesetzt werden kann.
- Die Abhängigkeit von einem Hersteller wird nachlassen.

Techniken und Komponenten

Da die zu realisierende Funktionalität eines CCE umfangreich ist, besteht es selbst wiederum aus verschiedenen Komponenten. Auch wenn es noch keinen offiziellen Standard gibt, scheint DCE (Distributed Computing Environment) der Open Software Foundation als Basiselement eines genormten CCE fast sicher. DCE stellt für verteilte Anwendungen die grundlegenden Mittel zur Interaktion zu Verfügung. Die Funktionalität von DCE umfaßt RPC, Threads, Directory Services, Time Services und Security Services. DCE ist herstellerneutral und wird von der Industrie favorisiert, ist aber noch nicht völlig ausgereift. Eine weitere Schlüsselposition kommt DME (Distributed Managment Environment) zu, wobei gesagt werden muß, daß die ersten DME-Implementierungen frühestens in einem Jahr erscheinen werden. Im Gegensatz zu DCE, das als integriertes Ganzes zu sehen ist, ist DME eher eine Sammlung von Komponenten, die auf DCE basiert. Diese wären:

- Distributed Services. Sie erfüllen allgemeine Funktionen wie Ereignisverarbeitung, Drucken oder Filetransferdienste.
- Network Management Option (NMO). Die Hauptfunktionalität ist das XMP API, ein gemeinsames API für das SNMP und das CMIP.
- Object Management Framework (OMF). OMF soll die Interaktion zwischen den Management Objekten ermöglichen. Die Vorzüge des OMF ergeben sich aus den allgemeinen Vorzügen des Objekt-orientierten Paradigmas.

Ein dritter Ansatz ist DPE, das u.a. auf DCE und DME basiert und auf die Kommunikation zwischen sogenannten Buildingblocks (BB's) ausgerichtet ist. Diese BB's sind Funktionseinheiten, die selbst wiederum aus mehreren Objekten zusammengesetzt sind.

Aussichten

Daß es einen CCE-Standard geben wird, ist sehr wahrscheinlich. Allerdings ist noch nicht sicher, aus welchen Komponenten er aufgebaut sein wird. Wahrscheinlich ist, daß DCE als

Grundlage dabei sein wird. Möglicherweise wird es auch zunächst mehrere konkurrierende Ansätze geben.

6. Conclusio

Beim Vergleich von konventionellen und verteilten Management-Systemen zeigt sich, daß ab einer bestimmten Netzgröße der verteilte Ansatz Vorteile besitzt. Sowohl die Hersteller als auch die Netzbetreiber werden aus diesem Grund die Verabschiedung einheitlicher Standards forcieren. Diese sind allerdings entscheidend, um die theoretischen Anwendungsmöglichkeiten auch praktisch umzusetzen. Die Grundzüge dieser Standards sind mittlerweile absehbar. Man kann davon ausgehen, daß DCE in Kombination mit einem Objekt-orientierten Zusatz für die Management-Funktionalität, etwa DME, als Basis-System verwendet werden wird.
Mittelfristig werden wohl auch kleinere Netze mit dem dann standardisierten verteilten Ansatz verwaltet werden, der zwar bei geringer Ausdehnung des Netzes keine direkten Vorteile mehr bieten kann, aber durch die Möglichkeit der praktisch unbegrenzten Erweiterbarkeit und der dann vorhandenen Standardapplikationen die Flexibilität erhöhen wird.

7. Glossar

CCE	Common Computing Environment
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CO	Central Office
DBM	Distributed Broadband Management
DCE	Distributed Computing Environment
DME	Distributed Management Environment
DMS	Distributed Management System
EL	Element Layer
EM	Element Manager
EML	Element Management Layer
MIB	Management Information Base
NE	Network Element
NML	Network Management Layer
NMO	Network Management Option
OMF	Object Management Framework
O(S)S	Operations (Support) System
PM	Performance Monitoring
QoS	Quality of Service
SML	Service Management Layer
SNC	Sub Network Controller
SNMP	Simple Network Management Protocol
TCA	Threshold Crossing Alert
TMN/DCN	Telecommunication Management Network - Data Communications Network

8. Literatur

/STRAT94/ Stratman, Robert H.
Development of an Integrated Network Manager for Heterogeneous Networks Using OSI Standards and Object-Oriented Techniques
IEEE Journal on selected areas in communications,
Vol. 12, No. 6, S.1110 ff.

/AHREN94/ Ahrens, Mike
Key Challenges in Diestributed Management of Broadband Transport Networks
IEEE Journal on selected areas in communications,
Vol. 12, No. 6, S.991 ff.

/FALK90/ Falkner, Rüdiger und Müllner, Harald
Telecommunications Management Network (TMN): Architektur, Schnittstellen und Anwendungen
ntz Bd. 43 (1990) Heft 6