UNIVERSITÄT ROSTOCK

Diplomarbeit

Analyse und Visualisierung des Kommunikationsaufkommens in paketorientierten Netzen

vorgelegt am: 15.01.2005

Studienfach: Informationstechnik

von: cand. ing. Udo Brunswig
Budapester Straße 32
18057 Rostock

Matrikelnummer: 098201500

Betreuer: Dr.-Ing. Hans-Dietrich Melzer
Dipl.-Ing. Thomas Kessler
Dipl.-Ing. Thomas Vergin

Aufgabenstellung

Analyse und Visualisierung des Kommunikationsaufkommens in paketorientierten Netzwerken

Die Anforderungen an die Qualität und die Verfügbarkeit von Datennetzen erhöhen sich stetig. Eine Analyse des Kommunikationsaufkommens stellt neben der Prognose über die Nutzung von Diensten eine wesentliche Komponente für die Planung von Netzstrukturen dar. Die statistische Auswertung von Datenverkehr liefert wesentliche Parameter wie Datenaufkommen, Nutzer- und Protokollverteilung für die Modellierung von Datenquellen und einer damit verbundenen Optimierung von z.B. Zugangsnetzen. Die Visualisierung der Parameter unterstützt diesen Prozess bei der Erkennung von Trends oder der Darstellung komplexer Zusammenhänge.

Der Kandidat soll im Rahmen der Diplomarbeit Verfahren und Algorithmen der Visualisierung von Datenströmen und deren Parametern zur Unterstützung der Analyse des Kommunikationsaufkommens in paketorientierten Datennetzen untersuchen. Die Ergebnisse dieser Untersuchung sind in einem Tool zur Visualisierung von Verkehrsparametern in einer Form umzusetzen, die es gestattet, für die Modellierung von Datenquellen, das Management von Netzwerken und für die Planung der Netzstruktur geeignete Aussagen zu erzielen. Insbesondere folgende Punkte sind in der Diplomarbeit zu berücksichtigen:

- Klassifizierung und Vergleich von Verfahren zur Analyse von Performance- und QoS/CoS Eigenschaften in paketorientierten Kommunikationsnetzen anhand geeigneter Vergleichskriterien unter Berücksichtigung des ISO/OSI Referenzmodells
- Klassifizierung aktueller Darstellungsformen, graphischer Analyseverfahren und Algorithmen; Betrachtung der Vor- und Nachteile in Bezug auf die in der Zielstellung definierten Parameter. Das Einbeziehen von Darstellungsformen, die in thematisch entfernten Fachbereichen genutzt werden, ist ausdrücklich erwünscht.
- Entwicklung eines generischen, an Standards orientierten Datenformats für die Speicherung der Messdaten in einer zur Analyse und grafischen Darstellung geeigneten Form unter Berücksichtigung der zuvor ausgewählten Analyseverfahren (paket-, flowbasiert)
- Realisierung eines parametrisierbaren Tools zur visuellen Analyse von paketorientierten Datenverkehr (insbesondere Ethernet und IP Datenverkehr).

Zusammenfassung

Analyse und Visualisierung des Kommunikationsaufkommens in paketorientierten Netzen

von cand. ing. Udo Brunswig

Die Simulation von Netzwerken ist für die Planung und Gestaltung von zukünftigen Kommunikationsstrukturen und eine Zusicherung von Dienstgüteparametern (QoS/CoS) unabdingbar. Grundlage von Simulationen sind Verkehrsmodelle, die unter anderem mit Hilfe der Visualisierung des Datenaufkommens bestehender Netze erstellt werden können.

Im Rahmen dieser Arbeit wurden die Parameter der Übertragungsprotokolle vorgestellt, die eine Charakterisierung des Netzwerkverkehrs bezüglich der Visualisierung ermöglichen. Des Weiteren werden neue und unkonventionelle Visualsierungstechniken beschrieben und in Hinsicht auf die Effizienz der Darstellung des Datenverkehrs in paketorienterten Netzen untersucht und beurteilt.

Für die Speicherung der Messdaten, die bei der Untersuchung der bestehenden Netzstrukturen anfallen, ist ein einheitliches und flexibles Datenformat notwendig. Auf Grundlage des zukünftigen IPFIX-Standards wurde ein Datenformat zur Speicherung der erfassten Messdaten entwickelt.

Um die theoretischen Gesichtspunkte der erfolgten Untersuchung in die Praxis umzusetzen, wurde ein Tool für die visuelle Analyse des Datenverkehrs in paketorientierten Netzen implementiert.

Inhaltsverzeichnis

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
ACK	Acknowledge-Flag
AMD	Advanced Micro Devices
ATM	Asynchron Transfer Mode
BGP	Border Gateway Protocol
CoS	Class of Service
CPU	Central Processing Unit
DF	Don't Fragment-Flag
DDR	Double Data Rate
DSCP	Differentiated Services Code Point
ERF	Extensible Record Format
FIN	Finish-Flag
FPGA	Field Programmable Gate Array
HDLC	High Level Data Link Control
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IEEE	Institute of Electric and Electronic Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Multicast Protocol
IHL	Internet Header Length
IP	Internet Protocol
IPFIX	IP Flow Information Export
IPv4	Internet Protocol Version 4

IPv6	Internet Protocol Version 6
ISO	International Standards Organization
JPG	Joint Photographic Experts Group
LLC	Logical Link Control
LVM	Logical Volume Manager
MAC	Medium Access Control
MB	Megabyte
MEZ	Mitteleuropäische Zeitzone
MF	More Fragments
MPLS	Multi-Protocol Label Switching
MTU	Maximum Transfer Unit
OPMAN	Open IP-based Multiservice Access Network
OSI	Open System Interconnection
PAR	Positive Acknowledgement with Retransmission
PCI	Peripheral Component Interface
PDF	Portable Document Format
PSH	Push Data to Application-Flag
QoS	Quality of Service
RAM	Random Access Memory
RFC	Request for Comments
RST	Reset-Flag
SYN	Synchronisation-Flag
TCP	Transmission Control Protocol
TOS	Type of Service
TTL	Time to Live

UDP	User Datagram Protocol
UNIX	Uniplexed Information and Computing System
URG	Urgent Pointer
VIA	VIA Technologies, Inc

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

MOTIVATION UND ZIELSTELLUNG

Der enorme Erfolg des Internets und das damit einhergehende rasante Wachstum des Datenverkehrs stellen hohe Anforderungen an die Übertragungsstrukturen des Internets. Neue Kommunikationsdienste wie Voice over IP, Video on Demand, E-Learning und Videokonferenzen stellen andere Ansprüche an die Qualität der Datenübertragung als die konventionelle Dateiübertragung.

Eine Analyse und Prognose der Nutzung dieser Kommunikationsdienste sind für die Planung und den Umbau von Netzwerken notwendig, um Aussagen über die Quantität und Qualität der Datenübertragung treffen zu können. Damit können Fehlplanungen erkannt und vermieden werden.

Um diesen Ansprüchen gerecht zu werden, kann eine Simulation von Datenübertragungsnetzen Aussagen liefern, die eine effizientere Netzwerkplanung unterstützt. Die Simulation von Datenübertragungsnetzen ist ein Teil des Forschungsprojektes 'Open IP-based Multiservice Access Network' (OPMAN), das am Institut für Nachrichtentechnik und Informationselektronik der Universität Rostock bearbeitet wird und in das sich diese Arbeit einbettet.

Die Grundlage dieser Simulation bilden Modelle, die mittels statistischer Analyse des Datenverkehrs bestehender Netzwerke erstellt werden.

Für vergleichbare Aussagen zum Datenverkehr in verschiedenen Netzwerken müssen die Übertragungsprotokolle der Netzwerke auf relevante Parameter untersucht werden, die den Datenverkehr charakterisieren.

Die Nutzung eines einheitlichen Datenformats für die Speicherung der charakterisienden Parameter des Datenverkehrs erleichtert die Analyse der Messdaten.

Eine Visualisierung der untersuchten Parameter unterstützt den Analyseprozess und ist beim Entdecken und der Darstellung von komplexen Zusammenhängen hilfreich. Dabei muss beachtet werden, dass nur aussagekräftige Visualisierungstechniken eine wirkliche Hilfestellung leisten können. Es ist deshalb notwendig, dass klassische, wie auch unkonventionelle Darstellungsverfahren auf Effizienz der Visualisierung in Bezug auf die relevanten Parameter des paketorientierten Datenverkehrs untersucht werden.

Aus der Aufgabenstellung und der Motivation abgeleitet, ergeben sich folgende Zielstellungen, die Grundlage für die Aufteilung der vorliegenden Arbeit sind:

- Vorstellung der Übertragungsprotokolle paketorientierter Netzwerke,
- Untersuchung und Auswahl der Parameter der Protokolle nach Darstellbarkeit,
- Vorstellung von Visualisierungsverfahren,
- Untersuchung und Bewertung von Visualisierungsverfahren in Bezug auf die Visualisierung gewählter Parameter zur Charakterisierung des Datenverkehrs,
- Aufbau, Beschreibung und Performanceanalyse eines Messsystems zur Messung von Datenverkehr,
- Entwicklung eines Datenformates für die Speicherung der Messdaten in einer zur Analyse und grafischen Darstellung geeigneten Form
- Realisierung eines Tools zur visuellen Analyse von paketorientiertem Datenverkehr

Kapitel 2

ANALYSE DES KOMMUNIKATIONSAUFKOMMENS IN PAKETORIENTIERTEN NETZEN

Um das Kommunikationsaufkommen in paketorientierten Netzen darstellen zu können, muss eine Analyse der übertragenen Daten vorgenommen werden. Die Daten werden in den zu betrachtenden Netzen paketorientiert transportiert. Die Applikationen segmentieren die zu übertragenden Daten, fügen Adressierungs- und Verwaltungsoptionen hinzu und übergeben die Daten als ein Paket bestimmter Länge an die darunter liegende Schicht (Abbildung 2.1). Der sendende Host arbeitet den Potokollstapel von oben nach unten ab. Dabei fügt jede Schicht ihre Adressierungs- und Verwaltungsdaten als einen Header hinzu. Die Bitübertragungsschicht prägt die segmentierten Informationen auf das Übertragungsmedium (Kupferkabel, Lichtwellenleiter, Luft) auf. Auf der Empfangsseite werden in der jeweiligen Schicht die Headerinformationen vom eigentlichen Nutzlastanteil abgetrennt und ausgewertet. Der Nutzlastteil des Pakets wird an die darüber liegende Schicht übergeben. Ob dies sofort oder nach einer Neuübertragung vonstatten geht, weil das Datensegment beschädigt übertragen wurde, hängt von der

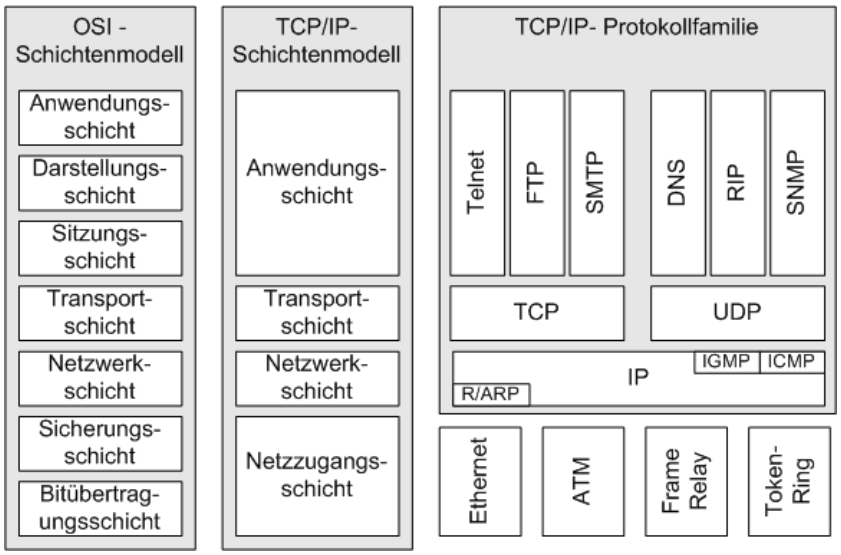

Abbildung 2.1: OSI- und TCP/IP-Schichtenmodell

Auswertung der Verwaltungsoptionen im Header des Pakets durch die jeweilige Schicht ab.

2.1 Bitübertragungsschicht

Die Bitübertragungsschicht (physical layer) ist für die ordnungsgemäße Übertragung der Bits über den physikalischen Kanal zuständig. In dieser Schicht werden die Codierung der Signale und eine Anpassung auf das Übertragungsmedium vorgenommen.

Zwecks Visualisierung existieren in dieser Schicht keine Parameter oder Attribute, die Aussagen über den Netzwerkverkehr zulassen.

2.2 Sicherungsschicht

Die Sicherungsschicht (data link layer) hat die Aufgabe, aus der Menge von Bits, die von der Bitübertragungsschicht empfangen wurden, weiter verwertbare Informationen zu gewinnen. Die Hauptaufgaben dabei sind das Erkennen von Rahmengrenzen und die Überprüfung der Korrektheit des empfangenen Rahmens. Weiterhin ist diese Schicht für die optimale Ausnutzung des physikalischen Mediums zuständig, sofern dies ein Medium ist, welches von mehreren Teilnehmern gleichzeitig genutzt wird. Die Sicherungsschicht wird oft in zwei Teile geteilt:

Schicht 2a: MAC (Medium Access Control). Steuert den Zugriff auf das Medium.

Schicht 2b: LLC (Logical Link Control) Zuständig für die Segmentierung der zu sendenden Daten, sowie für das korrekte Zusammensetzen der empfangenen Daten. Anforderung von Wiederholungssendungen, Senden von Quittungen für empfangene Rahmen.

Die Protokollparameter, die in der Sicherungsschicht benutzt werden, spielen nur in der direkten Verbindung zweier Instanzen eine Rolle. Es sind u. a. eine eindeutige Adressierung der Netzzugangsgeräte und Prüffelder, mit deren Hilfe festgestellt werden kann, ob die Rahmen richtig übertragen wurden.

Bei der Auswahl für Visualisierungsattribute spielt die Sicherungsschicht eine untergeordnete Rolle.

2.3 Vermittlungsschicht

Die Vermittlungsschicht (auch Netzwerkschicht) hat die Aufgabe, eine Verbindung zwischen den einzelnen Knoten im Netzwerk herzustellen. Die Vermittlungsschicht

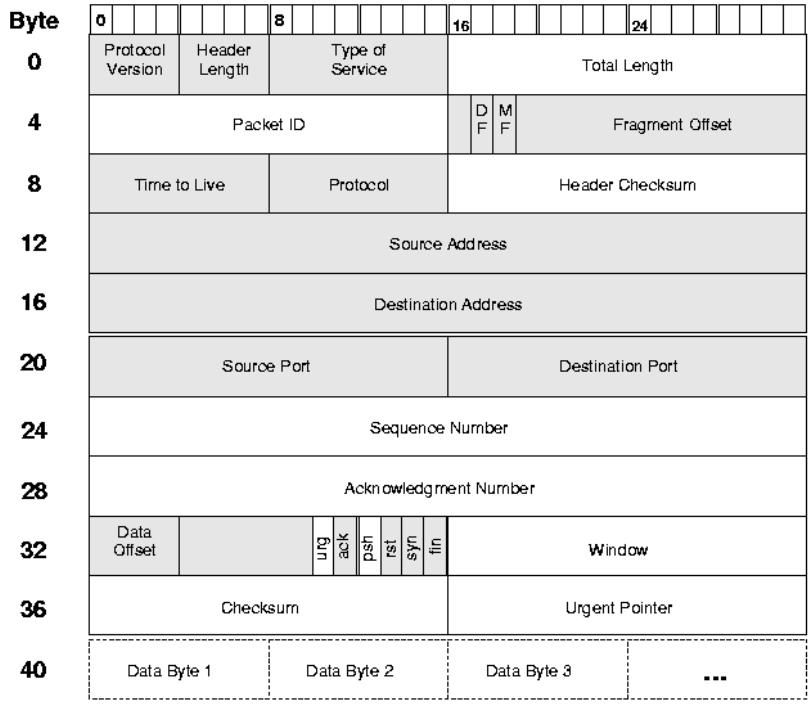

Abbildung 2.2: TCP/IP Header

soll dabei den höheren Schichten die Details der Paketübertragung über das Netzwerk verbergen. Die Hauptaufgabe der Vermittlungsschicht ist die Auswahl des Weges des Pakets (Routing) vom Absender zum Empfänger. Ein Beispiel für ein Protokoll der Vermittlungsschicht ist das Internet-Protokoll der Version 4 (IPv4).

2.3.1 Internet-Protokoll v4

Im Folgenden wird das Internet-Protokoll der Version 4 mit den Parametern vorgestellt, die für die Visualisierung des Netzwerkverkehrs von Interesse sind.

Das Internet-Protokoll v4 überträgt die Daten in einem paketorientierten Netz. Ein Paket ist ein Datenblock mit Nutzdaten und Informationen, die notwendig sind, um sie dem Empfänger zuzustellen. Das Datagramm (datagram) ist das Paketformat, welches das Internet-Protokoll definiert. Ein IP-Datagramm besteht aus einem Header und den zu übertragenden Daten. Der Header der Version 4 des IP (ersten 20 Bytes der Abbildung 2.2) hat einen festen 20 Byte großen Anteil, anschliessend folgt ein optionaler Teil variabler Länge. Der Header umfasst alle Informationen, die notwendig sind, um dass Datagramm dem Empfänger zuzustellen. Ein Datagramm kann maximal 64 KByte groß sein; in der Praxis

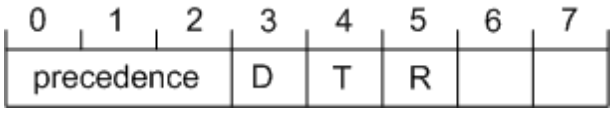

Abbildung 2.3: TOS/DCSP Feld

liegt die Größe bei ca. 1500 Byte.

Version: Der Wert des Version-Feldes zeigt die Version des Internet-Protokolls. Dadurch bleibt die Kompatibilität der Versionen untereinander erhalten. Anhand dieses Wertes ist es der Netzkomponente möglich, die Verarbeitung der folgenden Daten anhand des für die Version richtigen Bitmusters durchzuführen. Die aktuelle Versionsnummer des IP ist 4; Version 6 des Internet-Protokolls ist auf Teilstrecken des Internets probeweise in Betrieb.

Header Length: Das Feld Header Length (Internet Header Length - IHL) enthält die Länge des Protokollkopfs, da diese nicht konstant ist. Die Länge wird in 32-Bit-Worten angegeben. Der kleinste zulässige Wert ist 5 - das entspricht 20 Byte; in diesem Fall sind im Header keine Optionen gesetzt. Die Länge des Headers kann sich durch Anfügen von Optionen aber bis auf 60 Byte erhöhen (der Maximalwert für das 4-Bit-Feld ist 15).

Type of Service (TOS/DCSP Feld): Über das Feld Type of Service kann IPv4 angewiesen werden, Pakete nach vorbestimmten Kriterien zu behandeln. Als Dienstklassen sind Kombinationen aus Zuverlässigkeit und Geschwindigkeit möglich. In der Praxis wird dieses Feld bei der Verarbeitung von Paketen in den Netzkomponenten jedoch nicht berücksichtigt. Der Aufbau des Feldes ist in Abbildung 2.3 illustriert.

Total Length: Es enthält die gesamte Paketlänge, d.h. Header und Daten. Da es sich hierbei um ein 16-Bit-Feld handelt, ist die Maximallänge eines Datagramms auf 65.535 Byte begrenzt. In der Spezifikation von IP (RFC 791) wurde festgelegt, dass jeder Host in der Lage sein muss, Pakete bis zu einer Länge von 576 Bytes zu verarbeiten. In der Regel können von den Hosts aber Pakete größerer Länge verarbeitet werden.

Identification: Über das Identifikationsfeld kann der Zielhost feststellen, zu welchem Datagramm ein empfangenes Fragment gehört. Alle Fragmente eines Datagramms enthalten die gleiche Identifikationsnummer, die vom Absender vergeben wird.

Flags: Das Flags-Feld ist drei Bit lang. Die Flags bestehen aus zwei Bits na-

mens DF - Don't Fragment und MF - More Fragments. Das erste Bit des Flags-Feldes ist ungenutzt bzw. reserviert. Die beiden Bits DF und MF steuern die Behandlung eines Pakets im Falle einer Fragmentierung. Mit dem DF-Bit wird signalisiert, dass das Datagramm nicht fragmentiert werden darf. Mit dem MF-Bit wird angezeigt, ob einem IP-Paket weitere Teilpakete nachfolgen. Dieses Bit ist bei allen Fragmenten außer dem letzten gesetzt.

Fragment Offset: Wurde nicht das DF-Flag gesetzt, dann gibt das Fragment Offset Feld an, an welcher stelle des Datagramms ein neues Fragment beginnt. Somit kann der Zielhost das Originalpaket aus den Fragmenten zusammensetzen.

Time to Live: Das Feld Time to Live ist ein Zähler, mit dem die Lebensdauer von IP-Paketen begrenzt wird. Im RFC 791 ist festgelegt, dass ein Paket eine maximale Lebensdauer von 255 Sekunden (8 Bit) hat. Der Zähler wird von jedem Netzknoten, der durchlaufen wird, um mindestens eins dekrementiert. Bei einer längeren Zwischenspeicherung in einem Router wird der Inhalt mehrfach dekrementiert. Enthält das Feld den Wert 0, wird das Paket verworfen. Dadurch wird verhindert, dass ein Paket endlos in einem Netzwerk weitergeleitet wird. Der Absender wird in einem solchen Fall mittels einer ICMP-Nachricht (Internet Control Management Protocol- Nachricht) informiert.

Protocol: Enthält die Nummer des Transportprotokolls, an das das Paket weitergeleitet werden muss. Die Nummerierung von Protokollen ist im gesamten Internet einheitlich. Diese Aufgabe übernimmt die Internet Assigned Numbers Authority (IANA) (http://www.iana.org).

Header Checksum: Dieses Feld enthält die Prüfsumme der Felder im IP-Header. Die Nutzdaten des IP-Datagramms werden nicht geprüft. Die Prüfsumme wird von jedem Netzknoten, der durchlaufen wird, neu berechnet. Die Ursache dafür ist, dass sich der IP-Header durch die Dekrementierung des Time-to-Live-Feldes bei jeder Teilstrecke ändert.

Source Address, Destination Address: In diese Felder werden die 32-Bit langen Internet-Adressen eingetragen.

Options und Padding: Das Feld Options wurde im Protokollkopf aufgenommen, um die Möglichkeit zu bieten, das Internet-Protokoll um weitere Informationen zu ergänzen, die im ursprünglichen Design nicht berücksichtigt wurden. Das Optionsfeld hat eine variable Länge. Jede Option beginnt mit einem Code von einem Byte, über den die Option identifiziert wird. Es besteht die Möglichkeit, dass der ersten Option ein weiteres Optionsfeld von 1 Byte mit einem oder mehreren Datenbytes folgt. Das Feld Options wird über das Padding-Feld auf ein

Vielfaches von 4 Byte aufgefüllt. Derzeit sind die folgenden Optionen möglich:

- End of Option List: Kennzeichnet das Ende der Optionsliste.
- No Option: Kann zum Auffüllen von Bits zwischen Optionen verwendet werden.
- Security: Bezeichnet, wie geheim ein Datagramm ist. In der Praxis wird diese Option jedoch fast immer ignoriert.
- Loose Source-Routing, Strict Source-Routing: Diese Option enthält eine Liste von Internet-Adressen, die das Datagramm durchlaufen soll. Auf diese Weise kann dem Datenpaket vorgeschrieben werden, eine bestimmte Route durch das Internet zu nehmen. Beim Source-Routing wird zwischen "Strict Source and Record Route" und "Loose Source and Record Route" unterschieden. Im ersten Fall wird verlangt, dass das Paket diese Route genau einhalten muss. Des weiteren wird die genommene Route aufgezeichnet. Die zweite Variante schreibt vor, dass die angegebenen Router nicht umgangen werden dürfen. Auf dem Weg können aber auch andere Router passiert werden.
- Record Route: Die Knoten, die dieses Datagramm durchläuft, werden angewiesen ihre IP-Adresse an das Optionsfeld anzuhängen. Damit lässt sich ermitteln, welche Route ein Datagramm genommen hat. Die Größe für das Optionsfeld ist auf 40 Byte beschränkt. Deshalb kommt es heute auch oftmals zu Problemen mit dieser Option, da weit mehr Router durchlaufen werden.
- Timestamp: Diese Option ist mit der Option Record Route vergleichbar. Zusätzlich zur IP-Adresse wird bei dieser Option die Uhrzeit des Durchlaufs durch den Knoten vermerkt. Auch diese Option dient hauptsächlich zur Fehlerbehandlung, wobei zusätzlich z.B. Verzögerungen auf den Netzstrecken erfasst werden können.

2.3.2 Internet-Protokoll v6

Mit dem rasanten Anstieg der ans Internet angebundenen Hosts besteht nicht nur die Notwendigkeit, dass eine größere Bandbreite für den Datenverkehr bereitstehen muss. Der Adressraum, der für die eindeutige Adressierung jedes Hosts

bereitstehen muss, wird in absehbarer Zeit erschöpft sein. Des Weiteren funktioniert das Internet nach dem Prinzip "best effort". Es gibt keine Zusicherungen, dass eine Datenübertragung mit einer mindest geforderten Übertragungsgeschwindigkeit abläuft. Es können auch keine Aussagen zur Laufzeit eines Pakets vom Quell- zum Zielhost gemacht werden. Diese Dienstarten (QoS/CoS) sollen im Internet-Protokoll der Version 6 berücksichtigt werden. Die IETF (Internet Engineering Task Force) begann 1990 mit der Entwicklung der IPv6. Die wichtigsten Ziele diese Projektes sind:

- Unterstützung von Milliarden von Hosts, auch bei ineffizienter Nutzung des Adressraums
- Reduzierung des Umfangs der Routing-Tabellen
- Vereinfachung des Protokolls für ein schnelleres Paket-Routing
- Höhere Sicherheit (Authentifikation und Datenschutz) als das heutige IPv4
- Mehr Gewicht auf Dienstarten, insbesondere für Echtzeitanwendungen
- bessere Unterstützung von Multicasting
- Möglichkeit für Hosts, ohne Adressänderung von anderen Orten aus das Internet zu nutzen
- Freiraum für Weiterentwicklungen des Protokolls
- Unterstützung der alten und neuen Protokolle in Koexistenz für Jahre

[Hol97]

Die folgende Beschreibung von IPv6 orientiert sich an RFC 2460, welche den aktuellen Stand der Spezifikation des Internet-Protokolls v6 darstellt. RFC 2460 enthält einige wesentliche Änderungen der Spezifikation gegenüber RFC 1883.

Die meisten Felder des IPv4 wurden bei IPv6 übernommen. Trotz dessen ist IPv6 nicht zu der älteren Version kompatible. Da IPv6 nach einer Übergangsphase IPv4 ersetzen soll, ist IPv6 zu allen anderen Protokollen, wie TCP oder UDP voll kompatible. Die charakteristischen Merkmale von IPv6 sind:

- Adressgröße: IPv6-Adressen haben anstatt 32 Bit nun 128 Bit breite Adressierungsfelder. Theoretisch lassen sich damit $2^{128} = 3.4 * 10^{38}$ Adressen vergeben.

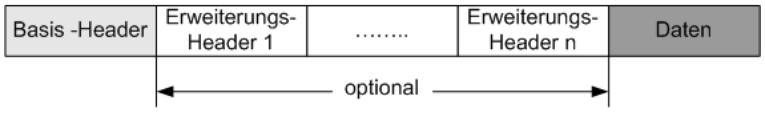

Abbildung 2.4: Form eines IPv6-Datengramms

- Header-Format: Der IPv6 (Basis)Header ist vollständig verändert worden. Der Header enthält 7 anstatt 13 Felder. Damit ist eine schnellere Verarbeitung der Pakete in Routern möglich. Im Gegensatz zu IPv4 gibt es bei IPv6 nicht mehr nur einen Header, sondern einen Basisheader und mehrere optionale Header.

- Erweiterte Unterstützung von Optionen und Erweiterungen: Die Erweiterung der Optionen ist notwendig geworden, da einige, bei IPv4 notwendige Felder nun optional sind. Darüber hinaus unterscheidet sich auch die Art, wie die Optionen dargestellt werden. Für Router wird es damit einfacher, Optionen, die nicht für sie bestimmt sind, zu überspringen. Dies ermöglicht ebenfalls eine schnellere Verarbeitung von Paketen.

- Dienstarten(QoS/CoS): IPv6 unterstützt Dienstarten und QoS/CoS. Damit kommt IPv6 den Forderungen nach einer verbesserten Unterstützung der Übertragung von Video- und Audiodaten entgegen. IPv6 bietet hierzu eine Option zur Echtzeitübertragung.

- Sicherheit: IPv6 beinhaltet im Protokoll Mechanismen zur sicheren Datenübertragung. Wichtige neue Merkmale von IPv6 sind hier Authentifikation (authentication), Datenintegrität (data integrity) und Datenverlässlichkeit (data confidentiality).

- Erweiterbarkeit: IPv6 ist ein erweiterbares Protokoll. Bei der Spezifikation des Protokolls wurde nicht versucht, alle potentiell möglichen Einsatzfelder für das Protokoll in die Spezifikation zu integrieren. Vielmehr bietet IPv6 die Möglichkeit über Erweiterungs-Header das Protokoll zu erweitern. Damit ist das Protokoll offen für zukünftige Veränderungen.

[Hol97]

Abbildung 2.4 zeigt die Form des IPv6-Datengramms mit dem Platz für die optionalen Erweiterungsheader.

Abbildung 2.5 veranschaulicht den Aufbau des IPv6-Basis-Headers. Dessen Attribute sollen im Folgenden genauer erläutert werden:

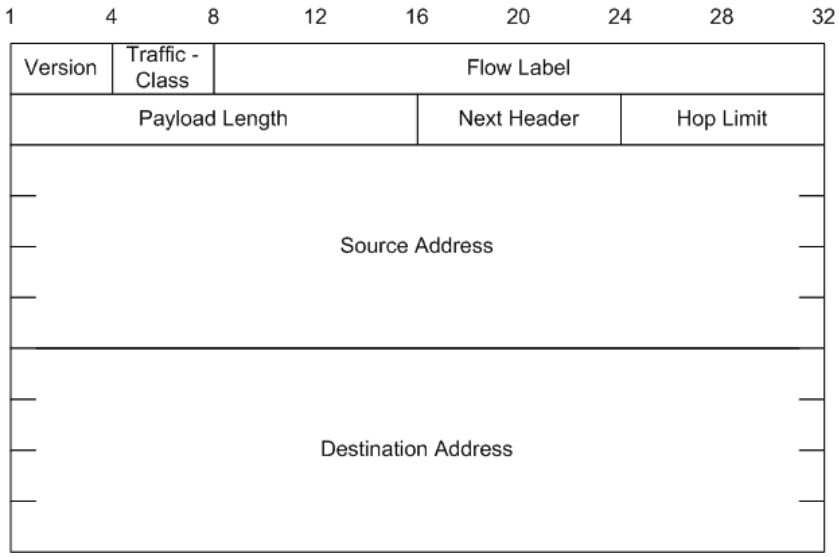

Abbildung 2.5: IPv6-Header

- Version: Wie auch bei IPv4 (siehe 2.3.1) dient das Versionsfeld der Klassifizierung der Protokollversion. Anhand dieses Feldes ist der weitere Aufbau des Headers den Netzkomponenten bekannt und es kann eine dementsprechende Weiterverarbeitung eingeleitet werden.

- Priority: (auch Traffic Class) Anhand dieses Feldes kann ein Router den Dateninhalt der Pakete in bestimmte Verkehrklassen (Videodaten, Sprachdaten) unterteilen. In Ausnahmesituationen entscheidet der Router an Hand dieses Feldes, welche Daten verworfen werden können und welche auf keinen Fall verworfen werden.

- Flowlabel: Das Flowlabel kennzeichnet mehrere Pakete eines Datenflusses, die von einer senderseitigen Applikation an eine empfangseitige Applikation verschickt werden. Anhand des Flowlabels erkennt ein Router das dazugehörige Paket eines Datenflusses und kann ohne in der Routingtsabelle nachschauen zu müssen, das Paket weiterleiten, da die Route von den vorhergehenden Paketen des Datenflusses bekannt ist.

- Payload Length: Das Feld Payload Length (Nutzdatenlänge) gibt an, wie viele Bytes dem IPv6-Basis-Header folgen. Die Erweiterungs-Header werden bei der Berechnung der Nutzdatenlänge mit einbezogen. Das entsprechende Feld wird in der Protokollversion 4 mit Total Length bezeichnet, wobei dort die gesamte Paketlänge gemeint ist.

- Next Header: Das Feld Next Header gibt an, welcher Erweiterungs-Header dem IPv6-Basis-Header folgt. Jeder folgende Erweiterungs-Header beinhaltet ebenfalls ein Feld Next Header, das auf den nachfolgenden Header verweist. Ist dies der letzte zu IPv6 zugehörige Header, so gibt das Feld an, welches Transportprotokoll (z.B. TCP oder UDP) folgt.
- Hop Limit: Mit dem Feld Hop Limit wird festgelegt, wie viele Router durchlaufen werden, bis das Paket gelöscht wird. Der Wert des Feldes wird nach jeder Teilstrecke dekrementiert. Ein Datengramm wird verworfen, wenn das Feld Hop Limit Null ist. IPv4 verwendet hierzu das Feld Time to Live.
- Source Address, Destination Address: Die beiden Felder Quell- und Zieladresse dienen zur Identifizierung des Senders und Empfängers eines IP-Datengramms. IPv6 verwendet zur Adressierung viermal so große Adressen wie IPv4: 128 Bit statt 32 Bit.

2.4 Auswertung der Darstellbarkeit der Internet-Protokolle

Für die Visualisierung des Datenstroms der Vermittlungsschicht lassen sich folgende Attribute nutzen:

Aussageattribut	IPv4	IPv6
Zieladresse	Destination Address	Destination Address
Quelladresse	Source Address	Source Address
Protokollversion	Version	Version
Payloadlänge	Total Length - Header Length	Payload Length
Alter des Pakets	Time to Live	Hop Limit
Protokoll des Payloads	Protocol	letztes Next Header Feld
Verkehrsklasse	TOS/DCSP	Priority/Traffic Class

Tabelle 2.1: Attribute der Vermittlungsschicht

Die Darstellungen die je Paket oder nach der Zeit realisierbar sind, werden in Tabelle 2.2 aufgeführt. Im Folgenden wird auf einige der möglichen Darstellungen eingegangen:

Die Darstellung der Source/ Destination Address nach der Anzahl der Pakete je Zeiteinheit kann nur Aussagen zur Belastung der CPU der Router, jedoch nur bedingt zur Ausnutzung der Bandbreite liefern, da jedes IP-Paket eine variable

darzustellende Attribute	Wertebereich	Aussage der Darstellung
Source/Destination Address	je 32/128 bit	Anzahl von Paketen je Zeiteinheit
Version	4 bit	Verteilung der Versionen je Zeiteinheit
Total Length	16 bit	Bruttobyterate je Zeiteinheit (nur IPv4)
Header Length	4bit	Verwaltungsoverhead in Bit je Zeiteinheit (nur IPv4)
Time to Live/Hop Limit	8 bit	Anzahl der Hops je Paket
letztes Next Header Feld	16 bit	Nettobyterate je Zeiteinheit
Protocol	8 bit	IANA-Nummer des Protokolls im Nutzlastanteil je Paket
TOS/DCSP / Priority/Traffic Class	8 bit	Paketverlustrate und Verzögerungsklasse je Paket

Tabelle 2.2: Attribute und Wertebereich der visualisierbaren IP-Attribute

Länge hat. Der Anstieg der Ankunftsrate von Paketen ist nicht proportional zum Anstieg der Nettodatenrate. Deshalb ist es nötig, die auswertbaren Informationen des Datenstroms noch weiter gehenden Analysen zu unterziehen.

Die in Tabelle 2.2 aufgeführten Attribute lassen sich direkt aus dem Bytestrom auslesen. Es existieren noch weitere Informationsdarsstellungen, die nicht direkt visualisiert werden können.

Die Differenz des Feldes ”Total Length” und ”Length” ergibt den Nutzdatenanteil in Bit/Byte des IP-Pakets an. Damit kann die Datenrate bezogen auf die IP-Adressen beispielsweise nicht nur in Paketen je Zeiteinheit, sondern auch in Nettobytes je Zeiteinheit dargestellt werden. Bei IPv6 kann dieser Wert direkt aus dem letzten ”Next Header”- Feld ausgelesen werden.

Diese Referenzierung ist auch nach genutzten Protokollen vorstellbar. Damit ist es möglich, die reale Verteilung der Datenrate in Byte je Zeiteinheit der einzelnen Protokolle untereinander darzustellen und nicht nur die Verteilung der Protokolle je Paket je Zeiteinheit.

2.5 Transportschicht

Die Transportschicht übernimmt den Transport von Nachrichten zwischen den Kommunikationspartnern. Die Transportschicht hat die grundlegenden Aufgaben, den Datenfluss zu steuern und die Unverfälschtheit der Daten sicherzustellen. Die Daten werden paketorientiert übertragen. Es wird zwischen zwei Verfahren

unterschieden.

- gesicherte Übertragung (Quittierung des Empfangs, wenn das Paket angekommen ist)
- ungesicherte Übertragung (keine Quittierung des Empfangs)

Beispiele für Transportprotokolle sind TCP (gesicherte Übertragung) und UDP (ungesicherte Übertragung).

2.5.1 Transmission Control Protocol (TCP)

Das Transmission Control Protocol (TCP) ist ein zuverlässiges, verbindungsorientiertes, Bytestrom-Protokoll. Die Bereitstellung eines sicheren Transports von Daten durch das paketorientierte Netzwerk ist die Aufgabe von TCP. Die Urdefinition von TCP ist im RFC 793 dokumentiert. Es wurde mit der RFC 1122 abgeändert (Fehler und Inkonsistenzen wurden entfernt). Die RFC 1323 beschreibt weitere Änderungen und ist die derzeit gültige Fassung der Protokolldefinition.

Die drei Grundeigenschaften von TCP:

- Zuverlässigkeit (reliability)
- Verbindungsorientierung (connection-oriented) und
- Bytestrom (byte-stream)

werden näher betrachtet.

Positive Acknowledgement with Retransmission (PAR) ist der Mechanismus, mit dem TCP sicherstellt, dass die gesendeten Daten beim Empfänger angekommen sind. Es bedeutet, dass die Übertragung der Daten solange wiederholt wird, bis vom Empfänger der Erhalt der Daten bestätigt wird. Die zwischen den Instanzen ausgetauschten Dateneinheiten heißen Segmente. Das TCP-Segment beginnt mit einem mindestens 20 Byte großen Protokollkopf (siehe Abbildung 2.3) und den zu übertragenden Daten. Die Konsistenz jedes Segments wird mittels einer Prüfsumme festgestellt. War die Übertragung fehlerfrei, sendet der Empfänger eine Empfangsbestätigung an den Sender. Ist dies nicht der Fall, wird das Datagramm verworfen und keine Empfangsbestätigung verschickt. Hat der Empfänger nach einer bestimmten Zeitperiode (timeout-period) keine Quittung erhalten, wird das betreffende Segment erneut versendet.

Der Aufbau der logischen Ende-zu-Ende-Verbindung wird mit Hilfe des Dreiwege-Handshakes, durch Senden von Steuerinformationen realisiert. Zum Aufbau der Verbindung sendet ein Host (Host 1) dem anderen Host (Host 2) ein Segment, in dem das SYN-Flag (siehe Abbildung 2.2) gesetzt und eine Segmentnummer eingetragen ist. Host 2 kann den Verbindungswunsch mit Senden eines Segments, bei dem SYN und ACK-Bit gesetzt sind, sowie mit der inkrementierten Segmentnummer im Bestätigungsfeld an Host 1 bestätigen. Damit ist Host 1 bekannt, dass Host 2 die Verbindung annimmt. Host 1 bestätigt die Verbindung gegenüber Host 2 damit, dass die von Host 2 empfangene Sequenznummer inkrementiert im Bestätigungsfeld und dem gesetzten ACK-Bit an Host 2 übersendet wird. Bei diesem dritten Handshake werden die ersten Daten übertragen.

Zum Beenden einer Verbindung tauschen die beiden Hosts wiederum ein Dreiwege-Handshake aus, bei dem das FIN-Bit (siehe Abbildung 2.2) zum Beenden der Verbindung gesetzt ist.

TCP nimmt Datenströme von Applikationen der darüber liegenden Schichten an und segmentiert diese in höchstens 64 KByte große Teile (üblich sind ca. 1.500 Byte). Jedes dieser Segmente wird als IP-Datagramm verschickt. Werden IP-Datagramme mit TCP-Daten aus den darunter liegenden Schichten empfangen, werden diese an TCP weitergeleitet und zu den ursprünglichen Bytetströmen verkettet. Die Vermittlungsschicht gibt allerdings keine Gewähr dafür, dass die Datagramme richtig zugestellt werden. Es ist deshalb, wie schon erwähnt, die Aufgabe von TCP für eine erneute Übertragung korrupter Datagramme zu sorgen. Wenn IP-Datagramme in der falschen Reihenfolge auflaufen, muss TCP dafür sorgen, dass die Daten in die richtige Reihenfolge gebracht werden. Dies stellt TCP mit Sortierung der Segmente nach Sequenz- und Bestätigungsnummer sicher.

Eine weitere Aufgabe von TCP ist es, die Datenströme an die richtigen Applikationen weiterzuleiten. Diese Adressierung der Anwendungen findet auf der Transportebene mit den Portnummern statt. Das entsprechende Adressierungsfeld im TCP-Header ist 16 Bit groß. Damit wäre es einem Host möglich, theoretisch bis zu 65.535 verschiedene TCP-Verbindungen aufbauen zu können. Auch das verbindunglose UDP verwendet Portnummern zur Adressierung. TCP und UDP können die gleichen Portnummern belegen. Das heißt, dass die Portnummer 53 in TCP nicht identisch mit der Portnummer 53 in UDP ist. Der Gültigkeitsbereich einer Portnummer ist auf einen Host beschränkt.

Der Adressbereich des TCP/UDP ist in drei Bereiche aufgeteilt:

- 0-1023 - Well-known ports (von der IANA verwaltet).

- 1024-49151 - Registered ports. Registrierte Ports dienen für Dienste, die üblicherweise auf bestimmten Ports laufen.

- 49152-65535 - Dynamic and/or private ports. Dieser Bereich ist für die so genannten dynamischen Ports festgelegt. Dynamische Ports dienen zur Kommunikation zwischen den beiden TCP-Schichten, die an einer Kommunikation beteiligt sind. Ein dynamischer Port wird nicht von bestimmten Standarddiensten belegt.

Die Verwaltung zur Belegung der Ports hat die Internet Corporation for Assigned Names and Numbers (IANA) übernommen. Werden neue Anwendungen entwickelt, die einen bestimmten Port aus dem zweiten Bereich benutzen möchten, muss dies bei der IANA angemeldet werden.

2.5.2 Der TCP-Header

Am Beginn jedes Segments steht ein 20 Byte langer Header. Wie beim Internet-Protokoll ist es hier möglich, dass ein Feld mit Optionen folgt. Nach dem Optionsfeld folgen die Daten der darüber liegenden Schicht, der eigentliche Nutzdatenanteil. Die Größe eines Segments wird durch zwei Faktoren begrenzt: jedes Segment muss in das Nutzdatenfeld des IP-Protokolls passen (65.535 Byte); zweitens darf die maximale Transfereinheit (MTU - Maximum Transfer Unit) nicht überschritten werden. Allgemein ist die MTU einige tausend Byte groß und gibt damit das Maximum der Segmentgröße vor (z.B. Ethernet 1.500 Bytes). Werden vom einem Segment auf dem Weg durch verschiedene Netze, Netze mit einer geringeren MTU durchlaufen, müssen die Segmente fragmentiert werden.

Unabhängig von der Größe der MTU können dem TCP-Header und den Optionen maximal 65.535-20-20 = 65.495 Datenbyte folgen (die ersten 20 Byte beziehen sich auf den IP-Header, die zweiten auf den TCP-Header; die Länge der Optionen wird mit zu den Datenbytes gezählt). TCP-Segmente ohne Daten sind zulässig und dienen der Übermittlung von Bestätigungen und Steuernachrichten.

Source-/Destination-Port: Die Felder Source Port (Quellport) und Destination Port (Zielport) adressieren die Endpunkte der Verbindung. Die Größe der beiden Felder beträgt 16 Bit.

Sequence Number, Acknowledgement Number: Die Sequenznummer und die Bestätigungsnummer sind jeweils 32-Bit-Zahlen. Die Nummern geben die Stellung der Daten des Segments innerhalb des in der Verbindung ausgetauschten

Datenstroms an. Die Sequenznummer gilt in Senderichtung, die Bestätigungsnummer für Empfangsquittungen. Jeder der beiden TCP-Verbindungspartner generiert beim Verbindungsaufbau eine Sequenznummer, die sich während des Zeitraums der Verbindung nicht wiederholen darf. Dies ist durch den Adressraum von 32 Bit gesichert. Diese Nummern werden beim Verbindungsaufbau ausgetauscht und wechselseitig quittiert. Bei der Datenübertragung wird die Sequenznummer vom Absender jeweils um die Anzahl der bereits gesendeten Segmente erhöht. Mit der Quittungsnummer gibt der Empfänger an, bis zu welcher Segmentnummer er die Daten bereits korrekt empfangen hat. Die Nummer gibt allerdings nicht an, welches Segment zuletzt korrekt empfangen wurde, sondern welches Segment als nächstes zu erwarten ist. Dieses Verfahren wird Windowing genannt.

Offset: Das Feld Offset (auch Header Length) gibt die Länge des TCP-Headers in 32-Bit Worten an. Dies entspricht dem Anfang der Daten im TCP-Segment. Das Feld ist notwendig, da der Header durch das Optionsfeld eine variable Länge hat.

Flags: Mit den sechs 1-Bit-Flags im Flags-Feld werden bestimmte Aktionen im TCP-Protokoll aktiviert:

- URG: Ist das Flag URG gesetzt, bedeutet dies, dass der Urgent Pointer (Dringendzeiger) verwendet wird.

- ACK: Das ACK-Bit wird gesetzt, um anzugeben, dass die Bestätigungsnummer im Feld Acknowledgement Number gültig ist. Ist das Bit nicht gesetzt, wird das Feld Acknowledgement Number ignoriert.

- PSH: Ist das PSH-Bit gesetzt, so werden die Daten in dem entsprechenden Segment sofort bei Ankunft der adressierten Anwendung bereitgestellt ohne sie zu puffern.

- RST: Das RST-Bit dient dazu, eine Verbindung zurückzusetzen, falls ein Fehler bei der Übertragung aufgetreten ist. Dies kann der Fall sein, wenn ein ungültiges Segment übertragen wurde, ein Host abgestürzt ist oder der Versuch eines Verbindungsaufbaus abgewiesen werden soll.

- SYN: Das SYN-Flag (Synchronize Sequence Numbers) wird zum Verbindungsaufbau genutzt. Mit der Acknowledgement Number und dem ACK-Bit wird die Verbindung in Form eines Dreiwege-Handshakes aufgebaut (siehe oben).

- FIN: Das FIN-Bit dient zum Auslösen einer Verbindung. Ist das Bit gesetzt, gibt dies an, dass der Sender keine weiteren Daten zu übertragen hat. Das Segment mit gesetztem FIN-Bit muß quittiert werden.

Window: Das Feld Fenstergröße enthält die Anzahl der Segmente, die der Empfänger ab dem bereits bestätigten Segment empfangen kann. Mit der Angabe der Fenstergröße erfolgt in TCP die Flusssteuerung. Das TCP-Protokoll arbeitet nach dem Prinzip eines Schiebefensters mit variabler Größe (Sliding Window). Jede Seite einer Verbindung darf die Anzahl der Segmente senden, die im Feld für die Fenstergröße angegeben ist, ohne auf eine Quittung von der Empfangsseite zu warten. Während des Sendens können gleichzeitig Quittungen für die von der anderen Seite empfangenen Daten eintreffen (diese Quittungen können die Fenstergrößen verändern).

Checksum: Die Prüfsumme umfasst den Header, die Daten und den Pseudo-Header. Der Pseudo-Header enthält die 32-bit großen IP-Adressen des Quell- und Zielhosts sowie die Protokollnummer (für TCP 6) und die Länge des TCP-Segments. Die Einbeziehung der Felder des Pseudo-Headers in die Prüfsummenberechnung hilft, durch IP falsch zugeteilte Pakete zu erkennen. Die Verwendung von IP-Adressen auf der Transportebene stellt allerdings eine Verletzung der Protokollhierarchie dar.

Urgent Pointer: Der Urgent-Pointer ergibt zusammen mit der Sequenznummer einen Zeiger auf ein Datenbyte. Dies entspricht einem Byteversatz zu einer Stelle, an der dringende Daten vorgefunden werden. TCP signalisiert damit, dass sich an einer bestimmten Stelle im Datenstrom wichtige Daten befinden, die sofort gelesen werden sollten. Das Feld wird nur gelesen, wenn das Urgent-Flag gesetzt ist.

Options: Das Options-Feld soll eine Möglichkeit bieten, Funktionen bereitzustellen, die im normalen TCP-Protokollkopf nicht vorgesehen sind. In TCP sind drei Optionen definiert: End of Option List, No-Operation und Maximum Segment Size. Die wichtigste dieser Optionen ist die maximale Segmentgröße. Mit dieser Option kann ein Host die maximale Anzahl Nutzdaten übermitteln, die er annehmen will bzw. annehmen kann. Während eines Verbindungsaufbaus kann jede Seite ihr Maximum an Nutzdaten übermitteln. Der kleinere der beiden Werte wird als maximale Nutzdatengröße für die Übertragung übernommen.

Padding: Das Feld Padding wird verwendet, um sicherzustellen, dass der Header an einer 32-Bit Grenze endet und die Daten an einer 32-Bit Grenze beginnen. Das Füllfeld wird mit Nullen gefüllt.

2.6 Auswertung der Darstellbarkeit des TCP

Die folgenden Attribute sind für die Visualisierung der Datenströme des Transmission Convergence Protokolls relevant:

- Source/Destination Port
- Offset
- ACK/SYN/FIN/RST/UGR/PSH
- Window
- Options MTU

Diese Attribute sind als direkte Darstellungsattribute verfügbar, welche vor der Darstellung nicht weiterverarbeitet werden müssen.

- Anzahl der Segmente je Source/Destination Port je Zeiteinheit
- Anzahl der Segmente, nach denen eine Bestätigung erfolgen muss (window) je Segment
- Größe der Verwaltungsoverheads je Segment (offset)

Eine Weiterverarbeitung der Parameter der Messdaten mit Hilfe bestimmter Algorithmen ist für folgende Darstellungen nötig:

- erfolgreiche Verbindungen je Zeiteinheit (3 Wege-Handshake bei erfolgreichem Verbindungsauf- und abbau) Dabei müssen die bestehenden Verbindungen beachtet werden, die die Grenzen der Zeiteinheit überschneiden.
- Anzahl der gescheiterten Verbindungen je Zeiteinheit (3 Wege-Handshake bei Verbindungsauf- oder abbau nicht erfolgreich oder RST wurde gesetzt)
- Nettobyterate je Source/Destination Port je Zeiteinheit (berechnet unter Beachtung der gültigen MTU des aktuellen Segments)

Kapitel 3

VISUALISIERUNG WISSENSCHAFTLICHER DATEN

Sensoren und Simulationen erzeugen eine Unmenge von Daten. Es ist für einen Menschen nur schwerlich möglich, lange Datenreihen zu überschauen, zu analysieren und Gesetzmäßigkeiten in Messergebnissen zu erkennen. Mit Hilfe der Visualisierung, der Darstellung der Daten auf andere Weise als in Spalten oder Reihen ist es dem Menschen einfacher, die Ergebnisse der Simulationen oder Messwerte zu verstehen.

Die Zielstellung der Visualisierung wissenschaftlicher Daten ist es, eine leichter verständliche visuelle Darstellung der Datenmenge und somit eine effektivere Auswertung der Daten zu ermöglichen. Damit können u. a. erstellte Konzepte leichter analysiert werden, womit auch das Verständnis von Modellen erleichtert wird.

3.1 Zielstellung

Mit der Visualisierung wird die Darstellung von komplexen Sachverhalten erleichtert; es werden die Tatsachen so dargestellt, dass sie nicht nur gesehen sondern auch erkannt werden. Es muss dem Nutzer nach dem Betrachten möglich sein, den visualisierten Sachverhalt verstehen und bewerten zu können. Aus der Vielzahl der möglichen Visualisierungstechniken sind die Verfahren auszuwählen und anzuwenden, die diese Aufgabe am besten erfüllen. Es ist auf jeden Fall zu beachten, dass die darzustellenden Daten aufgrund der gewählten Visualisierungsform nicht zu verzerrt abgebildet werden, so dass sie nicht mehr das Darstellen, was sie eigentlich aussagen.

Die Visualisierung wissenschaftlicher Daten kann in drei Schritten des Analyseprozesses eingesetzt werden. [SM00]

1. explorative Analyse

2. konfirmative Analyse

3. Präsentation

Die **explorative Analyse** ist die Reindarstellung der Daten. Über die eventuelle Struktur oder eine bestimmte Ausrichtung der Daten ist zu diesem Zeitpunkt der Untersuchung noch nichts bekannt. Ausgehend von dieser Darstellung kann mittels Interaktion, Manipulation von Darstellungsparametern nach Informationen, Mustern, Strukturen oder Zusammenhängen gesucht werden. Dieser Schritt der wissenschaftlichen Visualisierung ist ein Prozess zur Generierung von Hypothesen. Eine weitere Untersuchung dieser Hypothesen ist mit Hilfe weiterer visuell-explorativer Prozesse, auch mittels statistischer Methoden oder Verfahren der künstlichen Intelligenz möglich. Die visuell-explorativen Verfahren haben den Vorteil, dass sie auch inhomogene und stark verrauschte Daten verarbeiten können und der Anwender keine speziellen mathematisch-statistischen Kenntnisse haben muss, um diese Verfahren einzusetzen.

Mittels der **konfirmativen Analyse** können gewonnene Hypothesen überprüft und gegebenenfalls bekräftigt oder verworfen werden. Hierbei ist eine geeignete Visualisierungsform zielgerichtet einzusetzen.

Die **Präsentation** der gewonnenen Erkenntnisse und damit die Kommunikation mit anderen ist ein weiteres Anwendungsgebiet der wissenschaftlichen Visualisierung. Die aus den vorangegangenen Prozessen erzielten Erkenntnisse müssen für Dritte an dieser Stelle als erkennbare Fakten effektiv, einfach und verständlich dargestellt werden.

3.2 Anforderungen an die Visualisierung

Die Erzeugung von Abbildungen oder Reihen von Abbildungen, welche bestimmte Eigenschaften der Datenmengen darstellen, ist das Ziel des Visualisierungsprozesses. Dafür werden bestimmte Eigenschaften der Datenmenge auf visuell-perzeptive Attribute abgebildet. Diese Transformation des Datenraumes auf den visuellen Raum kann elementar sein, wie z.B. eindimensionale Abbildung eines Zahlenwertes auf eine Farbe. Je höher der Freiheitsgrad der Datenmenge ist, umso komplexer kann auch die Visualisierung werden; z.B. Visualisierung eines Wirbelsturms in drei Dimensionen verbunden mit Animationen.

Bertin [Ber82] gibt drei Level der Informationsdichte an, die in einer Visualisierung dargestellt werden können. Die einfachste Art bildet die vorliegenden Informationen des Datenraumes in direkter Form auf den visuellen Raum ab. Somit ist für jede Information des Datenraums eine bestimmte spezifische Ausbildung eines Attributs des visuellen Abbildungsraums vorhanden. Bei der zweiten Art steigt der Abstraktionsgrad des Abbildungsraums; es wird schon das Wesentliche,

die Ergebnisse der Untersuchung verdeutlicht. Die Darstellung der Gesamtheit aller Daten und der Informationen, die sich aus der Visualisierung ergeben, ist das Hauptziel des Visualisierungsprozesses. Dieser dritte Level ist die Grundlage für Entscheidungen, es enthält immer auch die beiden anderen Level. Weiterhin werden auf diesem Level die Erkenntnisse und Schlussfolgerungen des Analyseprozesses in anschaulicher Form dargestellt.

Inzwischen hat die Visualisierung wissenschaftlicher Daten in allen Anwendungsgebieten einen hohen Stellenwert eingenommen, jedoch wird auf die Qualität und Effektivität der erzeugten Darstellungen in vielen Fällen nicht geachtet. So wird die komplette Bandbreite der Darstellungsmöglichkeiten moderner Darstellungstools vollständig ausgenutzt. Die Frage nach der expressivsten Darstellungsform wird nur selten gestellt. Eine überladene und zu redundante visuelle Darstellung kann in Einzelfällen zu Fehlinterpretationen und somit zu Fehlentscheidungen führen. Dies muss dem Urheber der Visualisierung immer bewusst sein.

Die Qualität einer visuellen Präsentation ist das Maß, in dem der Betrachter fähig ist, den Kontext der realen Welt aus der Abbildung zu rekonstruieren, die wahrgenommen Strukturen in der Abbildung mit tatsächlich existierenden Korrelationen zwischen Parametern bzw. Störungen in den Ergebnisdaten in Verbindung zu setzen und daraus die richtigen Entscheidungen zu ziehen.

3.3 Die explorative Analyse

Shneiderman [Shn96] untergliedert die explorative Analyse in drei Schritte:

- overview
- zoom and filter
- details-on-demand

Er bezeichnet diese drei Schritte auch als Information Seeking Mantra [Shn96]. Zur visuellen Analyse bedarf es dem Nutzer an erster Stelle einen Überblick über die Daten (overview).

Das Erkennen von Mustern oder Unregelmäßigkeiten wird mit Hilfe des zweiten Schrittes von Zoom- und Selektionstechniken (zoom and filter) unterstützt. Für die detaillierte Analyse der Muster wird dem Nutzer der dritte Schritt, die Möglichkeit, die Details der Daten zu betrachten (details-on-demand) bereitgestellt.

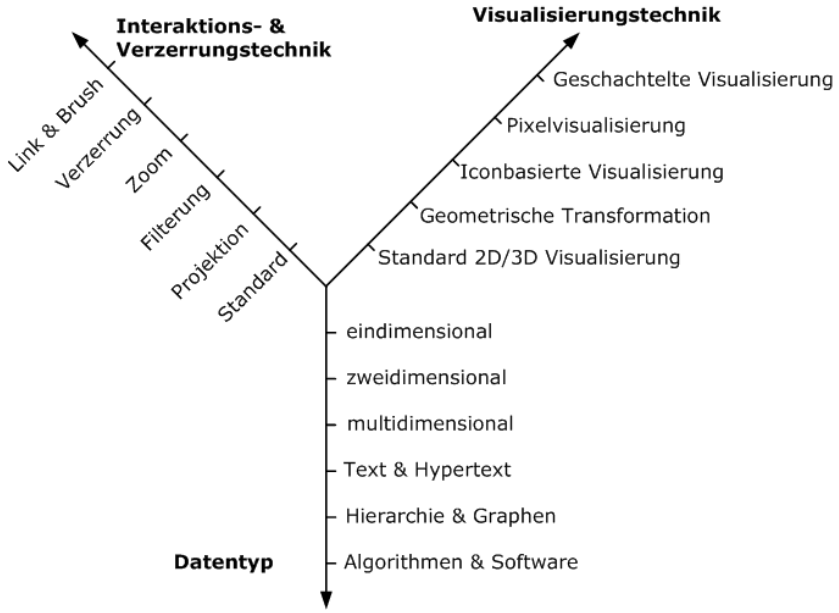

Abbildung 3.1: Klassifikation der explorativen Analyse [Kei01]

In allen drei Schritten der explorativen Analyse werden Visualisierungstechniken eingesetzt: Mit Hilfe dieser Techniken ist es einfach, einen Überblick über die Datenmenge zu erstellen und es ist dem Nutzer möglich, interessante Teilmengen in der Darstellung zu entdecken. Es ist notwendig, während des Betrachtens der Details den Überblick über die gesamte Datendarstellung zu behalten. Es muss dem Beobachter klar sein, wo er sich in der Visualisierung befindet. Dieser Überblick ist mit Hilfe der interaktiven Verzerrung der visuellen Überblicksdarstellung bezüglich der Fokusse möglich.

Für eine weiter gehende Exploration interessanter Teilmengen bedarf es zusätzlicher Optionen, um die Daten genauer zu betrachten und Details zu verstehen. Dafür ist es wichtig, dass die genutzten Visualisierungstechniken über die grundlegenden Visualisierungsverfahren hinaus, die Schwierigkeiten bei den Übergängen zwischen diesen Schritten überbrücken helfen.

3.4 Klassifikation der explorativen Analyse

Informations-Visualisierung wissenschaftlicher Daten mit mehr als einem Attribut, die keine 2D- oder 3D-Semantik besitzen, sind mittels X-Y-Plots, Liniendiagrammen und Histogrammen darstellbar. Diese Techniken können für die visuelle Datenexploration hilfreich sein, jedoch sind sie im Allgemeinen auf relativ klei-

ne und niedrigdimensionale Datenmengen beschränkt. In den vergangenen Jahren wurde eine Vielzahl neuartiger Techniken für hochdimensionale Datenmengen ohne interne 2D- oder 3D-Semantik entwickelt. Ein Überblick über diese Verfahren wird in [CMS99] [War00] [Spe00] [SM00] gegeben. Die Techniken können anhand folgender drei Kriterien klassifiziert werden [Kei01] (siehe Abbildung 3.1) :

1. der zu visualisierende Datentyp,
2. die verwendete Visualisierungstechnik,
3. die verwendeten Techniken für Interaktion und Verzerrung

Der zu visualisierende Datentyp kann folgendermaßen unterteilt werden [Shn96]:

- Ein-dimensionale Daten (zeit-abhängige Daten) - ThemeRiver Visualisierung [HHNW02] zeitabhängiger Nachrichten in Abbildung 3.3
- Zwei-dimensionale Daten (geographische Karten) - Gridfit Visualisierung [KH98] von Telefondaten in Abbildung 3.4
- Multi-dimensionale Daten (tabellarische Daten aus relationalen Datenbanken) - vgl. Parallele Koordinaten Visualisierung [ID90] in Abbildung 3.5
- Text und Hypertext (Nachrichten oder Web-Dokumente) - ThemeView Visualisierung [WTP+95] [Wis99] von Text Dokumenten in Abbildung 3.6
- Hierarchien und Graphen (Telefon- oder Internetverbindungen) - Skitter Visualisierung [HBkc+01] in Abbildung 3.7
- Algorithmen und Software (Debugging-Operationen) Tarantula Software Visualisierung [EHJS01] in 3.8

Die Visualisierungstechniken können in folgende Gruppen unterteilt werden:

- Standard 2D-/3D-Visualisierungen (Balkendiagramme oder X-Y-Diagramme) - Abbildung 3.4
- Geometrische Transformationen (künstliche Landschaften [Wis99]) - Abbildung 3.6 und Parallele Koordinaten Visualisierung [ID90] - Abbildung 3.5
- Iconbasierte Visualisierungen (Strichmännchen-Visualisierung [PG88]) - Abbildung 3.10

- Pixel-Visualisierungen (Recursive Pattern oder Circle Segments Techniken [Kei00]) - Abbildung 3.2 und Abbildung 3.12

- Geschachtelte Visualisierungen Treemaps [Shn92] [JS91]) - Abbildung 3.14 oder Dimensional Stacking [War94] - Abbildung 3.13.

Die Interaktions- und Verzerrungstechniken stellen die dritte Kategorie der Klassifikation der explorativen Analyse nach Shneiderman dar. Sie erlauben es dem Benutzer, direkt mit den Darstellungen zu interagieren. Interaktions- und Verzerrungstechniken können wie folgt untergliedert werden:

- Interaktive Projektion (GrandTour System [Asi85])

- Interaktive Selektion (Polaris System [STH02])

- Interaktives Zooming (Spotfire System [Shn99])

- Interaktive Verzerrung (Hyperbolic Tree [LRP95] [MB95])

- Interaktives Linking and Brushing (XGobi System [SCB92], [BSC96])

Die drei Dimensionen der Klassifikation nach Shneiderman können als orthogonal betrachtet werden. Zu jeder visualisierenden Datenmenge kann eine beliebige Visualisierungstechnik in Verbindung mit einer beliebigen Interaktions- und Verzerrungstechnik verwendet werden.

3.5 Datentypen

Die Daten, die in den verschiedensten Bereichen gemessen, beobachtet oder berechnet werden, unterscheiden sich in der Struktur, Dimension und anderen Eigenschaften. Die Dimension gibt an, wie viel Attribute ein Datensatz hat. Beispielsweise bei der Messung der Temperatur im Tagesverlauf und der späteren Darstellung dieser hat ein Datensatz genau zwei Attribute. Werden noch weitere Daten erfasst, wie z.B. Luftdruck oder Luftfeuchtigkeit, erhöht sich die Anzahl der Dimensionen auf vier. Dadurch erhöht sich die darzustellende Datenmenge. Der andere Faktor, welcher die Datenmenge noch beeinflusst, ist die Anzahl der Datensätze, beispielsweise die Anzahl der Beobachtungspunkte. Im Folgenden wird auf einige spezielle Datentypen eingegangen.

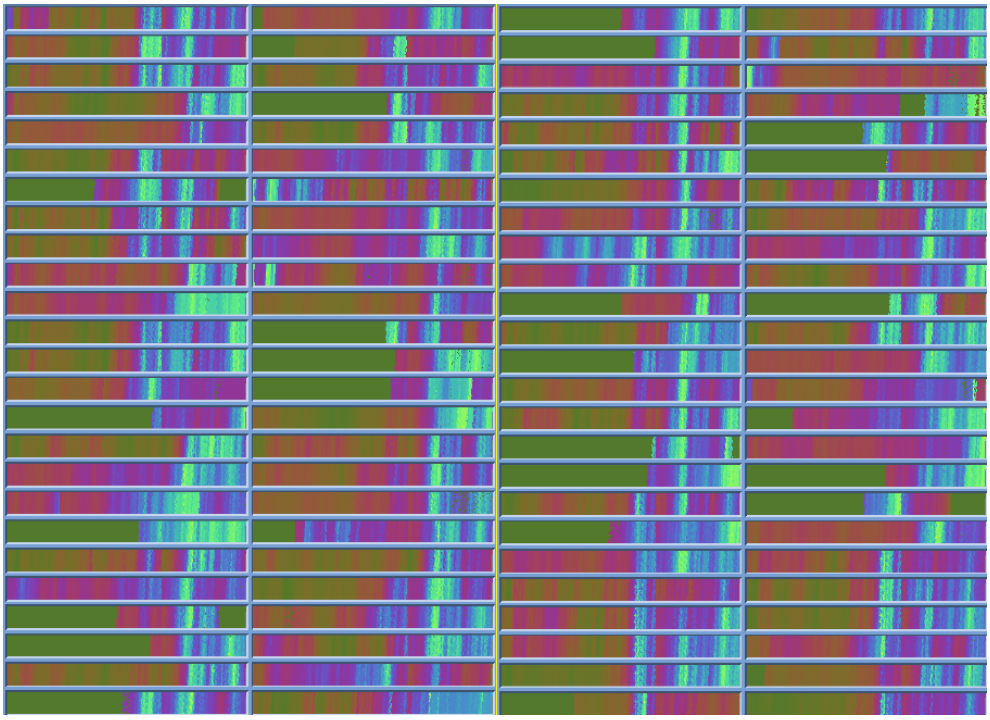

Abbildung 3.2: Die Recursive Pattern Technik [KKA95] (3.5.1)

3.5.1 Eindimensionale Daten

Beispiele für eindimensionale Daten sind die Kurscharts von Aktien (Recursive Pattern Visualisierung [KKA95] in Abbildung 3.2). Diese basiert auf einer rekursiven Darstellung einer zeilen- und spaltenorientierten Anordnung der Pixel. Die Werte des Attributs werden auf einen Farbraum projiziert. Die Anordnung jedes einzelnen Attributs ist gleich und folgt einem festgelegten System auf allen Rekursionsstufen. Jedes n-te Pixel in einer Rekursionsstufe hat die gleiche Anordnung wie das n-te Pixel einer anderen Rekursionsstufe. Die Darstellung zeigt die Tagesschlusskurse der 100 Aktien des FAZ-Index in einem Zeitraum von Januar 1974 bis April 1995. Die Projizierung der Kurswerte auf die Farbskala wurde so gewählt, dass helle Farben hohen Kursen und dunkle Farben niedrigen Kursen entsprechen.

Die ThemeRiver Visualisierungstechnik [HHNW02] in Abbildung 3.3 stellt die inhaltlichen Veränderungen von Textdokumenten in Datenbanken über die Zeit dar. Der betrachtete Zeitraum ist auf der X-Achse aufgetragen. Die alternierende Breite der verschiedenen Themengebiete stellt die Änderungen im Aufkommen der Textdokumente dar. In diesem Beispiel zeigt das Archiv der Nachrichtenagentur Associate Press die Tickermeldungen von Juni-Juli 1990. Eine starke Höhe zu

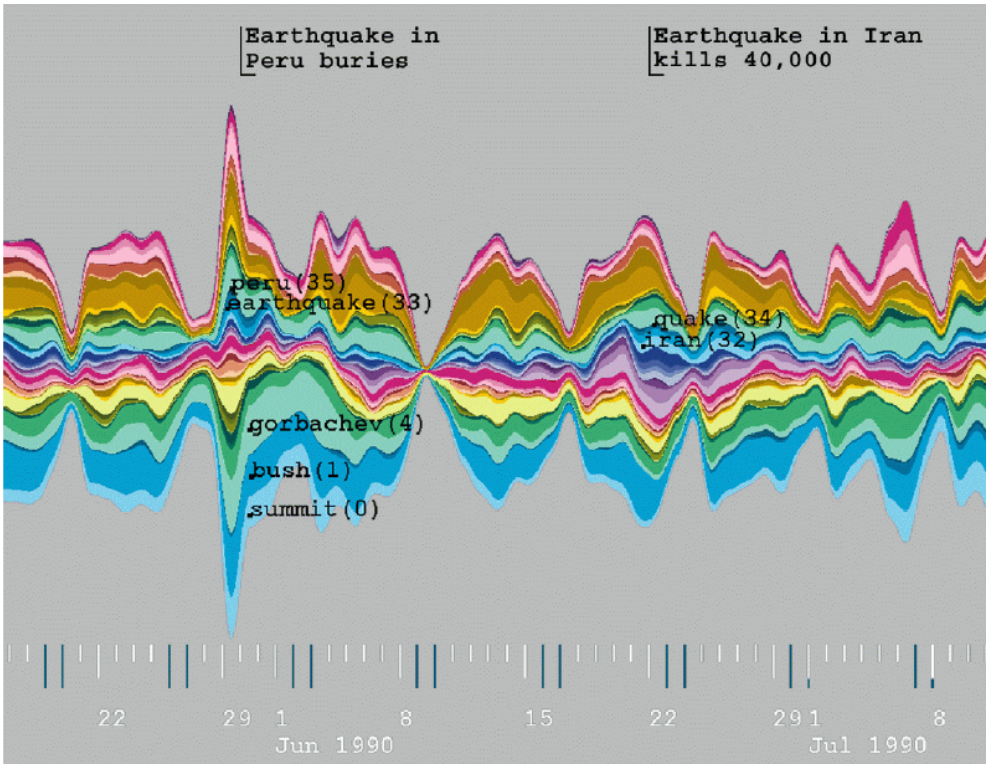

Abbildung 3.3: ThemeRiver Visualisierungstechnik (sh. 3.5.1)

den Themen Erdbeben in Peru und dem Gipfeltreffen zwischen Bush und Gorbatschow zeigen, dass dort viele Meldungen veröffentlicht wurden. Das Typische für eindimensionale Daten ist, dass sie meist ein kontinuierliches Attribut besitzen. Die genannten Beispiele (zeitabhängige Daten) sind typische Vertreter dieses Datentyps.

3.5.2 Zweidimensionale Daten

Die Beschreibung eines Wertes im Darstellungsraum obliegt bei zweidimensionalen Daten zwei Attributen eines Datensatzes. Die einfachste Darstellungsform dieses Datentyps sind normale X-Y-Diagramme. Geographische Karten stellen eine Sonderform der Visualisierung zweidimensionaler Daten dar. Eine hohe Anzahl von Datensätzen kann die Darstellung ortsabhängiger Daten verfälschen, da es in der Übersichtsansicht Überlagerungen von dargestellten Werten auf der Karte geben kann. Die Gridfit Visualisierungstechnik [KH98] (Abbildung 3.4) ist ein Beispiel dafür, wie die Pixel speziell überlagert werden, damit sich die geographische Position der Pixel nur minimal verändert. Die Abbildung zeigt Telefon-Anruf-Volumen-Daten in den USA.

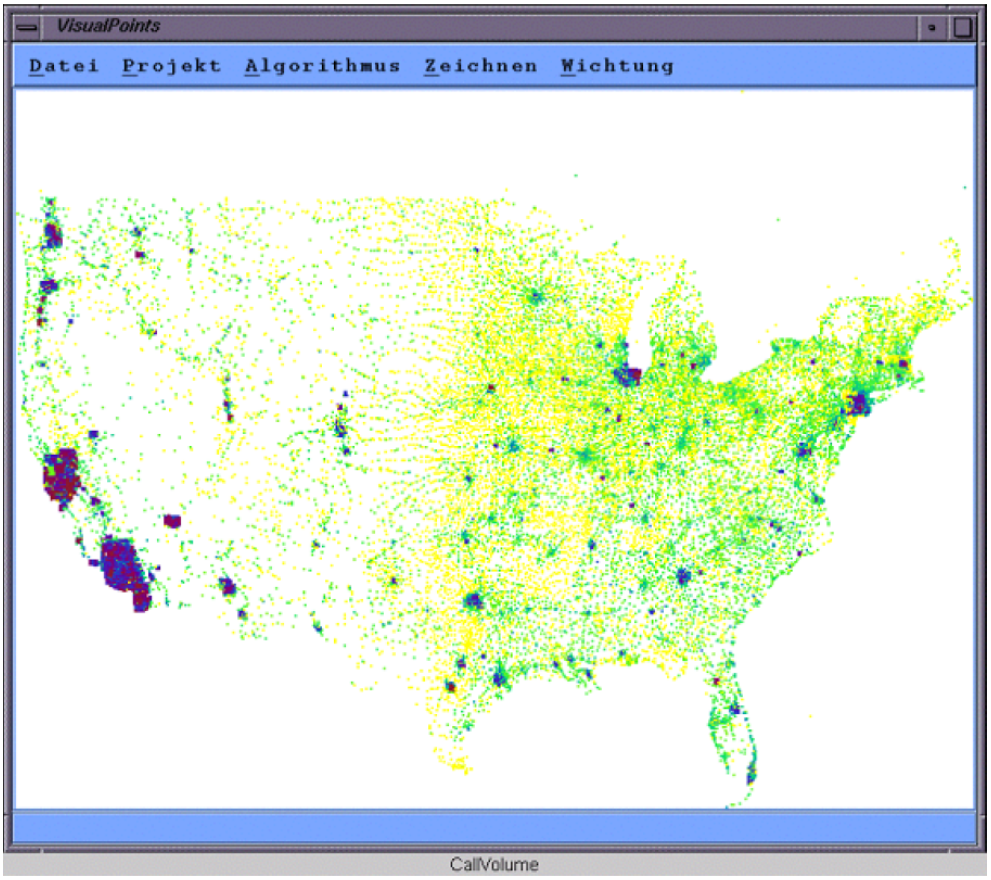

Abbildung 3.4: Gridfit Visualisierungstechnik (sh. 3.5.2)

3.5.3 Mehrdimensionale Daten

Viele Daten des Beobachtungsraumes haben mehr als drei Attribute und sind damit nicht mittels 2D- oder 3D-Visualisierungen darstellbar. Ein Beispiel für mehrdimensionale Daten sind Tabellen in relationalen Datenbanken, die oft hunderte von Attributen besitzen. Diese Daten können nicht direkt in den 2D- oder 3D-Darstellungsraum projiziert werden. Ein Ausweg ist die Parallele Koordinaten Technik [ID90] in Abbildung 3.5. Jede der senkrechten parallelen Linien bildet ein Attribut aus dem k-dimensionalen Datenraum ab. Die Wertigkeit des Attributs ist der Schnittpunkt mit der polygonalen Linie, die alle Senkrechten miteinander verbindet. Durch das Ausblenden von Datensätzen (jeder Datensatz ist eine polygonale Linie) nach bestimmten Gesichtspunkten können schnell Gemeinsamkeiten vieler Datensätze gefunden werden.

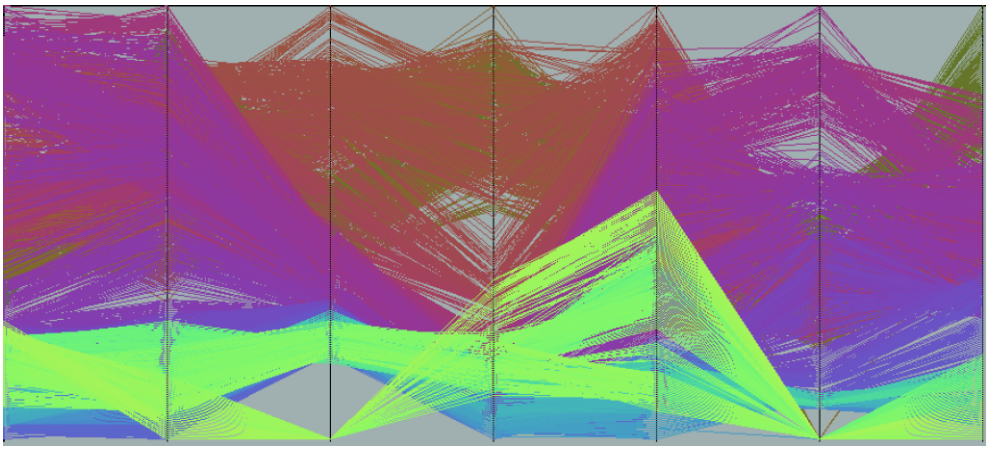

Abbildung 3.5: Parallele Koordinaten Technik (sh. 3.5.3)

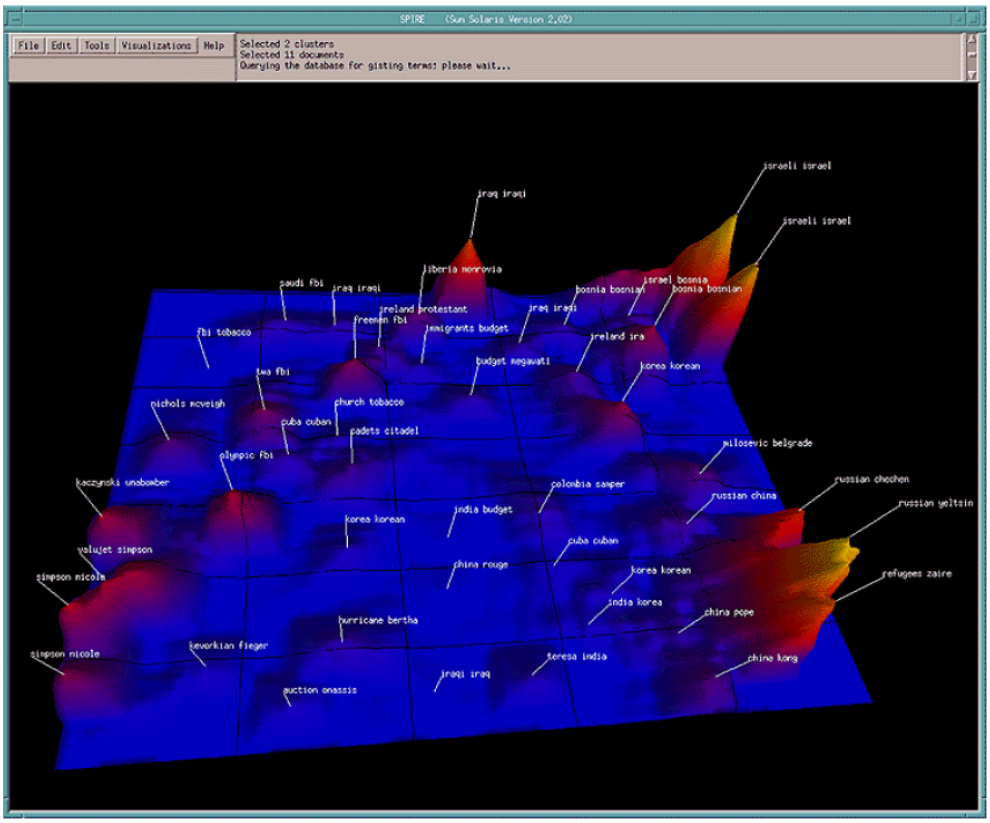

Abbildung 3.6: Die ThemeView^{TM} Visualisierungstechnik (sh.3.5.4)

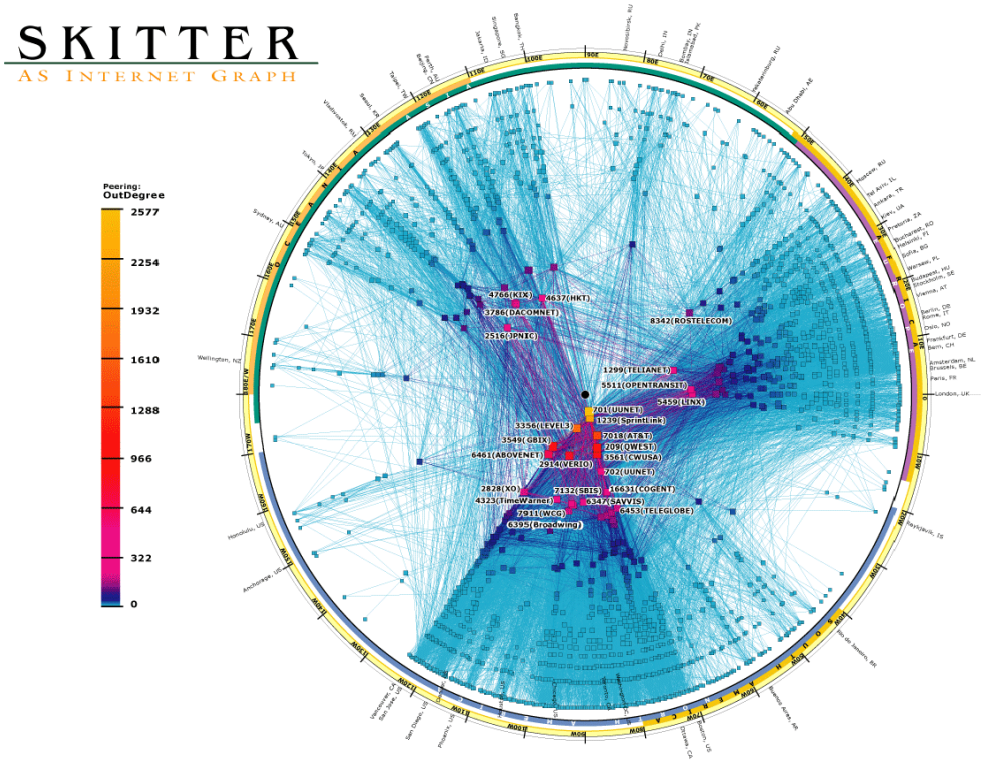

Abbildung 3.7: Graphdarstellung im Skitter-Projekt (sh. 3.5.5)

3.5.4 Texte und Hypertext

Bei der Darstellung dieser Datentypen besteht das Problem, dass es meist keine sinnvoll auswertbaren Attribute gibt. Denn die Darstellung nach Länge der Dokumente kann keinen Überblick darüber geben, ob bestimmte Themen gehäuft in der Dokumentensammlung vorliegen. Es muss eine Beschreibung bestimmter vorher festgelegter Merkmale vorliegen. Diese Beschreibung wird in so genannte Featurevektoren transformiert. Dies kann per Hand durch die Klassifizierung der Dokumente nach Schlagwörtern geschehen. Auch ein automatisiertes Zählen der nichttrivialen Suchbegriffe und eine Darstellung nach den gesuchten nichttrivialen Wörtern in einer 2D- oder 3D-Darstellung sind geeignete Visualisierungsformen, wie sie in der ThemeView^{TM}-Visualisierung [WTP+95] angewendet werden (Abbildung 3.6). Sie stellt eine Vielzahl von Textdokumenten als eine Landschaft dar; die Berge visualisieren eine Themenhäufung in verschiedenen Dokumenten der Datenbank.

3.5.5 Hierarchien und Graphen

Eine Sonderform der mehrdimensionalen Datentypen sind Hierarchien und Graphen. Die Komplexität und Abhängigkeit der einzelnen Datensätze untereinander sind die Gründe für die Sonderstellung. Kanten (Verbindungen) können die Abhängigkeiten der Knoten (Datensätze) gut darstellen. Die Hierarchien werden mittels Verknüpfung von Graphen visualisiert, wobei nur Verbindungen vom Vaterknoten zu Sohnknoten (nicht Sohn -Sohn) erlaubt sind. Typische Darstellungen in diesem Sinne sind Baumstrukturen, die je nach Anzahl der Sohnknoten in 2D- oder 3D- dargestellt werden. Beispiele dafür sind Dateisystemstrukturen auf Festplatten oder die Darstellung der Weisungsbefugnis in Organisationen. Weiterführende Literatur zu graphbasierten Darstellungstechniken sind in [Che99] und zur Darstellung des Internets in [Dod01] zu finden. Auf Algorithmen zur Graphdarstellung wird in [BETT99] vertiefend eingegangen. Abbildung 3.7 zeigt die Visualisierungstechnik des Skitter- Projektes [HBkc+01]. Es zeigt den globalen Internet-Graphen. Die Kreisanordnung stellt die Hauptknoten fokussiert im Mittelpunkt dar. Je weiter man sich dem Rand nähert, desto mehr verlieren die Knoten in Bezug auf die Größe der Datenrate an Bedeutung.

3.5.6 Algorithmen und Software

Abbildung 3.8 zeigt die Bildschirmdarstellung des Programms Tarantula. Die Übersicht über den Entwicklungsstand von größeren Softwareprojekten zu behalten, ist schwierig. Die Visualisierung zeigt dem Projektleiter eine Übersicht über den Stand der Tests und die Fertigstellung von Teilprojekten, um die Arbeitkraft der Mitarbeiter besser einzuteilen und eine schnellere Fertigstellung des Projektes zu bewerkstelligen. Die Visualisierung gibt einen Einblick, welche Teile des Quellcodes erfolgreiche Tests durchlaufen haben. Dadurch können Fehler schneller gefunden werden, was ohne Tarantula schwierig und zeitintensiv ist. [EHJS01]

Beim Verbessern des Verständnisses für Algorithmen oder Programmteile kann die Visualisierung den Nutzer unterstützen. Abbildung 3.9 zeigt eine Animation, welche die Effizienz von Sortieralgorithmen demonstriert. [Gos95]

3.6 Visualisierungstechniken

Für die Darstellung von Daten sind Diagramme in 2D- oder 3D- Darstellungen weit verbreitet. Diese sind zum Beispiel Balken-, Linien-, Punkte oder Kreis-

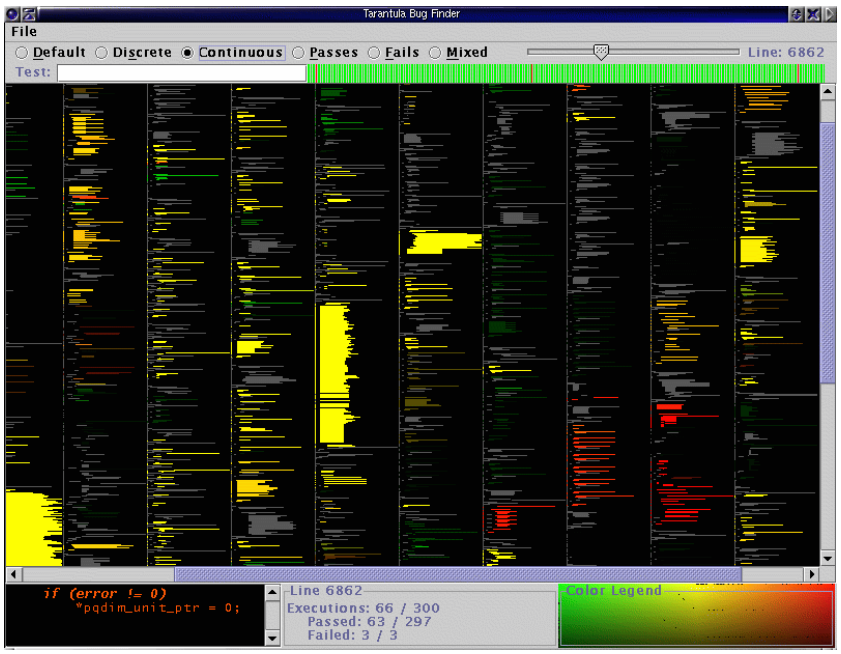

Abbildung 3.8: Tarantula-Softwarevisualisierung (sh. 3.5.6)

Abbildung 3.9: Die Visualisierung von Sortieralgorithmen (sh. 3.5.6)

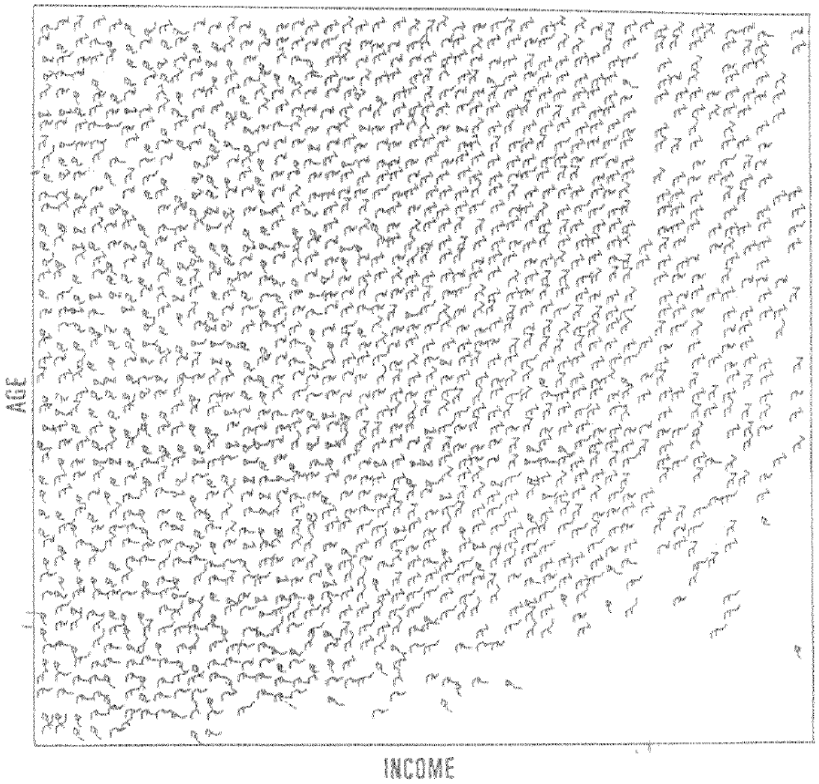

Abbildung 3.10: Strichmännchen Visualisierung (sh. 3.6.2)

diagramme. Neben diesen bekannten Visualisierungstechniken stehen auch neue, noch nicht so stark verbreitete Darstellungstechniken zur Verfügung. Auf diese soll im Folgenden genauer eingegangen werden.

3.6.1 Geometrische Transformation

Geometrische Transformationen werden von mehrdimensionalen Datensätzen gebildet. Unter Berücksichtigung der verschiedenen Attribute können die verschiedensten Sichten im Projektionsraum erstellt werden. Damit ist diese Visualisierungstechnik geeignet, um die explorative Analyse anzuwenden. Beispiele für diese Technik sind Scatterplot Matrizen [And72] [Cle93], Projection Pursuit-Techniken [Hub85] und die schon erwähnte Parallele Koordinaten Projektion [ID90].

3.6.2 Iconbasierte Visualisierung

Die Projektion der Attributwerte des Datensatzes auf ein Icon ist die Grundidee dieser Visualisierungstechnik. Je nach Wert eines Attributs verändern sich bestimmte Merkmale des Icons. So sind im Beispiel Abbildung 3.10 die Winkel

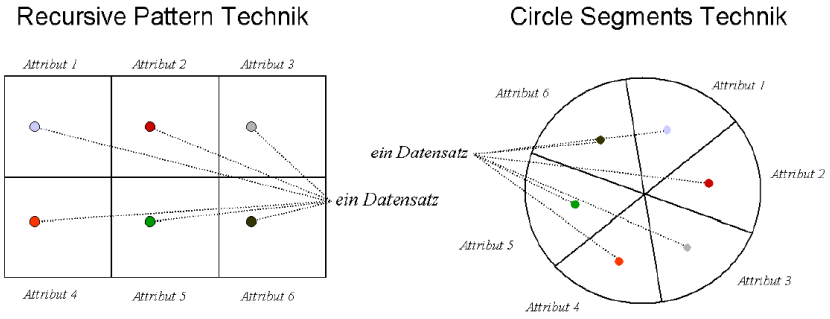

Abbildung 3.11: Pixel-Visualisierungen (sh. 3.6.3)

und Länge der Arme und Beine der Strichmännchen die Projektionsdimensionen von vier Attributen des Datenraumes. Korrelieren die Werte benachbarter Icons, so sind Ähnlichkeiten durch eine homogene Textur in der Darstellung erkennbar. Die genutzten Icons können je nach gewünschter Darstellung variieren. So können Daten über den Lebensstandard beispielsweise mit Gesichtern visualisiert werden. [Che73]

Die Strichmännchen Visualisierung [PG88] (Abbildung 3.10) zeigt Volkszählungsdaten der USA, wobei die Strichmännchen auf der X-Achse nach dem Einkommen und auf der Y-Achse nach dem Alter abgebildet sind. Die Attribute Abstammung, Schulbildung, Religion usw. sind auf Längen und Winkel der Arme und Beine der Strichmännchen projiziert. Homogene oder heterogene Strukturen sind für das menschliche Auge leicht zu erfassen; damit lassen sich Beziehungen zwischen den Attributen leicht erkennen. Weiterhin gibt es Darstellung mit Nadel-Icons [AK02], Stern-Icons [War94], Strichmännchen Icons [PG88], Farb-Icons [Lev91], [KK94] oder TileBars [Hea95].

3.6.3 Pixelvisualisierungen

Die Projektion eines Datenwertes auf ein farbiges Pixel ist der Grundgedanke der Pixelvisualisierungen. Abbildung 3.11 zeigt, dass jede Dimension des Datensatzes auf ein Pixel abgebildet wird und somit ein 6-dimensionaler Datensatz mit Hilfe von sechs Pixel dargestellt wird. Die Position der sechs Pixel innerhalb des Datensatzes bleibt gleich. Die Darstellung für sechs Dimensionen erfolgt in sechs Teildarstellungen. Die Datensätze sind auf alle sechs Teildarstellungen verstreut und stehen nur noch über ihre relative Position zueinander in Verbindung. Mit dieser Form der Visualisierung lassen sich Korrelationen und Abweichungen der Attribute zwischen und innerhalb der Teildarstellung gut finden.

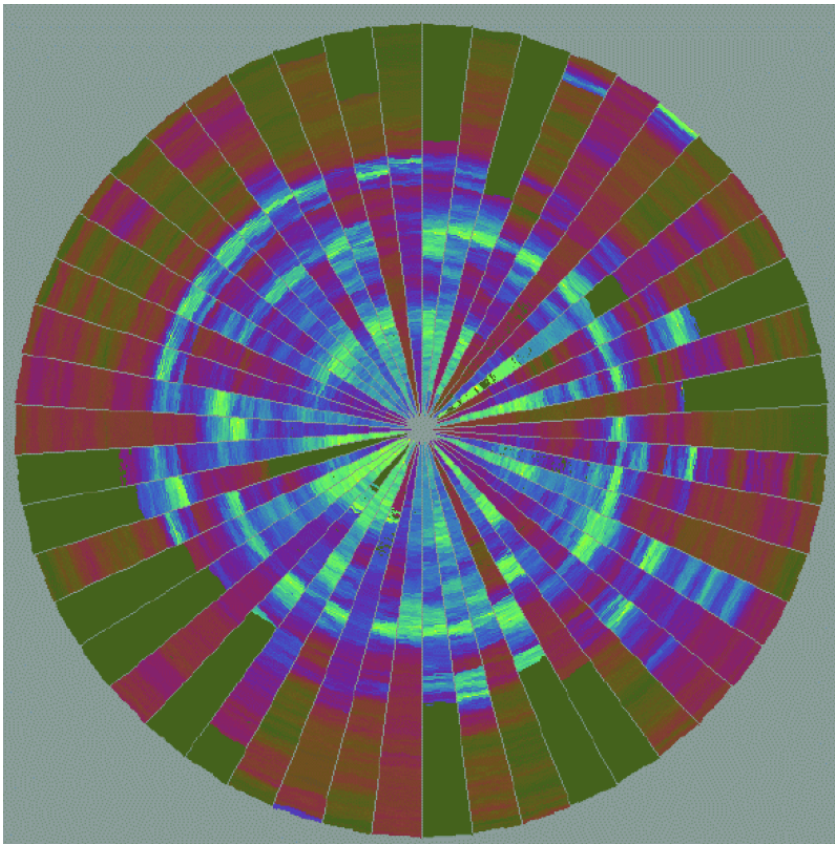

Abbildung 3.12: CircleSegments Technik (3.6.3)

Diese Form der Darstellung hat weiterhin eine hohe Informationsdichte, so können auf einem Bildschirm bist zu 1.000.000 Datenwerte dargestellt werden. In den folgenden Beispielen finden pixelorientierte Visualisierungen Anwendung: Spiral Technik [KK94], die Recursive Pattern Technik [KKA95] (Abbildung 3.2) und die Pixel Bar Chart Technik [KHDH02]. Die CircleSegments Technik [AKK96] (Abbildung 3.12) stellt wie die Recursive Pattern Technik die Attribute pixelorientiert dar. Die Zeitachse ist hier vom Kreismittelpunkt entlang des Radius zu betrachten. In dieser Darstellung werden 50 Werte des FAZ-Indexes dargestellt. (hell = steigende Kurse - dunkel = fallende Kurse)

3.6.4 Geschachtelte Visualisierungen

Die Aufteilung nach einem oder mehreren Attributen mit anschließender hierarchischer Darstellung nach diesen ist der Sinn, der sich hinter den geschachtelten Visualisierungen verbirgt. Dabei werden die Daten der unteren Hierarchie in die Darstellung der höheren Hierarchie eingeschachtelt. Die Auswahl der richtigen At-

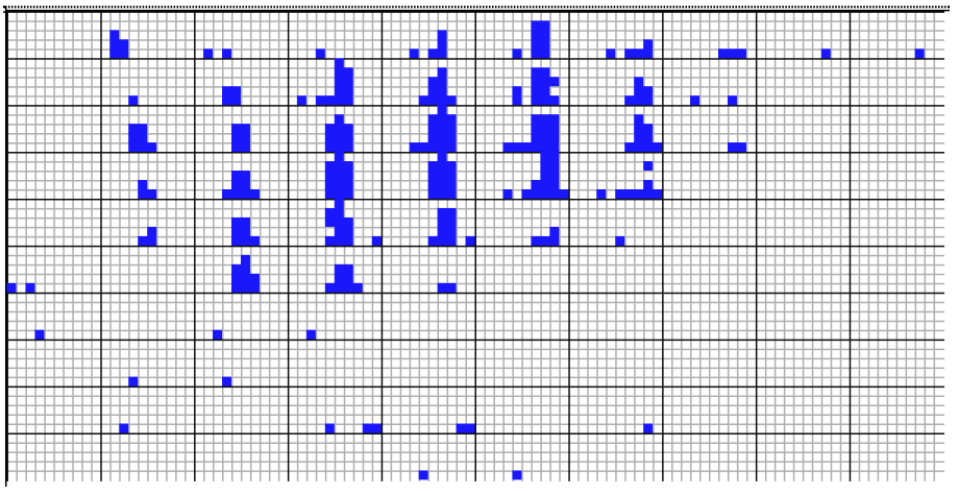

Abbildung 3.13: Dimensional Stacking Technik (sh. 3.6.4)

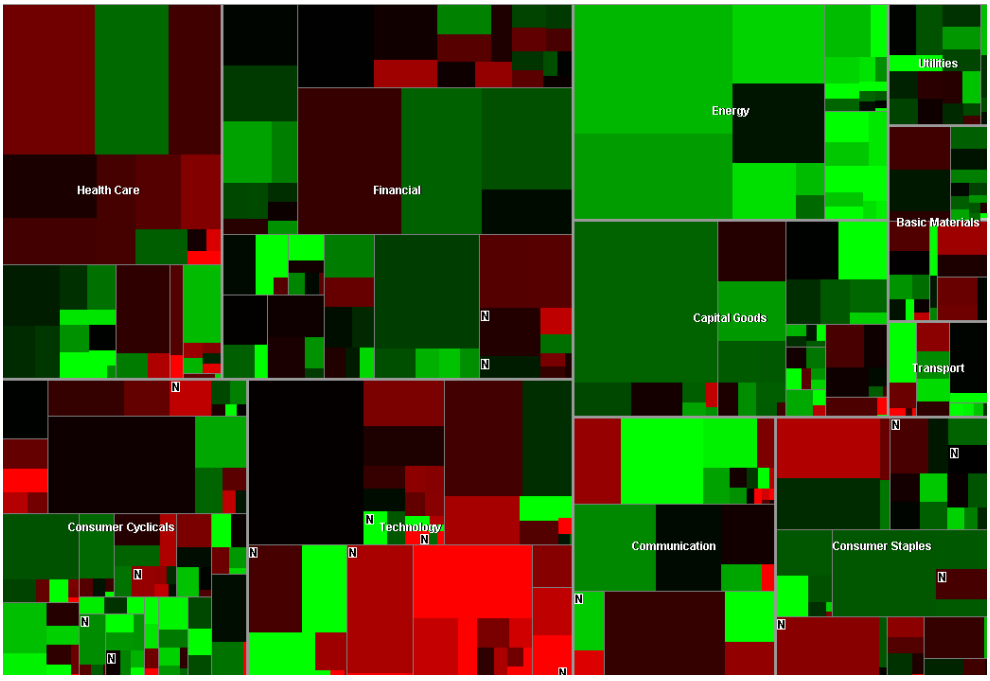

Abbildung 3.14: Treemap-Visualisierung (sh. 3.6.4)

tribute, die jeweils Achsen für die Hierarchien aufspannen, sind dabei ausschlaggebend für die Aussagekraft der Darstellung. Einige Beispiele sind: Dimensional Stacking [LWW90] (vgl. Abbildung 3.13), World-within-Worlds [FB90] und Treemaps [JS91] [Shn92] (Abbildung 3.14)

Die Idee der Dimensional Stacking Technik [LWW90] (vgl. Abbildung 3.13) ist die Einbettung eines Koordinatensystems in ein anderes Koordinatensystem. Die visuelle Darstellung wird durch die Aufteilung des äußersten Koordinatensystems in rechteckige Zellen erzeugt. Innerhalb dieser rechteckigen Zellen spannen zwei weitere Attribute ein inneres Koordinatensystem auf. Das innere Koordinatensystem wird dann wiederum in rechteckige Zellen zerlegt usw. In diesem Beispiel werden verschiedene Attribute von Ölförderdaten wie Längengrad, Breitengrad, Tiefe und Qualität visualisiert.

Die Treemap Visualisierungstechnik [JS91] [Shn92] (Abbildung 3.14) gliedert die Darstellung in X- und Y-Richtung nach der Hierarchiedefinition des Datenraumes. Die Größe und Farbe der entstehenden Gebiete entspricht weiteren Attributen. Die Abbildung zeigt Finanzdaten, wobei die Anordnung nach Marktsegmenten erfolgt und die Größe der Marktkapitalisierung entspricht [Sma01]. Die Farben stellen die Kursentwicklung dar. (rot entspricht fallenden Kursen und grün steigenden Kursen)

3.7 Interaktions- und Verzerrungstechniken

Zum Betrachten der Details der Datenvisualisierung sind Zooming und Interaktionstechniken für die explorative Datenanalyse unverzichtbar. Mit Hilfe der Interaktionstechniken kann zwischen verschiedenen Visualisierungstechniken gewechselt werden, um eine andere Sicht auf die Daten erhalten zu können und neue Zusammenhänge zu entdecken. Dabei kann es dazu kommen, dass der Betrachter die Übersicht über die Daten verliert; deshalb ist es nötig, Navigationselemente oder andere Hilfsmittel wie Verzerrungstechniken anzuwenden. Diese Verzerrungstechniken haben einen Bereich mit hoher Detailschärfe, die übrigen Bereiche eine geringere Detaildichte. Im weiteren Verlauf wird auf die Interaktions- und Verzerrungstechniken genauer eingegangen.

3.7.1 Dynamische Projektion

Multidimensionale Datenmengen können nur umständlich mit Hilfe von X-Y-Scatterplots dargestellt werden, bei denen jeweils eine Dimension gegen eine an-

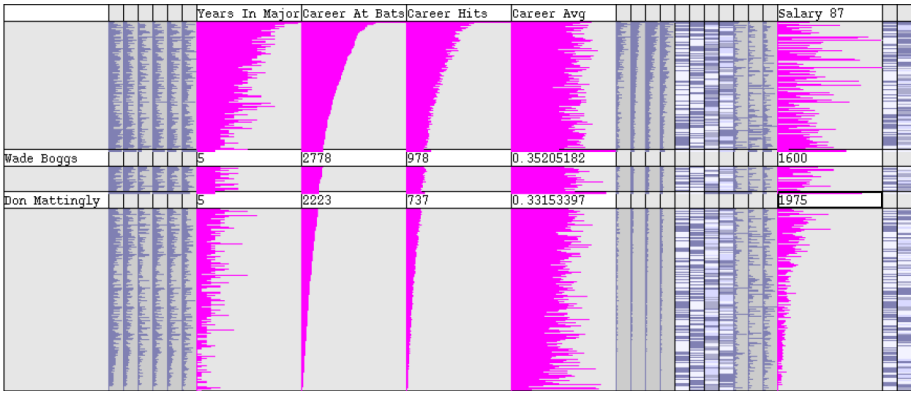

Abbildung 3.15: Table Lens Technik [RC94]

dere aufgetragen wird. Die Anzahl der möglichen Scatterplots ist $\frac{n^2}{2}$, wobei n die Anzahl der Dimensionen ist. Sie steigt exponentiell mit der Anzahl der Attribute des Datentyps. Im GrandTour System [Asi85] werden manuell, zufällig oder in der Abhängigkeit von Daten ein Teil der $\frac{n^2}{2}$ möglichen Sichten auf die Daten erstellt. Dieses System erstellt solange anhand der Messdaten verschiedene geometrische Transformationen, bis eine Sicht entsteht, auf der ein typisches Muster oder eine Clusterbildung zu erkennen ist. Die wissenschaftliche Relevanz dieser Darstellung muss von einem Betrachter bewertet werden. Weitere Beispiele für Anwendungen der dynamischen Projektionstechniken sind XGobi [SCB92][BSC96], XLispStat [Tie91] und ExplorN [CWL96].

3.7.2 Interaktive Filterung

Aufgrund der großen Datenmenge ist eine Aufteilung respektive Filterung eine wichtige Funktion für die explorative Analyse. Die Suche kann auf Erfolg versprechende Teile der Daten eingegrenzt werden (browsing). Weniger relevant erscheinende Daten werde somit ausgegrenzt. Die Auswahl der viel versprechenden Daten kann auch durch die Definition spezieller Eigenschaften festgelegt werden (querying). So ist es bei dem Tool Magic Lenses [BSP+93] möglich, in einem Bereich um den Mauszeiger eine andere Visualisierung zu nutzen. Andere interaktive Filter sind InfoCrystal [Spo93], Dynamic Queries [AS94] und Polaris [STH02].

3.7.3 Interaktives Zooming

Zooming ist die Fokussierung auf einen bestimmten Datenbereich. Die Vergrößerung desselben ist nicht nur eine Vergrößerung der Datenobjekte, sondern mit

höherer Zoomstufe werden auch mehr Details erkennbar. Datenobjekte, die in der Übersichtsdarstellung beispielsweise als mehrere Pixel dargestellt werden, repräsentieren sich in der nächsten Zoomstufe als Icon. In der maximalen Zoomstufe könnten sie dann ein beschriftetes Objekt darstellen.

Die Verwendung von TableLenses [RC94] ist ein schöne Umsetzung der Zoom-Idee auf große Tabellen. (Abbildung 3.15) Bei diesem Beispiel werden Zeilen und Spalten einer Tabelle als komprimierte Balken dargestellt. Mit Hilfe des Zooming können Details in Zeilen und Spalten betrachten werden.

Die folgenden Techniken und Systeme nutzen interaktives Zooming: PAD++ [PF93], IVEE/Spotfire [BH94] und DataSpace [ADLP95]. In [SDZ+93] werden Fisheye und Zooming weitergehend betrachtet.

3.7.4 Interaktive Verzerrung

Um alle Daten überblicken und auch Details wahrnehmen zu können, werden Verzerrungstechniken als Visualisierungsunterstützung angewendet. Im Gegensatz zum interaktiven Zooming werden hier alle Details dargestellt. Verzerrt wird beispielsweise die Länge einer Kante. Im Focus ist sie länger als am Rand der Darstellung. Die bekanntesten interaktiven Verzerrungstechniken sind hyperbolische und sphärische Verzerrungen. Hierarchien und Graphen werden vorzugsweise mit Hilfe interaktiver Verzerrung dargestellt. Einen weiteren Einblick in das Gebiet der interaktiven Verzerrungstechniken geben [LA94] und [CCF97]. Anwendungsbeispiele sind: Bifocal Displays [SA82], Perspective Wall [MRC91], Graphical Fisheye Views [Fur86], Hyperbolische Visualization [LRP95] und Hyperbox [AC91].

3.7.5 Interaktives Linking and Brushing

Beim interaktiven Linking und Brushing werden verschiedene Visualisierungstechniken miteinander verbunden. So werden Datensätze in einer Darstellungsform markiert (z.B. eingefärbt) und in derselben Markierung in einer anderen Sicht hervorgehoben.

Beispielsweise werden zwei verschiedene Scatterplots einer hochdimensionalen Datenmenge durch die Einfärbung und Verknüpfung von Punkten in allen Projektionen kombiniert. Nach diesem Prinzip ist es möglich, fast alle beschriebenen Visualisierungstechniken durch Linking and Brushing miteinander zu verbinden. Dadurch, dass nun bestimmte Datensätze in allen Visualisierungen hervorgehoben sind, können Korrelationen und Abhängigkeiten besser erkannt werden. Mit

Hilfe der Verknüpfung ist mehr zu erkennen als mit der Gesamtheit der Einzelvisualisierungen.

Einige Darstellungen, die mit Hilfe von interaktivem Linking and Brushing verbunden werden, sind Scatterplots, Balkendiagramme, Parallele Koordinaten, und Pixel-Visualisierungen. Viele interaktive visuelle Datenexplorations-Systeme unterstützen interaktives Linking and Brushing. Beispiele sind: Polaris [STH02], Scalable Framework [MKS02], XGobi [SCB92] und DataDesk [WUT95].

3.8 Auswertung der Attribute der Visualisierung für paketorientierte Netze

Die in Kapitel 2.4 und 2.6 aufgeführten Möglichkeiten der Visualisierung des Verkehrsaufkommens in paketorientierten Netzen werden im Folgenden auf ihre Aussagekräftigkeit untersucht. Weiterhin müssen Rückschlüsse darauf gezogen werden, welche Visualisierungsform für welche Attributdarstellung die effizienteste Darstellungsmöglichkeit ist.

1. IP-Adresse (Sender - Empfänger), je 32/128 bit
2. Version (Verteilung der Versionen je Zeiteinheit), 4 bit
3. Total Length (Bruttobyterate je Paket je Zeiteinheit), 16 bit
4. Header Length (Verwaltungsoverhead je Paket je Zeiteinheit), 4 bit
5. Time to Live (Anzahl der Hops bis Löschen des Pakets), 8 bit
6. Protokoll (IANA-Protokollnummer im Nutzdatenanteil je Paket), 8 bit
7. TOS/DSCP-Feld oder Priority (IPv4/IPv6 Header) - vier QoS Verkehrsklassen mit drei Paketverlustraten definiert, 8 bit
8. TCP-Port (Sender - Empfänger), je 16 bit
9. Offset (TCP-Overhead je Segment), 4 bit
10. ACK/SYN/FIN/RST/UGR/PSH (Verbindungsaufbau, -abbau, -abbruch je Segment je Zeiteinheit), je 1 bit
11. Window (Anzahl der Segmente, nach denen eine Bestätigung erfolgen muss), 16 bit

darzustellende Attribute	Wertebereich	Anzahl der Dimensionen	Pakets/s	Bytes/s	Flow/s	TCP-Ports	IP-Adresse
IP-Adresse SRC/DEST	2*32/128 bit	3	X-Y-Plot	X-Y-Plot	X-Y-Plot	Pixelvisualisierung	Pixelvisualisierung
Maskierung TCP-Header SRC/DEST	2-16 bit	mehr als 3	X-Y-Plot	X-Y-Plot	X-Y-Plot	Pixelvisualisierung	Pixelvisualisierung
TOS/Priority	6 bit	mehr als 3	X-Y-Plot	X-Y-Plot	X-Y-Plot	Pixelvisualisierung	Pixelvisualisierung
TCP-Ports SRC/DEST	2*16 bit	3	X-Y-Plot	X-Y-Plot	X-Y-Plot	Pixelvisualisierung	Pixelvisualisierung

Tabelle 3.1: Attribute und Wertebereich der darzustellenden Daten

12. Options MTU (Nutzdatenanteil in Bit des jeweiligen Segments), 16 bit

Die Darstellung dieser Attribute, die direkt aus dem Beobachtungsraum gewonnen wurden, schafft einen ersten Überblick über das Verkehrsaufkommen.

Weiterführende Verarbeitungsschritte der Attribute wurden im Kapitel 2.4 und 2.6 beschrieben.

Neben dem Bytestrom je IP-Adresse oder dem Port ist der Datenfluss (flow) je Applikation von Interesse. Ein Flow ist dadurch gekennzeichnet, dass Source- und Destination-IP-Adresse sowie Portnummern in mehreren Paketen gleich sind. Es ist zu beachten, dass TCP auch Verbindungen multiplexen kann. Dabei werden beim Verbindungsaufbau mehrere Ports für die Datenübertragung ausgehandelt und parallel übertragen; somit kann nicht jeder Fluss explizit zu einer Applikation zugeordnet werden. Daher ist eine Darstellung der Flows je Zeiteinheit eine weitere Darstellungsart. Die Darstellung der Datenströme kann somit in Paketen/Segmenten je Zeiteinheit, in Bit/Bytes je Zeiteinheit oder in Flows je Zeiteinheit erfolgen.

Die folgenden acht Vorschläge für Visualisierungen des Kommunikationsaufkommens in paketorientierten Netzen beziehen sich auf die auf Seite 40 vorgestellten Attribute.

Attributdarstellung 3, 4 und 9: Für die Darstellung der Paket- und Byterate ist eine Trennung nach eingehendem und ausgehendem Verkehr von Vorteil. In der Darstellung als X-Y-Plot kann dies folgendermaßen geschehen: Die Abszisse des X-Y-Plots stellt den Zeitverlauf dar. Auf der positiven Ordinate ist der ausgehende Datenstrom aufgetragen und auf der negativen Ordinate der eingehende Datenstrom. Diese Form der Darstellung lässt sich nicht für den Flow verwenden, da dieser ungerichtet ist.

Attributdarstellung 6: Um einen Überblick der Datenrate der Protokolle zu erhalten, ist die Darstellungsform, die in Abbildung 3.3 (Abschnitt 3.5.1) gezeigt wird, sinnvoll. Die Datenrate der einzelnen Protokolle wird mittels verschiede-

ner Farben und Höhen visualisiert. Dabei werden die Datenraten der Protokolle summiert gestapelt dargestellt.

Eine Kombination der Darstellung der Datenrate der Protokolle unter Berücksichtigung des ein- und ausgehenden Datenstroms (Trennung mittels positiver und negativer Ordinate) ist ebenfalls möglich.

Attributdarstellung 1 und 8: Zur Visualisierung der Nutzung der TCP-Ports ist die Pixelvisualisierung eine effiziente Darstellungsmethode. Zwar ist die Darstellung einer Matrix von 65.536*65.536 Datenwerten auf einem Bildschirm nicht möglich, werden aber 64 Ports in einem Pixel summiert, ist diese Darstellungsmatrix 1.024*1.024 Pixel groß. Die Spaltenwerte der Matrix repräsentieren die Quellports und die Reihenwerte die Senkenports. Die Farbe des resultierenden Pixels repräsentiert, wie oft diese Portkombination im untersuchten Zeitraum genutzt wird oder wie viele Bytes transportiert wurden. Dabei muss die Zuweisung der Farbwerte der Pixel logarithmisch zur Anzahl der transportierten Bytes oder Pakete je Port erfolgen. Eine ähnliche Darstellung ist für die Visualisierung der IP-Adressen und deren Datenaufkommen vorstellbar.

Attributdarstellung 2: Eine Darstellung des "Version"-Feldes nach Paket je Zeiteinheit kann lediglich einen Überblick darüber geben, welche Version des Internet-Protokolls genutzt wird. Diese Information ist für die Darstellung des Datenstroms nur von indirekter Bedeutung. Es ist daher nicht notwendig, dass diese Information visualisiert wird.

Attributdarstellung 5: Mit Auswertung des Felds "Time to Live" kann die Anzahl der Routerpassagen angeben werden, indem die Differenz zu 255 gebildet wird. Diese Angabe ist nicht in jeden Falle richtig, da das TTL-Feld auch mehrfach dekremetiert werden kann, wenn sich z.B. ein Paket in der Warteschlange eines Routers befindet. Mit der Darstellung dieses Attributs nach Zeiteinheiten, Ports oder IP-Adressen können keine Informationen über das Ausmaß des Datenstroms gewonnen werden, sondern nur zur Wegstrecke, welche die Daten zurückgelegt haben.

Attributdarstellung 7: Eine segment-, byte- und floworientierte Darstellung des Datenstroms mittels des TOS/DSCP-Feldes kann mit der modifizierten ThemeRiver-Visualisierung geschehen. Die einzelnen Verkehrsklassen werden in verschiedene Farben, nach ihrem Anteil summiert, gestapelt je Zeiteinheit dargestellt. Es ist weiterhin möglich, den Datenstrom nach Richtungen getrennt darzustellen (siehe Attributdarstellung 6). Momentan wird dieses Feld von den meisten Routern ignoriert, so dass eine Darstellung nicht angebracht ist.

Die **Attributdarstellung 9 und 12** lassen sich indirekt zur Visualisierung verwenden. Mit Hilfe des Offset und des MTU-Wertes aus dem Options-Feld kann die Nutzlast des Segments berechnet werden. Somit wird indirekt der Nettodatenstrom je Zeiteinheit angegeben.

Für die **Attributdarstellung 10** ist eine segment-, byte- und flowbasierte Darstellung nicht sinnvoll. Diese Flags können bei der Detektion von Flüssen behilflich sein, da diese bei Auf- und Abbau sowie Verbindungsabbrüchen gesetzt werden.

Mit Hilfe der **Attributdarstellung 11** können im gewissen Rahmen Aussagen zur Qualität der Verbindung gemacht werden. TCP vergrößert die Window-Size Schritt für Schritt, wenn keine Neuübertragung von Segmenten angefordert wurde. Die effektivste Darstellungsmethode ist in diesem Falle ein 2D-Diagramm, bei dem auf der Abszisse der Beobachtungszeitraum und auf der Ordinate die Window-Size abgebildet ist.

Wird das Datenaufkommen mit Hilfe einer Maskierung von Feldern des TCP-Headers untersucht, ist die Parallele Koordinaten Technik (sh. 3.5.3 und Abbildung 3.5) ein geeignetes Mittel der Visualisierung. Die Auswahl der Parameter für die Maskierung können Hypothesen sein, die u. a. mit Hilfe der zuvor beschriebenen Visualisierungsverfahren aufgestellt wurden.

Kapitel 4

MESSUNG DES KOMMUNIKATIONSAUFKOMMENS

4.1 Messmethoden in Netzwerken

Die Messung des Datenverkehrs in paketorientierten Netzen kann mit Hilfe einer Netzwerkkarte erfolgen. Die Karte liest den Netzwerkverkehr mit und kopiert ihn paketweise auf einen Datenträger. Dabei wird zwischen zwei Verfahren unterschieden: Es ist möglich, den gesamten Datenverkehr zu speichern oder nur Pakete zu speichern, die vorher festgelegten Filterregeln entsprechen (Offlineanalyse/ Onlineanalyse).
Es kann eine weitere Klassifizierung der Messmethoden unternommen werden.

Die erste Methode ist die aktive Messung des Netzwerkverkehrs. Die Karte sendet in diesem Falle Messpakete aus und ermittelt aus den Antworten und den Antwortzeiten Jitter-, Entfernungs- und Lastparameter. Ist das Datennetz bereits ohne Messung voll ausgelastet, kann eine zusätzliche Einbringung von Paketen zu Überlastsituationen führen. Ferner verfälscht diese Dateneinbringung die Charakteristik des Netzwerkverkehrs.

Die zweite Messmethode ist passiv, hier werden keine zusätzlichen Messpakete in den Datenstrom eingebracht. Ist die Netzlast zu gering, können mit dieser Messmethode nicht genügend Daten gesammelt werden.

Die dritte Messmethode (hybrid) verbindet die Vorteile der aktiven und passiven Methode. Im Falle, dass beim passiven Messen nicht genügend Messwerte ermittelt werden, findet die Generierung von Messpaketen statt. [KMV03]

Die einfache Messung des Datenverkehrs mittels einer Netzwerkkarte hat den Nachteil, dass die Vergabe der Timestamps für jedes Paket ungenau ist. Das heisst, dass der Zeitstempel bei dieser Messmethode erst vergeben wird, wenn das Paket vollständig angekommen ist und in den Arbeitsspeicher geschrieben wurde. Dieser Erfassungsprozess ist nicht deterministisch. Es ist vorstellbar, dass bei zu hoher Systemlast Pakete nicht erfasst werden. Diese Art der Messdatenerfassung kann für grobe Schätzungen des Datenverkehrs und die Visualisierung von Übertragungsraten oder Mittelwerten ausreichend sein, jedoch nicht für Modellbildung von Zwischenankunftsprozessen und Datenquellen.

4.2 Beschreibung des Messsystems

Um die beschriebenen in Abschnitt 4.1 aufgezeigten Nachteile der nicht deterministischen Vergabe der Zeitstempel und des möglichen Paketverlustes bei CPU-Überlast zu umgehen, wurden die Messungen mit einer speziellen Messkarte vorgenommen. Die Messkarte DAG 3.5E der Firma Endace führt alle notwendigen Verarbeitungsschritte mit Hilfe eines Xilinx FPGA on chip aus. Die ankommenden Frames werden mit einem 64 bit langen Zeitstempel versehen. Dabei beträgt die Auflösung $\frac{1}{2^{32}}$ Sekunden (<100 ns).

Die Messkarte wird als ein passives Netzelement in den zu beobachtenden Netzabschnitt eingefügt. Jedes Paket, das von der Messkarte erfasst wird, erhält mit Eintreffen des Paketheaders einen 64 bit genauen Zeitstempel und wird im Messsystem gespeichert.

Die Messung fand in einem 100 Mbit-Vollduplex-Netzwerk statt. Damit kann eine maximale Datenrate von 100.000.000*2 bit/s = 25 MByte/s auftreten. Der Messrechner, der für die Speicherung der Daten zuständig ist, muss eine ausreichende Systembus- und Speicherbandbreite besitzen, um diese maximale Last verarbeiten zu können.

Abbildung 4.1 zeigt den Aufbau des Messsystems, dessen Eckdaten im Folgenden aufgelistet sind:

- AMD Athlon 3,2 GHz - 64 bit Prozessor
- 2 x 512 MB DDR400 Arbeitspeicher
- VIA K8M800 Chipsatz
- Messkarte DAG 3.5E von Endace
- 2 x 250 GB Maxtor DiamondMAX10 6B250S0
- Betriebssystem SUSE-Linux 9.0 Kernel 2.4.21

Um den Nachweis zu führen, dass alle Pakete bei einer maximalen Linklast gespeichert werden können, wurde ein Benchmarktest der Schreibperformance des Systems mit dem Programm IOZONE [NOR04] vorgenommen. Abbildung 4.2 zeigt den Datendurchsatz je Sekunde beim Schreiben von 64 - 1.048.576 kB großen Dateien mit Datensatzgrößen von 4 - 16.384 kB. Der Datendurchsatz je Sekunde ist durch die Farben visualisiert. Auf der Abszisse sind die Datensatzgrößen und auf der Ordinate die Dateigröße aufgetragen. Auf Abbildung 4.2 ist ein Maxima

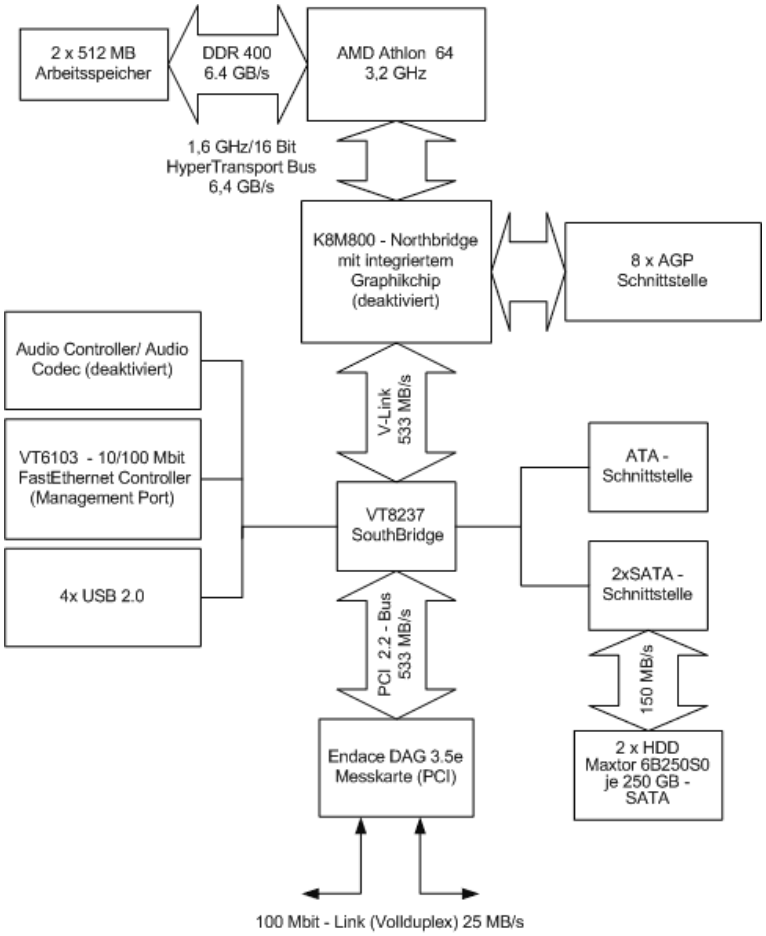

Abbildung 4.1: Blockschaltbild des Messsystems

bei einer Dateigröße von 262.144 kB und der Recordgröße von 128 kB erkennbar. Dieses Maximum ist damit begründet, dass Buffer für den Datentransport genutzt werden. Die eigentliche Performance der Festplatten und deren Controller zeigt Abbildung 4.3. Diese Werte sind auf jeden Fall größer als die geforderten 25 MB/s, dadurch kann ein Datenverlust bei der Speicherung der Daten ausgeschlossen werden.

Es war vorgesehen, dass das Messsystem aus zwei Festplatten (250 GB Maxtor DiamondMAX10 6B250S0) besteht. Aufgrund eines Defektes einer Festplatte musste das Messsystem ein zweites Mal aufgesetzt werden; dabei wurde nur noch eine Festplatte genutzt. Der Festplattenspeicher war ausreichend, um eine Messreihe über den Zeitraum von einer Woche aufzunehmen.

In [KMV03] wurden verschiedene Dateisysteme für ein ähnliches Messsystem analysiert. Aufgrund dessen wurde zur Verwaltung des Festplattenspeichers der LVM (logical volume manager) eingesetzt. Damit ist es möglich, auch mehre-

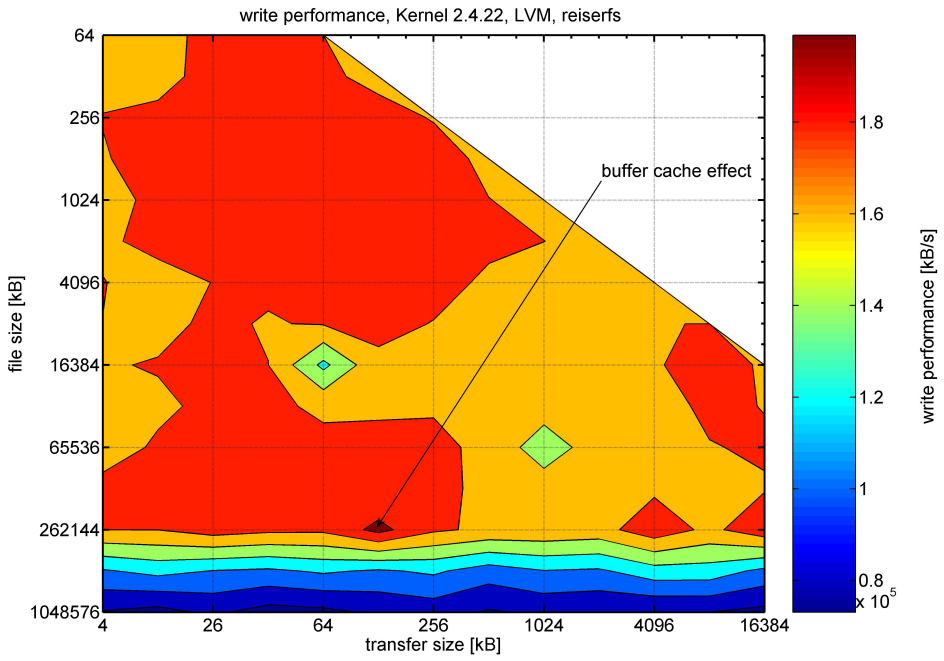

Abbildung 4.2: Schreibperformance der Festplatten nach Record- und Dateigröße

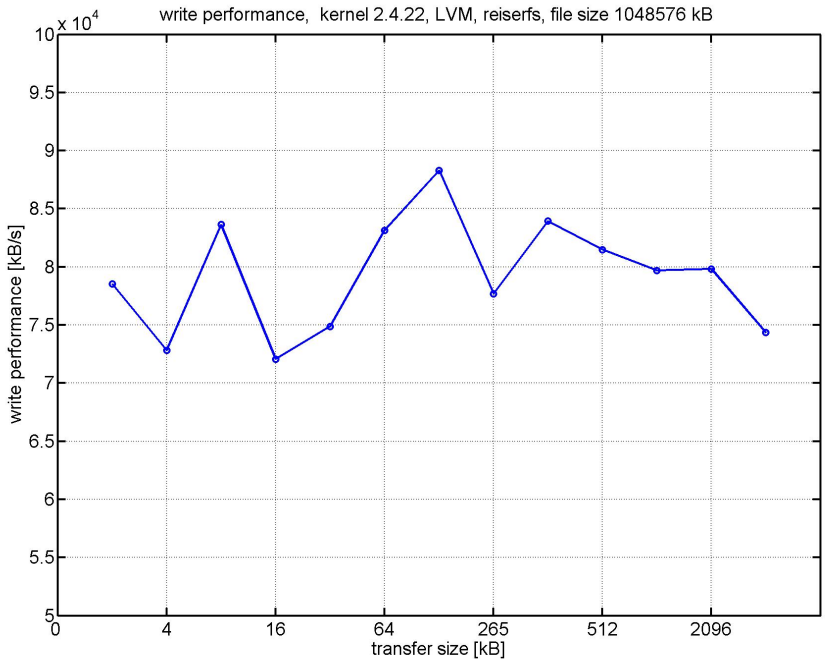

Abbildung 4.3: Schreibperformance der Festplatten ohne Buffereffekt

re physikalische Partitionen zu einer logischen Partition zusammenzufassen. Des Weiteren können im Nachhinein zusätzliche physikalische Partitionen zu dieser logischen Partition hinzugefügt werden.

4.3 Dauer und Umfang der Messung

Die Messung fand im Netzwerk des Instituts für Nachrichtentechnik und Informationselektronik der Universität Rostock statt. Der Zeitraum der Messung erstreckte sich von 25.10.2004 14:09 bis zum 02.11.2004 14:06. Da am 30.10.2004 die Uhrzeit von Mitteleuropäischer Sommerzeit auf Mitteleuropäische Winterzeit umgestellt wurde, beträgt die Länge der Messung somit 7 Tage 57 Minuten. Dies muss bei der Auswertung der Messdaten berücksichtigt werden.

Aus datenschutzrechtlichen Gründen wurden von jedem Ethernetframe nur die ersten 70 Byte aufgenommen. In diesen 70 Byte sind der IP- und TCP-Header (siehe Abbildung 2.2) mit allen für die vorgesehene Visualisierung notwendigen Parametern enthalten.

Kapitel 5

DATENFORMAT FÜR DIE SPEICHERUNG DER MESSDATEN

5.1 Datenformat der Messdaten

Die gemessenen Daten der Messkarte Endace DAG 3.5E werden auf der Festplatte im ER-Format (Extensible Record Format) abgelegt. Abbildung 5.1 zeigt den Aufbau eines Records. Jeder Frame wird in einem Record gespeichert. Für die gesamte Messdatei gibt es keinen Header. Es werden nur die Records hintereinander abgespeichert. Damit kann die Messdatei zwischen zwei Records getrennt werden, was ein einfaches Dateihandling ermöglicht. Der Aufbau eines Records (Typ 2 - Ethernet) wird im Folgenden beschrieben;

1. Zeitstempel (64 bit) erweitertes Unix Time Format (Zeit nach dem 1. Januar 1970)
2. type (8 bit) Art des Frameformats (Ethernet, HDLC, ATM ...)
3. flags (8 bit) mit folgender Belegung:
   - 0-1: Nummer der Messschnittstelle 0-3
   - 2: variierende Recordlänge
   - 3: nicht vollständiger Record (bei nicht ausreichendem Pufferspeicher)
   - 4: Empfangsfehler (link layer error)
   - 5: interner Fehler (internal error)

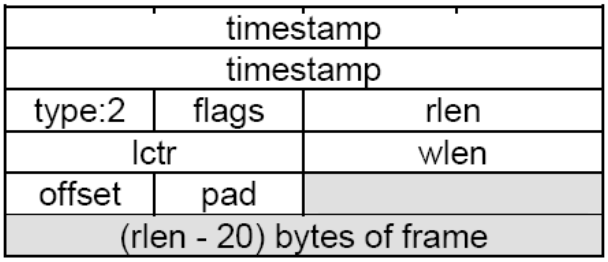

Abbildung 5.1: ERF-Format (Typ2)

- 6-7: reserviert

4. rlen (16 bit) Länge des Records, der via PCI-Bus übertragen wurde
5. lctr (16 bit) Zähler, der die Anzahl der verlorenen Frames zählt, die durch Überlast des PCI-Busses entstanden sind
6. wlen (16 bit) Länge des Frames (mit Protokolloverhead)
7. offset (8 bit) Anzahl der Bytes, die am Anfang des Frames übersprungen und somit nicht aufgezeichnet wurden. (Zur Zeit noch nicht implementiert.)
8. pad (8 bit) Füllfeld
9. (rlen - 20 bit) aufgenommene Frames (Messrohdaten)

[End04]

5.2 Anforderungen an ein generisches Datenformat

Die in den vorangegangenen Kapiteln vorgestellten Visualisierungsverfahren stellen unterschiedliche Anforderungen an das Messdatenformat. Da jede Visualisierung neue Ansichten der Daten darstellt, dürfen nicht nur einige wenige Parameter bei der Speicherung der Messdaten berücksichtigt werden. Weiterhin ist es nur sinnvoll, ein generisches Datenformat für die Speicherung der Messdaten zu entwickeln, wenn dieses Datenformat einfacher zu verarbeiten ist oder eine höhere Informationsdichte als die Rohdaten aufweist. Ein Teil der Informationen, die in den Paketheadern enthalten sind, spielen für die Visualisierung keine Rolle. Diese Informationen können im generischen Datenformat außer Acht gelassen werden. Der relevante Teil der Informationen, die in dem Datenformat berücksichtigt werden, wurde in Kapitel 2.4 und 2.6 vorgeschlagen. Bevor die Entwicklung eines Datenformats begonnen werden kann, muss eine Analyse bestehender oder in der Entwicklung befindlicher Datenformate vorgenommen werden.

5.3 IP flow information export - Standard

IPFIX (IP flow information export) ist der RFC 3917, der zur Zeit als Draft vorliegt. Ziel dieses zukünftigen Standards ist es, den Informationsaustausch in IP-basierten Netzwerken bestimmten Informationsflüssen zuordnen zu können. Zur Zeit existieren mehrere proprietäre Lösungen z.B. Firma Cisco- NetFlowV9 welche

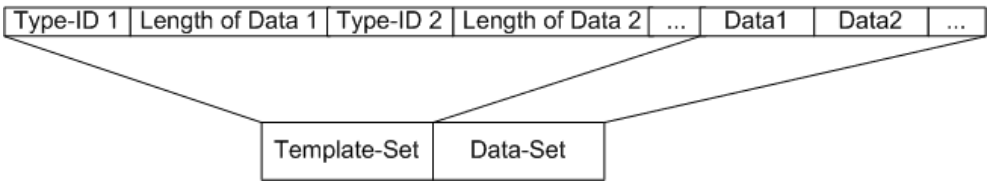

Abbildung 5.2: vereinfachter Aufbau des IPFIX-Datenformats

sich in wichtigen Punkten unterscheiden. Deshalb ist die exakte Definition eines Flusses noch schwierig. Ziel dieser ITEF-Group ist eine einheitliche Definition zu erarbeiten und ein Verfahren zur Erfassung, Speicherung und zum Transport von IP-Flüssen zu entwickeln. Dadurch wird es der Industrie und der Forschung auf standardisierte Weise ermöglicht, in Netzkomponenten wie Routern, IP-Flüsse zu extrahieren und diese Daten an weiter verarbeitende Instanzen, z.B. Abrechnungs- oder Netzwerkmanagmentsysteme weiterzuleiten.

Dieses IP flow information export system besteht aus einem Datenmodell, in dem die Flussinformationen enthalten sind, und einem Transportprotokoll. Die Ermittlung der IP-Flüsse geschieht in den aktiven Netzwerkomponenten (Router oder IP-Verkehrsmesspunkte). Die IP-Flüsse sind nach bestimmten Parametern gesammelte IP-Pakete. Diese Parameter sind beispielsweise IP-Ziel- und Quelladresse, Protokolltyp oder TCP-Ziel- und Quellport aber auch Informationen, die nur der sammelnden Netzkomponente bekannt sind. Dies sind z.B. die Schnittstelle, an der die IP-Pakete aufgelaufen/abgegangen sind oder die korrespondierenden Subnetzmasken.

Das Transportprotokoll wird benötigt, um die ermittelten Flüsse an Sammelpunkte weiterzuleiten, um dort analysiert zu werden. [IPFIX04]

Im Weiteren wird das IPFIX-Datenmodell zur Darstellung, Speicherung und zum Transport der ermittelten Flüsse vorgestellt.

Den vereinfachten Aufbau des IPFIX-Datenformats zeigt Abbildung 5.2. Wie zu erkennen ist, werden im ersten Teil des Datenformats die Reihenfolge und Länge der folgenden Datenfelder angegeben. Dies geschieht durch Angabe der Typ-ID (16 bit) und der Größe der Datenfelder (16 bit), welche im zweiten Teil des Datenformats folgen. Die Teilung von Referenzierung und Daten ermöglicht einen schnelleren Datenzugriff der Analysetools auf die Messdaten. Die möglichen Typ-ID's sind in Tabelle 5.1 dargestellt. Der vollständige Aufbau des Datenformats und die Größe der Datenfelder können der Spezifikation [IPFIX04] entnommen werden.

Es existieren im IPFIX-Datenfomat 12 Datentypen, welche die Wertebereiche

ID	Field Name	ID	Field Name
1	inDeltaCount	44	sourceIPv4Prefix
2	inPacketDeltaCount	45	destinationIPv4Prefix
3	totalFlowCount	46	mplsTopLabelType
4	protocolIdentifier	47	mplsTopLabelIPv4Address
5	classOfServiceIPv4	48-51	RESERVED
6	tcpControlBits	52	minimumTTL
7	sourceTransportPort	53	maximumTTL
8	sourceIPv4Address	54	identificationIPv4
9	sourceIPv4Mask	55	RESERVED
10	ingressInterface	56	sourceMacAddress
11	destinationTransportPort	57-59	RESERVED
12	destinationIPv4Address	60	ipVersion
13	destinationIPv4Mask	61	RESERVED
14	egressInterface	62	ipNextHopIPv6Address
15	ipNextHopIPv4Address	63	bgpNextHopIPv6Address
16	bgpSourceAsNumber	64	ipv6OptionHeaders
17	bgpDestinationAsNumber	65-69	RESERVED
18	bgpNextHopIPv4Address	70	mplsLabelStackEntry1
19	OutMulticastPacketCount	71	mplsLabelStackEntry2
20	OutMulticastOctetCount	72	mplsLabelStackEntry3
21	flowEndTime	73	mplsLabelStackEntry4
22	flowCreationTime	74	mplsLabelStackEntry5
23	outOctetDeltaCount	75	mplsLabelStackEntry6
24	outPacketDeltaCount	76	mplsLabelStackEntry7
25	minimumPacketLength	77	mplsLabelStackEntry8
26	maximumPacketLength	78	mplsLabelStackEntry9
27	sourceIPv6Address	79	mplsLabelStackEntry10
28	destinationIPv6Address	80-84	RESERVED
29	sourceIPv6Mask	85	octetTotalCount
30	destinationIPv6Mask	86	packetTotalCount
31	flowLabelIPv6	87-127	RESERVED
32	icmpTypeCode	128	bgpNextHopAsNumber
33	igmpType	129	ipNextHopAsNumber
34-35	RESERVED	130	exporterIPv4Address
36	flowActiveTimeOut	131	exporterIPv6Address
37	flowInactiveTimeOut	132	droppedOctetDeltaCount
38-39	RESERVED	133	droppedPacketDeltaCount
40	exportedOctetCount	134	droppedOctetTotalCount
41	exportedPacketCount	135	droppedPacketTotalCount
42	exportedFlowCount	136	flowEndReason
43	RESERVED	137	classOfServiceIPv6
		138	octetDeltaCount
		139	packetDeltaCount
		140	mplsTopLabelIPv6Address

Tabelle 5.1: Übersicht der möglichen ID-Felder des IPFIX Datenformat [IPFIX04]

der Daten vorgeben. Diese Datentypen werden in Folgenden beschrieben:

OCTET natürliche Zahl im Bereich von 0 bis 255

UNSIGNED16 natürliche Zahl im Bereich von 0 bis 65535

UNSIGNED32 natürliche Zahl im Bereich von 0 bis 4294967295

UNSIGNED64 natürliche Zahl im Bereich von 0 bis 18446744073709551615

FLOAT32 32-bit Zahl in IEEE-Fließkommadarstellung

BOOLEAN binärer Wert 0 oder 1

OCTETARRAY endliche Aneinanderreihung des Datentyps 'octet'

STRING endlich lange Reihe von gültigen Werten der Unicode-Notation

DATETIMESECONDS Zeitangabe mit der Auflösung einer Sekunde bezogen auf MEZ. Pendant zum UNIX-Zeitformat (32-bit Integer)

DATETIMEMICROSECONDS Zeitangabe mit der Auflösung einer Millisekunde bezogen auf MEZ.

IPV4ADDRESS Wert einer IPv4-Adresse (32-bit Integer)

IPV6ADDRESS Wert einer IPv6-Adresse (128-bit Integer)

Tabelle 5.1 schafft weiterhin eine Übersicht über alle Parameter, die u.a. aus dem TCP/IP-Header gewonnen werden. Diese Parameter können dazu genutzt werden, Pakete einem bestimmten Fluss zuzuordnen.

Es ist nicht notwendig, dass alle ID-Felder zur Beschreibung eines IP-Flusses genutzt werden müssen. Aufgrund der Teilung des IPFIX-Datenformats in einen Referenzierungs- und Datenteil, kann die Beschreibung eines IP-Flusses flexibel gestaltet werden.

Die Zuordnung eines Pakets zu einem IP-Fluss geschieht nach folgendem Ablauf:

Wird beispielsweise ein Fluss an Hand des Quellports identifiziert, ist das Feld "ID 7" das relevante Attribut für die Zuordnung zu einem IP-Fluss. Für die Zuordnung eines Pakets zu einem Fluss ist es beispielsweise irrelevant, wenn sich das Feld TTL im IPv4-Header ändert. Bei gleichem Feld 7 ist dieses Paket für diesen Fluss identifiziert. Damit kann Feld 3 (totalFlowCount) inkrementiert werden und

ID	Field Name	ID	Field Name
60	ipVersion	13	destinationIPv4Mask
8	sourceIPv4Address	30	destinationIPv6Mask
27	sourceIPv6Address	45	destinationIPv4Prefix
9	sourceIPv4Mask	5	classOfServiceIPv4
29	sourceIPv6Mask	137	classOfServiceIPv6
44	sourceIPv4Prefix	31	flowLabelIPv6
12	destinationIPv4Address	54	identificationIPv4
28	destinationIPv6Address	4	protocolIdentifier

Tabelle 5.2: ID-Felder des IPFIX-Datenformats des IP-Headers [IPFIX04]

ID	Field Name	ID	Field Name
7	sourceTransportPort	32	icmpTypeCode
11	destinationTransportPort	33	igmpType

Tabelle 5.3: ID-Felder des IPFIX-Datenformats des TCP-Headers [IPFIX04]

das analysierte Paket ist einem Fluss zugeordnet. Kann ein Paket keinem Fluss zugeordnet werden, wird auf Grundlage der Attribute des Pakets ein neuer Fluss erstellt.

Die Zuordnung der ID-Felder im IPFIX-Datenformat wurde so gewählt, dass der proprietäre NetFlow V9 Standard von Cisco berücksichtigt wurde. Feld 1 des Datenformats ist reserviert. Die Felder 1 - 127 besitzen dieselbe Zuordnung wie der NetFlow V9 Standard. Die Werte, die in den ID-Feldern 128 - 32767 berücksichtigt werden, stehen unter der Verwaltung der IANA. Die Felder 32768 - 65535 können nutzerspezifisch verwendet werden.

Tabelle 5.2 und 5.3 geben die Informationen an, die aus den jeweiligen Headern der korrespondierenden OSI-Schichten gewonnen werden.

Tabelle 5.4 zeigt die Informationsfelder, die auf OSI-Layer 2 gewonnen werden. MPLS (Multi Protocol Label Switchting) ist ein Verfahren, bei dem ein Quasi-Routing schon in Switches vorgenommen werden kann. Jeder Frame wird mit einer Markierung (Label) versehen zum Ziel geschickt. Anhand der Markierung kann eine schnellere Wegeleitung vonstatten gehen, da nicht für jeden Frame eine Abfrage der Routingtabelle geschehen muss.

Tabelle 5.5 zeigt die Felder, deren Informationen direkt aus Switches oder Routern gewonnen werden.

Das IPFIX Datenformat ist stark auf die Speicherung und den Transport von IP-Flüssen ausgerichtet. Eine Nutzung des IPFIX-Datenformats zu Speicherung von paketorientierten Messdaten ist möglich. Die flexible Gestaltung des Daten-

ID	Field Name	ID	Field Name
56	sourceMacAddress	75	mplsLabelStackEntry6
70	mplsLabelStackEntry1	76	mplsLabelStackEntry7
71	mplsLabelStackEntry2	77	mplsLabelStackEntry8
72	mplsLabelStackEntry3	78	mplsLabelStackEntry9
73	mplsLabelStackEntry4	79	mplsLabelStackEntry10
74	mplsLabelStackEntry5		

Tabelle 5.4: ID-Felder des IPFIX Datenformats des OSI-Layer 2 [IPFIX04]

ID	Field Name	ID	Field Name
15	ipNextHopIPv4Address	128	bgpNextHopAsNumber
62	ipNextHopIPv6Address	18	bgpNextHopIPv4Address
14	egressInterface	63	bgpNextHopIPv6Address
129	ipNextHopAsNumber	46	mplsTopLabelType
16	bgpSourceAsNumber	47	mplsTopLabelIPv4Address
17	bgpDestinationAsNumber	140	mplsTopLabelIPv6Address

Tabelle 5.5: ID-Felder des IPFIX Datenformats aus Switches/Routern [IPFIX04]

formats erlaubt es, ID-Felder ausser Acht zu lassen, die keine relevanten Attributwerte in den Messdaten besitzen.

5.4 Datenformate für paketorientierte Messwerte

In diesem Abschnitt werden zwei Datenformate für paketorientierte Messwerte vorgestellt, deren Grundlage das IPFIX-Datenformat bildet. Die entwickelten Datenformate sind so gestaltet, dass sie zum IPFIX-Datenformat kompatible sind. Es bestünde die Möglichkeit, mit dem zu entwickelnden Datenformat beide Versionen des Internet-Protokolls zu beschreiben. Dies ist aber aus Effizienzgründen inakzeptabel, da ein Paket nur als IPv4 oder IPv6 vorliegen kann. Es würden die für IPv4 oder IPv6 vorgesehenen Felder nicht genutzt werden.

Deshalb wurde für jede der zwei Versionen der Internet-Protokolls ein Datenformat entwickelt. Diese werden im Folgenden näher erläutert:

5.4.1 Datenformat für IPv4

Wie in Tabelle 5.6 zu ersehen ist, bildet das IPFIX-Datenformat die Grundlage des Datenformats zur Speicherung der Messwerte für das Internet-Protokoll der Version 4. Nicht genutzte Felder wurden nicht übernommen und es wurden sechs zusätzliche Felder eingeführt. Im Folgenden werden die Felder erklärt. Detaillierte

Nr.	Name des Feldes	IPFIX-ID	Größe des Feldes
01	protocolIdentifier	4	8 bit
02	classOfServiceIPv4	5	8 bit
03	tcpControlBits	6	8 bit
04	sourceTransportPort	7	16 bit
05	sourceIPv4Address	8	32 bit
06	destinationTransportPort	11	16 bit
07	destinationIPv4Address	12	32 bit
08	IPversion	60	8 bit
09	HeaderLengthIPv4	32769	8 bit
10	ttlIPv4	32770	8 bit
11	timestampERF	32771	64 bit
12	wlen /bytes on wire	32772	16 bit
13	TCPheaderLength/ TCPDataOffset	32773	8 bit
14	PacketLengthIPv4	32774	16 bit
15	FlagsERF	32775	8 bit

Tabelle 5.6: Datenformat für paketorientierte Messdaten des IPv4

Informationen zu den folgenden Feldern können den Kapiteln 2.3.1 (IPv4), 2.5.2 (TCP) und 5 (ER-Format) entnommen werden.

1. Protokollnummer des Segments im IPv4 Payload, IPv4 Protocol-Feld
2. classOfServiceIPv4, IPv4 TOS/DCSP-Feld
3. tcpControlBits, Flags des TCP-Headers
4. sourceTransportPort, TCP-Quellport
5. sourceIPv4Address, IPv4 Quelladresse
6. destinationTransportPort, TCP-Zielport
7. destinationIPv4Address, IPv4 Zieladresse
8. ipVersion, Versionsnummer des Internet-Protokolls
9. HeaderLengttIPv4, IPv4 HeaderLength-Feld
10. ttlIPv4, IPv4 Time to Live-Feld
11. timestampERF, Im Gegensatz zum IPFIX-Timestampformat besitzen die von der Messkarte ermittelten Timestamps eine Genauigkeit von 64 bit.

Diese Genauigkeit ist für die Analyse und Auswertung der Messdaten notwendig. Daher wird das Timestampfeld des IPFIX-Datenformats nicht berücksichtigt. Statt dessen wird ein nutzerspezifisches Feld erstellt.

12. wlen/bytes on wire - gibt die Länge des Frames an, der von der Messkarte erfasst wurde. Im Datenformat der Messkarte ist dies das wlen-Feld.

13. TCPheaderLength/ TCPDataOffset - repräsentiert die Größe des TCP-Headers, damit ist mit Hilfe eines weiteren Verarbeitungschrittes die Größe des Layer-4-Payloads anzugeben.

14. PacketLengthIPv4, IPv4 Total Length-Feld

15. FlagsERF - Anhand der Flags des Datenformats der Messkarte kann u.a. die Nummer der Messschnittstelle ausgelesen werden. Damit ist es möglich, die Messdaten nach ein- und ausgehendem Datenverkehr zu unterscheiden.

Der Speicheraufwand für die Daten eines Pakets beträgt 32 Byte, die Größe des ID-Typ-Feldes 30 Byte und das Feld der Längenangabe der Daten 30 Byte. Somit ergibt sich ein Speicherplatzaufwand von 92 Bytes je Paket. Damit beträgt der Verwaltungsoverhead des Datenformats 60 Bytes. Dies ist aus Effizienzgründen inakzeptabel. Der Verwaltungsoverhead kann entweder mit Verkleinerung des ID-Typ-Feldes (16 bit) oder des Length-Feldes (16 bit) minimiert werden. Da das Datenformat zu IPFIX kompatibel sein soll, ist eine Verkleinerung des ID-Typ-Feldes ausgeschlossen. Das Length-Feld stellt bei der Breite von 16 bit einen Datentyp mit einer Länge von 65.535 Byte dar. Die im IPFIX-Datenformat spezifizierten Datentypen haben hingegen eine maximale Länge von 128 bit. Eine Verkleinerung des Length-Feldes auf 8 bit kann eine Länge des Datentyps von 256 Byte anzeigen. Dies reicht auch bei Erweiterung des IPFIX-Datenfomates aus, um die Länge der Datentypen anzeigen zu können. Mit Verkleinerung des Length-Feldes ergibt sich für den Speicherplatzaufwand eines Pakets ein Wert von 77 Bytes und somit ein Verwaltungsoverhead von 45 Bytes.

5.4.2 Datenformat für IPv6

Das Messdatenspeicherformat für das IPv6 ist dem Format für IPv4 ähnlich. Bei der Entwicklung des Formats wurden sechs nutzerspezifische ID-Felder dem IPFIX-Datenformat hinzugefügt. Die berücksichtigten Felder sind in der Tabelle 5.7 dargestellt und werden im Folgenden kurz erläutert. Nähere Informationen sind den Kapiteln 2.3.2 (IPv6), 2.5.2 (TCP) und 5 (ER-Format) zu entnehmen.

ID	Name des Feldes	IPFIX-ID	Größe des Feldes
01	protocolIdentifier	4	8 bit
02	tcpControlBits	6	8 bit
03	sourceTransportPort	7	16 bit
04	destinationTransportPort	11	16 bit
05	sourceIPv6Address	27	128 bit
06	destinationIPv6Address	28	128 bit
07	flowlabelIPv6	31	32 bit
08	ipVersion	60	8 bit
09	ipv6OptionHeader	64	32 bit
10	classOfServiceIPv6	137	8 bit
11	hoplimIPv6	32770	8 bit
12	timestampERF	32771	64 bit
13	wlen /bytes on wire	32772	16 bit
14	TCPheaderLength/ TCPDataOffset	32773	8 bit
15	FlagsERF	32775	8 bit
16	PayloadLengthIPv6	32776	16 bit

Tabelle 5.7: Datenformat für paketorientierte Messdaten des IPv6

1. Protokollnummer des Segments im IPv6 Payload,IPv6 Protocol-Feld
2. tcpControlBits, Flags des TCP-Headers
3. sourceTransportPort, TCP-Quellport
4. destinationTransportPort, TCP-Zielport
5. sourceIPv6Address, IPv6 Quelladresse
6. destinationIPv6Address, IPv6 Zieladresse
7. flowlabelIPv6, Markierung der zu einem IP-Fluss gehörigen Pakete
8. ipVersion, Versionsnummer des Internet-Protokolls
9. ipv6OptionHeader, Anzahl der Optionsheader, die gesetzt wurden. Dieses Feld ist in der Spezifikation von IPFIX z.Z. nicht vollständig bestimmt.
10. classOfServiceIPv6, IPv6 Class of Service-Feld
11. hoplimIPv6, IPv6 HopLimit-Feld
12. timestampERF - Im Gegensatz zum IPFIX-Timestampformat besitzen die von der Messkarte ermittelten Timestamps eine Genauigkeit von 64 bit.

Diese Genauigkeit ist für die Analyse und Auswertung der Messdaten notwendig. Daher wird das Timestampfeld des IPFIX-Datenformats nicht berücksichtigt und statt dessen ein nutzerspezifisches Feld erstellt.

13. wlen/bytes on wire - gibt die Länge des Frames an, der von der Messkarte erfasst wurde. Im Datenformat der Messkarte ist dies das wlen-Feld.

14. TCPheaderLength/TCPDataOffset - repräsentiert die Größe des TCP-Headers, damit ist mit Hilfe eines weiteren Verarbeitungschrittes die Größe des Layer-4-Payloads anzugeben.

15. FlagsERF - Anhand der Flags des Datenformats der Messkarte kann u.a. die Nummer der Messschnittstelle ausgelesen werden. Damit ist es möglich, die Messdaten nach ein- und ausgehendem Datenverkehr zu unterscheiden.

16. PayloadLengthIPv6 - Die Angabe in diesem Feld bezieht sich auf die Länge des IPv6-Pakets ohne Basisheader. Es sind alle optionalen Header mit einbezogen. Ein korrekter Wert für den Layer-3-Payload ergibt sich, wenn die Wert aller HdrExtLen-Felder (Header Extended Length) in den optionalen IPv6-Headern abgezogen werden.

Damit ergibt sich mit der in Kapitel 5.4.1 beschriebenen Kürzung des Referenzteils des Datenformats ein Speicherplatzaufwand von 111 Bytes je Paket, wovon 64 Byte auf den Referenzteil des Datenformats fallen.

Kapitel 6

IMPLEMENTIERUNG EINER VISUALISIERUNG FÜR PAKETORIENTIERTEN DATENVERKEHR

In diesem Kapitel werden die Verarbeitungsschritte und Algorithmen, die bei der Implementierung genutzt wurden, vorgestellt. Für OSI-Layer 3 und 4 wurden unterschiedliche Visualisierungen entwickelt. Die Visualisierung wurde nach Recherche und Vergleich von Softwarepaketen (SAS V8.02, SPSS, MATLAB 7.0 SP1, Microsoft EXCEL 2003), welche die Datenverarbeitung großer Datenmengen effizient ermöglichen, mit Hilfe von Matlab 7.0 SP1 entwickelt.

SAS V8.02 kann schnell große Datenmengen verarbeiten, bietet jedoch bei der Visualisierung nicht genügend Unterstützung. SPSS - eine Softwarelösung, die für die statistische Sozialforschung entwickelt wurde, besitzt ein gutes Statistikpaket. Die Erstellung von Grafiken ist mit dieser Software einfach. Die angebotenen Visualisierungen sind stark auf statistische Aufgaben beschränkt und somit für die Visualisierung des Kommunikationsaufkommens paketorientierter Netzwerke nicht ausreichend. Die Tabellenkalkulation Excel von Microsoft ist ebenfalls für die Aufgabenstellung nicht ausreichend, da hier die maximale Anzahl der Datensätze beschränkt ist.

Um die Messdaten, die mit der Endace-Messkarte gewonnen wurden, mit Matlab verarbeiten zu können, muss eine Definition der Importschnittstelle des Visualisierungstools geschehen. Diese umfasst die Beschreibung des Datenformats zum Import der Messdaten. Da in der vorliegenden Version der Implementierung verschiedene Visualisierungen für OSI-Layer 3 und 4 entwickelt wurden, erfolgte die Nutzung zweier Importschnittstellen. Im Folgenden werden die Schnittstellen vorstellt.

6.1 Schnittstellen

6.1.1 Importschnittstellen

Die gemessenen Pakete werden zu einem Datensatz, der einem Zeitraum von 256 Sekunden entspricht, zusammengefasst, um eine Messdauer von einer Woche mit einer angemessenen Anzahl von 2.300 Datensätzen darstellen zu können. Bei ei-

ner differierenden Messdauer kann der Zeitraum der Zusammenfassung der Pakete dementsprechend angepasst werden, so dass ca. 2.000 - 2.500 Datensätze anfallen. Eine höhere Anzahl von Datensätzen verlangsamt die Visualisierung durch den höheren Rechenaufwand. Die dadurch resultierende höhere Auflösung der einzelnen Werte bei der Visualisierung kann bei einer Übersichtsdarstellung der gesamten Messdaten nicht präsentiert werden, da die Darstellungsgrenze der heutigen Bildschirme ca. 1.000-1.600 Pixel beträgt.

Die Importschnittstelle der Layer-3-Visualisierung wurde so ausgelegt, dass ein Datenformat mit der folgenden Beschreibung eingelesen werden kann: Daten je Datensatz:

- Timestamp 64 bit Wert (extended UNIX-Timeformat)
- Angefallene Daten je Datensatz je Protokoll in:
  - Anzahl der Frames (number of frames) (nof)
  - Größe des Pakets (bytes in payload) (bip)
  - Größe des Payloads (bytes on wire) (bow)

Die Anordnung der zeichenseparierten Werte in der zu importierenden Messdatendatei ist folgende:

```
Timestamp0,nof0,bow0,bip0,nof1,bow1,bip1,...,nof255,bow255,bip255
Timestamp1,nof0,bow0,bip0,nof1,bow1,bip1,...,nof255,bow255,bip255
.
.
.
TimestampN,nof0,bow0,bip0,nof1,bow1,bip1,...,nof255,bow255,bip255
```

Für die Importschnittstelle der Layer-4-Messdaten ist der folgende Aufbau vorgesehen: Wie in Layer 3 wird eine Zusammenfassung auf 256 Sekunden bei einer Messdauer von einer Woche vorgesehen. Daten je Datensatz:

- Timestamp 64 bit Wert (extended UNIX-Timeformat)
- Angefallene Daten je Datensatz je Protokoll in:
  - incoming TCP-Port (in)
  - outgoing TCP-Port (out)
  - übertragene Bytes (bytes)

Die Anordnung der zeichenseparierten Werte in der zu importierenden Messdatendatei ist folgende:

```
if Timestamp
  Spalte 1 = Timestamp;
  Spalte 2 = 0;
  Spalte 3 = 0;

if Daten
   Spalte 1 = in-Port;
   Spalte 2 = out-Port;
   Spalte 3 = Bytes;
```

als Beispiel:

```
Timestamp1,0,0
in0,out0,bytes00
in0,out1,bytes01
.
.
in0,out65635,bytes065535
in1,out0,bytes10
.
.
in65535,out65535,bytes6553565535
Timestamp2,0,0
in0,out0,bytes00
in0,out1,bytes01
.
.
in0,out65635,bytes065535
in1,out0,bytes10
.
.
in65535,out65535,bytes6553565535
```

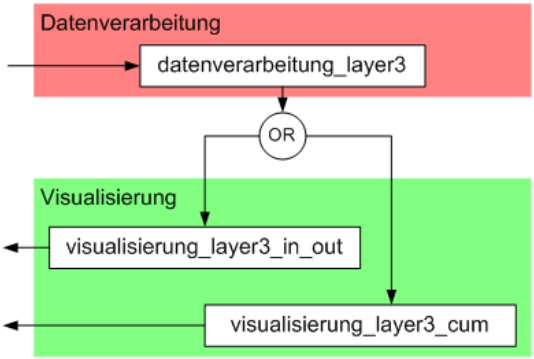

Abbildung 6.1: Übersicht und Einordnung der Skripte der Layer-3-Visualisierung

6.1.2 Exportschnittstellen

Die Ausgabe der Visualisierung geschieht auf dem Bildschirm, als JPG-Datei oder als PDF-Dokument. Bei der Bildschirmwiedergabe besteht die Möglichkeit, mittels Interaktionstechniken bestimmte Bereiche der Visualisierung näher zu untersuchen. Der Export als JPG- oder PDF-Datei dient der Dokumentation und Weiterverwendung der Visualisierungen.

6.2 Beschreibung der Layer-3-Visualisierung

Abbildung 6.1 zeigt einen Überblick der genutzten Skripte der Layer-3-Visualisierung. Es wurde eine Unterteilung in Datenverarbeitung und Visualisierung vorgenommen. In den anschliessenden Abschnittten 6.2.1 und 6.2.2 wird der Ablauf der Visualisierung anhand der gewählten Unterteilung erklärt.

6.2.1 Ablauf der Datenverarbeitung

Die Datenverarbeitung, Filtern von Information und die Skalierung der Messwerte werden mit Hilfe des Skriptes "**datenverarbeitung_layer3**" gestartet. Dort findet nach dem Einlesen der Messdaten eine Plausibilitätsprüfung der Zeitstempel der Pakete statt. Es wird geprüft, ob die Zeitstempel der zwei Messdatendateien (von jedem Messinterface eine) gleich sind.

Anschließend findet eine Konvertierung des Zeitstempelformats von Unix-Zeitstempelformat in das Matlab-Zeitformat statt.

Im nächsten Schritt werden die für die weitere Berechnung nicht benötigten Daten gefiltert. Das Ergebnis ist eine Sicht, in der nur noch die Frames /Bytes in payload/ Bytes on wire - Angaben (siehe 6.1.1) vorhanden sind.

Da alle Werte dieser Sicht auf 256 Sekunden bezogen sind, wird in diesem Schritt eine Skalierung der Werte auf eine Sekunde vorgenommen. Somit sind die dargestellten Werte Mittelwerte, die über einen Zeitraum von 256 Sekunden gebildet wurden. Diese Skalierung ist in zwei Zweige aufgeteilt. Der erste Zweig wird genutzt, wenn die Zeitstempel über den Beobachtungszeitraum äquidistante Abstände besitzen. Der zweite Zweig wird genutzt, wenn dies nicht der Fall ist. Die Skalierung im ersten Zweig ist durch die Vektorisierung des Codes schneller als die des zweiten Zweiges.

Der nächste Abschnitt des Visualisierungsskriptes kann genutzt werden, um eine Glättung der darzustellenden Daten vorzunehmen. Der Parameter "filter_span" bestimmt die Stärke der Glättung der Kurve.

6.2.2 Ablauf der Visualisierung

Für die Visualisierung existieren zwei Skripte. Im ersten Skript "**visualisierung_layer3_in_out**" wurde der ein- und ausgehende Datenverkehr jeweils in der positiven und negativen Ordinatenrichtung nach der Zeit abgebildet.

Die zweite Visualisierung, die mit Hilfe des Skriptes "**visualisierung_layer3_cum**" startet, visualisiert den summierten ein- und ausgehenden Datenverkehr auf OSI-Layer 3.

Der Ablauf der Visualisierung ist in beiden Skripten ähnlich. Anhand des Parameters "result_plot" wird bestimmt, welcher Wert (frames/second, bytes/second oder payloadbytes/second) auf der Ordinate abgebildet wird.

Der Parameter "min_val" bestimmt, welche Protokolle bei der Visualisierung abgebildet werden. Ist der Parameter beispielsweise "0.01", werden alle Protokolle in der Visualisierung berücksichtigt, deren Wert im Beobachtungszeitraum jemals 1% des lokalen Maximums überschritten haben.

Der nächste Abschnitt des Skriptes wählt anhand des Parameters "result_plot" und des Ergebnisses des vorhergehenden Skriptabschnittes die Spalten der Messdatensicht aus, die dargestellt werden.

Im Abschnitt "Labeling of x axis" des Skriptes wird die Skalierung der Abszisse berechnet. Das serielle Zeitformat in Matlab ist so aufgebaut, dass das Datum durch eine Zahl repräsentiert wird, die seit dem 1.1.0000 täglich um eins erhöht wird. Die Uhrzeit wird durch die Nachkommastellen dieser Zahl repräsentiert. Eine Markierung der Abszisse wird gesetzt, wenn der Teil der Nachkommastellen der Timestamp minimal ist. Dies ist jeden Tag im Beobachtungszeitraum kurz nach Mitternacht der Fall. Weitere Markierungen werden gegen 12 Uhr mittags

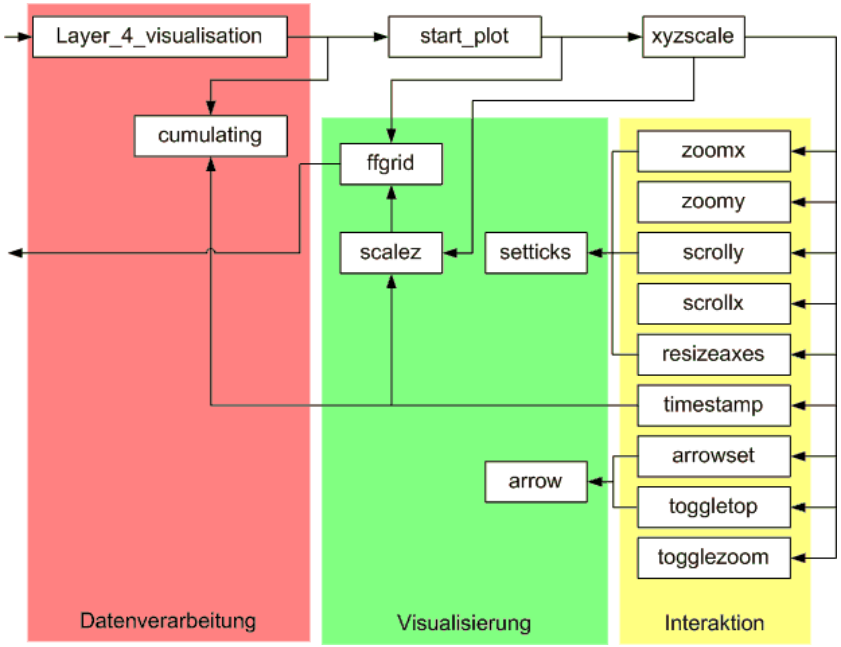

Abbildung 6.2: Übersicht und Einordnung der Skripte der Layer-4-Visualisierung

eingefügt. Diese sind die korrespondierenden Timestamps, beim halben Abstand zwischen zwei Mitternachtstimestamps.

Die Beschriftung Abszisse, der Legende und der Ordinate findet in den nächsten Schritten statt.

Im Weiteren werden die Visualisierung und der Diagrammtitel erstellt und es findet die Positionsberechung für die Abszissebeschriftung statt. Zum Abschluss wird der Export der Darstellung als JPG-Datei und PDF-Dokument vorgenommen.

6.3 Beschreibung der Layer-4-Visualisierung

Abbildung 6.2 zeigt einen Überblick der genutzten Skripte der Layer-4-Visualisierung. Es ist eine Unterteilung in Datenverarbeitung, Visualisierung und Interaktion vorgenommen worden. Im Folgenden wird der Ablauf der Visualisierung anhand der gewählten Unterteilung näher erklärt.

6.3.1 Ablauf der Datenverarbeitung und der Visualisierung

Das Skript "**visualisierung_layer_4**" startet die Visualisierung und die Datenexploration für die Messdaten, die im Format, wie in Kapitel 6.1.1 beschrieben, vorliegen. Nach der Definition der Pfade, in denen die Messdaten liegen und die

Exportdokumente geschrieben werden, findet das Einlesen der Messdaten statt. Es ist in den folgenden Programmzeilen möglich, gespeicherte Arbeitsumgebungen oder Testdaten für die Visualisierung einzulesen anstatt die Messdaten selbst.

Im folgenden Ablauf findet eine Plausibilitätsprüfung der eingelesenen Messdaten statt. Dabei werden alle Werte markiert, die kleiner als die erste Timestamp und größer als der maximale Wert sind, der bei den TCP-Ports zulässig ist. Die markierten Messpunkte werden bei der Visualisierung nicht berücksichtigt.

Im folgenden Schritt werden aus den plausiblen Messpunkten die Timestamps extrahiert und daraus Sichten erstellt.

Das Skript "**cumulating**" dient der weiteren Datenverarbeitung. Dort findet eine Sortierung der Messpunkte nach Größe der übertragenen Bytes statt. Anhand des Parameters "top_talk" kann die Anzahl der Einträge der Toptalk-Liste bestimmt werden (Liste mit Verbindungen, bei denen die meisten Daten übertragen wurden).

Der folgende Abschnitt des Skriptes dient der Zuordnung des ermittelten Toptalk-Wertes zur dazugehörigen Timestamp. Im weiteren Verlauf des Skriptes werden die in den beiden vorangegangen Schritten ermittelten Werte in eine Matrix geschrieben.

Mit Hilfe des folgenden Programmabschnittes findet eine Summierung aller gleichen IN-OUT-TCP-Portkombinationen statt. Durch Erstellen einer Matrix des Typs "Sparse" aus den Vektoren "in", "out" und "byte" findet eine Summierung der entsprechenden Werte der dritten Spalte statt, wenn die Werte der ersten und zweiten Spalte gleich sind. Im vorliegenden Fall geschieht dies bei gleichen IN- und OUT-TCP-Ports.

Es ist vor der Erstellung der Matrix der Datentyps "Sparse" notwendig, die IN- und OUT-Ports um eins zu inkrementieren, da der Wert "0" bei Sparse-Matritzen nicht zugelassen ist, es aber möglich ist, dass der Wert "0" vorkommen kann.

Nach dem Zusammenfassen wird die Sparse-Matrix in drei Vektoren aufgeteilt und der IN- und OUT-Vektor wird um jeweils eins dekrementiert.

Zum Abschluss des Skripts "cumulating" wird der Diagrammtitel aus Daten der ersten und letzten dargestellten Timestamps erstellt.

Die folgenden Skripte "**start_plot**" und "xyzscale" dienen der Initialisierung der Visualisierung und der Bedienoberfläche der Interaktion.

Der Parameter "pixrng" bestimmt die Auflösung der Visualisierung. Er gibt an, wie viele Messpunkte zu einem Pixel zusammengefasst werden. Der Parameter "pixrng05" dient dem Ausgleich der durch das Skript "ffgrid" bedingten

Verschiebung der Visualisierung in X- und Y-Richtung.

Das Skript "ffgrid" teilt die Daten der Eingangsvektoren in äquidistante Teile der Größe "pixrng" und fasst diese Teile zu einem Datenwert zusammen. Diese Datenwerte sind die Grundlage für die daraus resultierende Pixelvisualisierung.

Auf der Abszisse wird der erste Eingangsvektor (incoming TCP-Port), auf der Ordinate der zweite Eingangsvektor (outgoing TCP-Port) dargestellt. Auf dem resultierenden Pixel in der Visualisierung wird ein dem Wert des dritten Eingangsvektors (übertragen Bytes) entsprechender Farbwert abgebildet.

Im weiteren Verlauf der Skriptes "start_plot" wird die Konfiguration der Farbpalette der Visualisierung angepasst und die Beschriftung der Abszisse und Ordinate vorgenommen. Mit Aufruf des Skriptes "xyzscale" wird die Einrichtung der Interaktionsoberfläche gestartet. Zum Abschluss des Skriptes findet der Export der Visualisierung als JPG-Datei und als PDF-Dokument statt.

6.3.2 Interaktionselemente

Das Skript "**xyzscale**" bestimmt Position, Art, Eigenschaften und Callback-Funktion der Interaktionselemente.

Die Skripte "**zoomx**", "**zoomy**","**scrolly**", "**scrollx**" und "**resizeaxes**" erfüllen alle eine ähnliche Funktion. Mit diesen Skripten werden die Darstellungsgrenzen der Abszisse und Ordinate verändert. Es wird eine Filterung der dargestellten Daten vorgenommen; die Werte, die dem Filter nicht entsprechen, werden nicht dargestellt. Mit Hilfe des Skriptes "resizeaxes" kann die Darstellung wieder auf die Ausgangswerte zurückgesetzt werden.

"**togglezoom**" ermöglicht ein gleichzeitiges Zoomen der Abszisse und Ordinate. Die eigentliche Funktionalität dieses Skriptes befindet sich im Skript "zoomx".

Die erwähnten Skripte rufen das Skript "**setticks**" auf. Dort findet die Korrektur des Darstellungsoffsets bedingt durch das Darstellungsskript "ffgrid" statt. Dabei werden die Markierungen der Abszisse und Ordinate um den Parameter "pixrng05" verschoben.

"**toggletop**" aktiviert die Liste der TCP-Portkombinationen, die im dargestellten Beobachtungszeitraum den größten Datendurchsatz verursacht hatten. Des Weiteren kann durch Auswahl eines Wertes der Anzeigenliste eine Markierung in Form eines Pfeils auf das korrespondierende Pixel in der Visualisierung aktiviert werden.

Das Skript "**timestamp**" stellt Funktionen bereit, mit denen die Möglichkeit besteht, bestimmte Zeitabschnitte des Beobachtungszeitraumes darzustellen. Dies

wird mittels Filterung bestimmter Timestamps realisiert. Damit ändert sich der für die Darstellung relevante Beobachtungszeitraum. Die Skripte "cumulating" und "scalez" müssen dafür ausgeführt werden. Das Skript "**scalez**" dient u.a. dazu , dass mittels des Aufrufs von "ffgrid" die Visualisierung mit dem neuen Beobachtungszeitraum aktualisiert wird. Eine weitere Funktionalität von "scalez" besteht darin, dass die Maximalwerte der übertragenen Bytes schrittweise ausgeblendet werden können, um einen besseren Überblick über die Verteilung TCP-Portkombinationen zu erhalten. Des Weiteren kann eine Filterung der angezeigten TCP-Portkombinationen nach Auslastung der Bandbreite vorgenommen werden. So ist es möglich alle TCP-Portkombinationen anzeigen zu lassen, die beispielsweise 50 - 80 % der höchsten aufgetretenen Datenrate haben.

6.4 Zusammenfassung

Das realisierte Tool zur visuellen Analyse paketorientierten Datenverkehrs umfasst Visualisierungen des Layer-3- und Layer-4-Datenverkehrs. Die Layer-3-Visualisierung stellt die genutzte Datenrate der verschiedenen Protokolle auf der Zeitachse summiert oder nach ein- und ausgehendem Datenverkehr getrennt dar. Dabei kann die Datenrate die Anzahl der Frames je Zeiteinheit, die Größe der Frames je Zeiteinheit oder die Größe der Nutzdaten je Zeiteinheit sein. Bei der Darstellung werden nur die Protokolle berücksichtigt, die auf der Visualisierung sichtbar sind. Die Layer-4-Visualisierung stellt die Datenrate nach IN-OUT-TCP-Portkombinationen dar. Es besteht die Möglichkeit, entlang der vierten Dimension (Zeit) zu navigieren und diese mit in die Visualisierung einfließen zu lassen. Des Weiteren kann mittels Interaktion, Filterung, Zoom und Markierung eine weitergehende explorative Analyse der Messdaten vorgenommen werden.

Kapitel 7
AUSWERTUNG

Die Aussage der Visualisierungen der Daten in Abbildungen 7.1 - 7.6 haben keine wissenschaftliche Relevanz. Zwar handelt es sich um Messdaten, jedoch ist es nicht bekannt inwieweit diese durch vorhergehende Analysen verändert wurden.

7.1 Auswertung der Layer-3-Visualisierung

Unter anderem standen folgende Visualisierungen nach Analyse der Attribute der Übertragungsprotokolle des OSI-Layer 3 (siehe Kapitel 2.4) als mögliche Darstellungen für die entwickelte Visualisierung zur Auswahl.

In Abbildung 7.1 wird auf der Ordinate die Richtung des Verkehrs und die genutzten Protokolle des Layer-3-Payloads durch verschiedene Farben repräsentiert. Auf der Abszisse ist der Zeitraum der Beobachtung aufgetragen.

Diese Darstellungsform ist nicht akzeptabel, da zu viele Graphen in der Abbildung gezeigt werden. Damit ist es dem Betrachter unmöglich alle Informationen zu erfassen. Das Ziel der Visualisierung wurde nicht erreicht, da die dargestellten Daten nicht sofort erkannt und bewertet werden können.

Abbildung 7.2 ist im Gegensatz zu Abbildung 7.1 übersichtlicher, dies ist Folge der Reduzierung der Dimensionalität der Darstellung. Damit sinkt aber der Informationsgehalt der Darstellung. Auf der Ordinate sind in Abbildung 7.2 nur noch die summierten Werte des ein- und ausgehenden Verkehrs aufgetragen.

Die Nutzung der positiven und negativen Ordinatenrichtung für den ein- und ausgehenden Verkehr in Abbildung 7.3 revidiert die Einschränkung, die in Abbildung 7.2 durch die Reduzierung der Dimensionalität getätigt wurde. Durch die Nutzung der positiven und negativen Ordinatenrichtung ist es dem Betrachter sehr schnell möglich, einen Vergleich der Protokolle in den jeweiligen Verkehrsrichtungen zu tätigen. Sind die Datenraten zweier Protokolle in Verlauf der Messung nahezu gleich, kommt es zu einer Überlagerung beider. Dies ist in Abbildung 7.1 beim UDP-Protokoll der Fall. Bei diesem Beispiel ist das Problem der Überlagerung durch die Aufteilung in positive und negative Ordinatenrichtung für die Darstellung nicht mehr relevant.

Werden, wie in Abbildung 7.4, die Datenraten der genutzten Layer-3-Protokolle

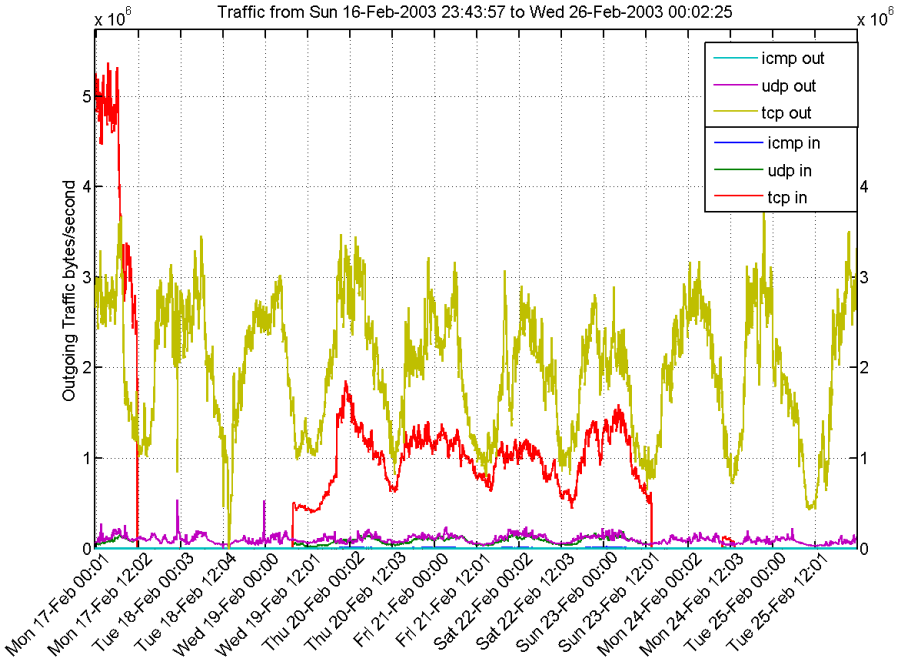

Abbildung 7.1: Layer-3-Visualisierung mit Unterscheidung des ein- und ausgehenden Verkehrs in positiver Ordinatenrichtung

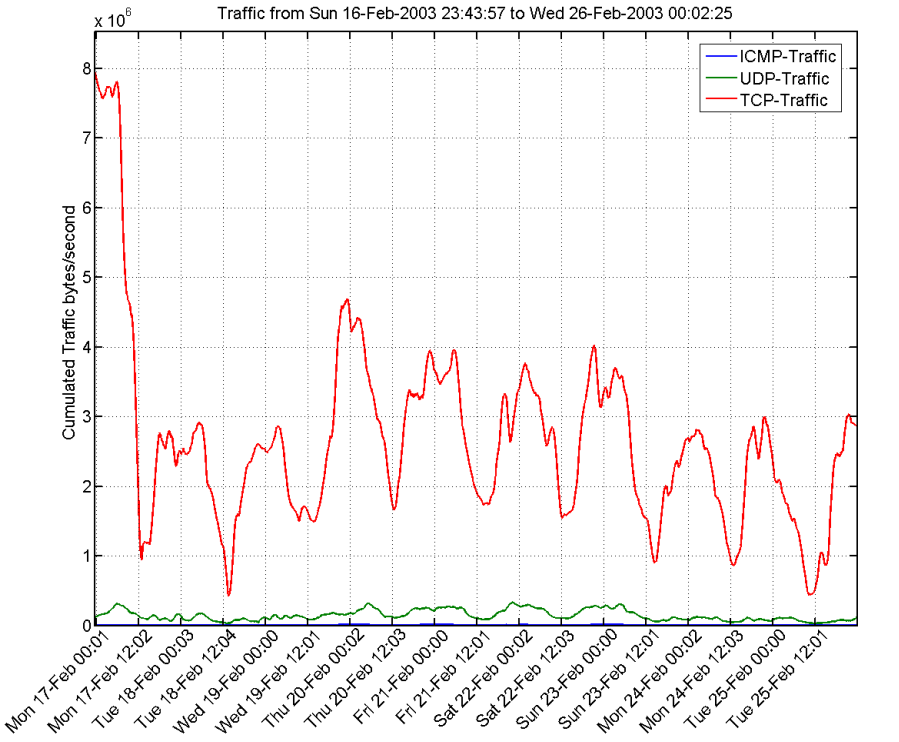

Abbildung 7.2: Layer-3-Visualisierung; gesamter Verkehr in positiver Ordinatenrichtung

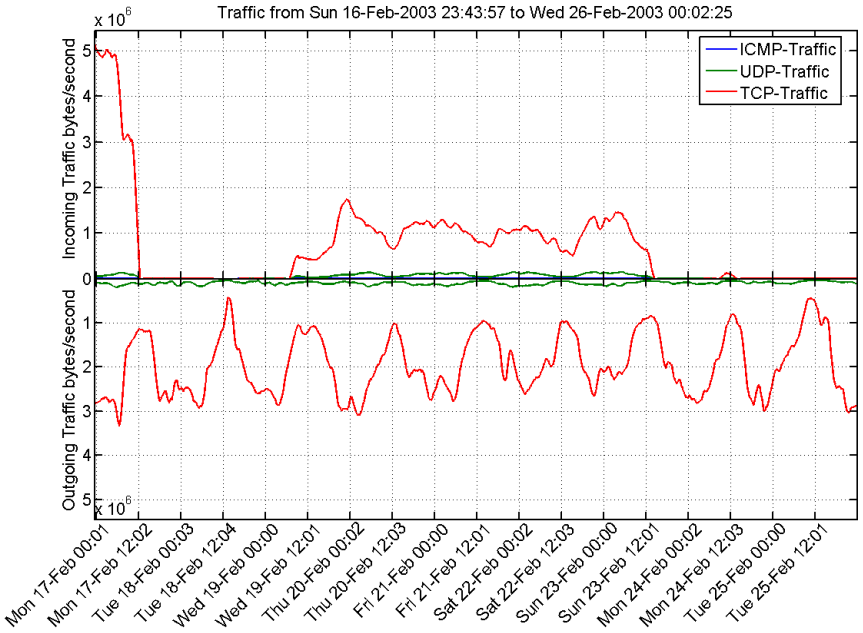

Abbildung 7.3: Layer-3-Visualisierung mit Unterscheidung des ein- und ausgehenden Verkehrs in negativer und positiver Ordinatenrichtung

gestapelt dargestellt, kann es zu keiner Überlagerung der Datenrate kommen. Es ist sinnvoll, die Protokolle mit der geringsten Datenrate bei der gestapelten Darstellung am dichtesten zur Abszisse aufzutragen. Das Ausfüllen des relevanten Bereiches in der Darstellung mit der dem Protokoll repäsentierenden Farbe erleichtert die Zuordnung und die Bewertung der Datenrate des jeweiligen Protokolls. Mit Hilfe dieser Darstellungsform ist es möglich, die Datenrate aller Protokolle zu erfassen, da die Einzelwerte der Protokolle gestapelt dargestellt werden und es somit zu keinen Überlagerungen kommen kann. Diese Visualisierung ist die effizienteste Darstellungsform und wurde daher auch bei dem entwickelten Tool zur visuellen Analyse von paketorientiertem Datenverkehr eingesetzt.

Es ist auch möglich, diese Visualisierung für die Darstellung von Layer-4-Verkehr zu nutzen: Auf der Abszisse wird die Zeit (der Beobachtungszeitraum) aufgetragen. Die Ordinate bildet den Datenverkehr in charakteristischen TCP-Portbereichen ab (z.B. Port 80 und die im Verbindungsaufbau festgelegten dynamischen Ports).

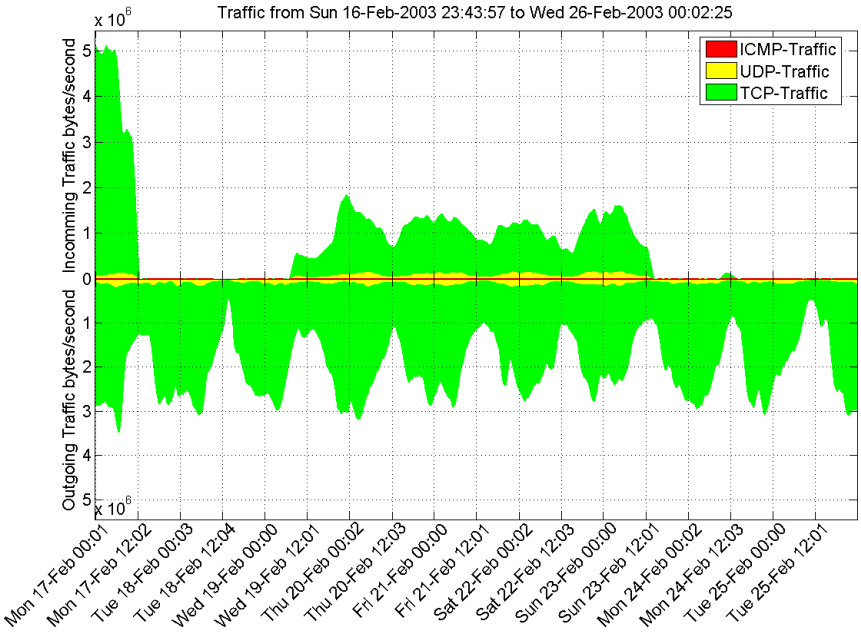

Abbildung 7.4: Visualisierung des Layer-3-Verkehrs gestapelt nach Verkehrsrichtung getrennt

7.2 Auswertung der Layer-4-Visualisierung

Abbildung 7.5 zeigt die übertragenen Daten eines Beobachtungszeitraumes dargstellt nach dem eingehenden TCP-Port (Abszisse) und dem ausgehenden TCP-Port (Ordinate). Die resultierende Datenmenge einer beliebigen Portkombination wird im Darstellungsraum durch die Farbe und Höhe eines Balkens am Schnittpunkt der TCP-Ports abgebildet. Es ist nicht möglich, einen vollständigen Überblick über alle Daten zu erhalten, da sich in der 3D-Ansicht die Balken überlagern. In Abbildung 7.5 wurde die Darstellung um eine Dimension verringert. Die Höhe der Balken wird bei der Darstellung nicht mehr berücksichtigt. Dieses Darstellungsattribut ist redundant, da die Menge der übertragenen Daten in Abbildung 7.5 zusätzlich noch als Farbe abgebildet wird. Auf Abbildung 7.6 sind alle Messdaten ohne eine Überlagerung zu erkennen. Diese Darstellungsform wurde bei der Implementierung der Tools zur visuellen Analyse genutzt.

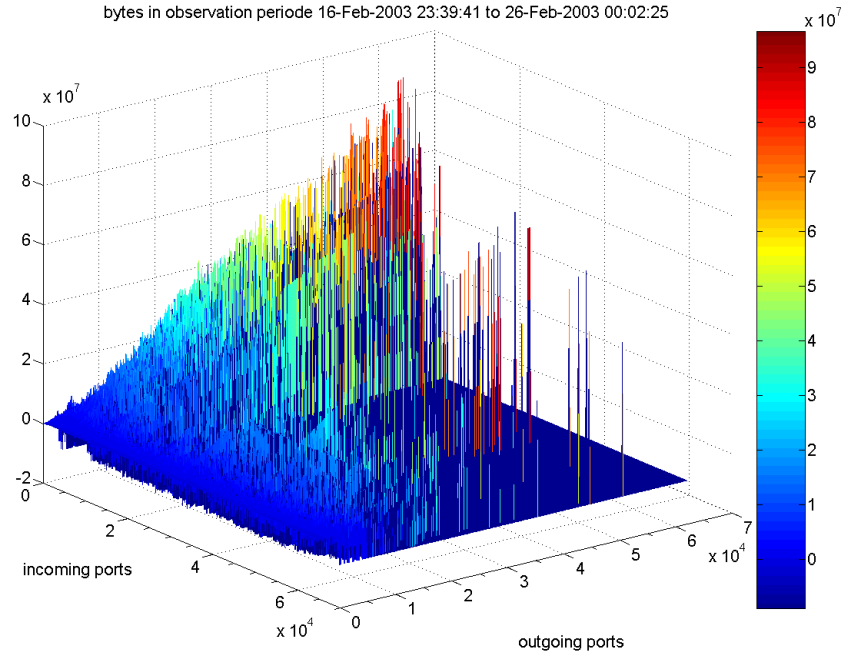

Abbildung 7.5: 3D-Visualisierung des Layer-4-Verkehrs nach ein- und ausgehenden TCP-Ports

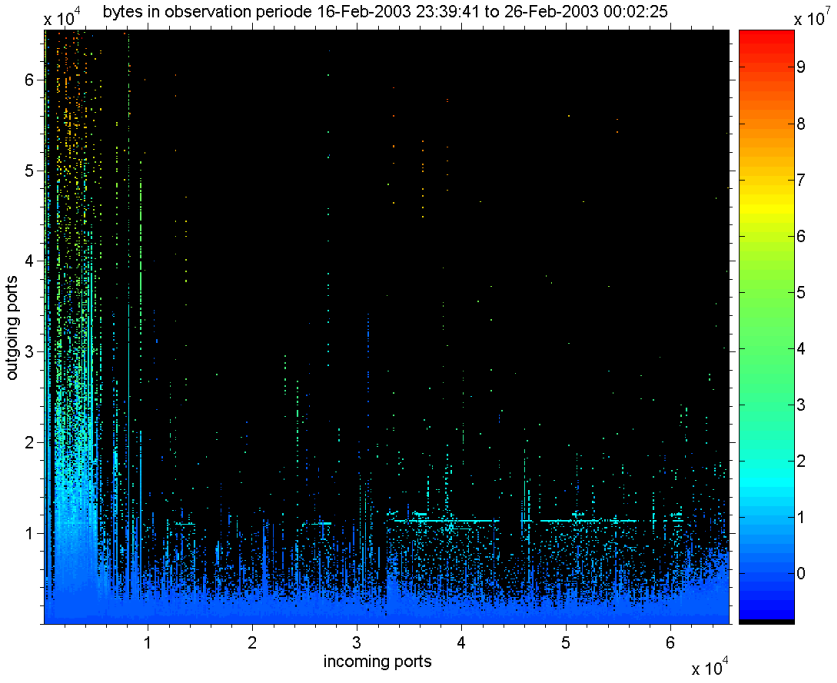

Abbildung 7.6: Pixelvisualisierung des Layer-4-Verkehrs nach ein- und ausgehenden TCP-Ports

Kapitel 8

ZUSAMMENFASSUNG UND AUSBLICK

Die ständige Verfügbarkeit und eine garantierte Mindestqualität von Datennetzen sind für die heutige Informationsgesellschaft unabdingbar. Mit einer Vergrößerung des Anwendungspektrums der Datennetze erhöhen sich auch die Anforderungen in Bezug auf Qualität und Verfügbarkeit der Datennetze. Mit der Analyse der Verkehrscharakteristik bestehender Datennetze können Rückschlüsse auf die applikations- und nutzerspezifischen Anteile am Gesamtverkehrsaufkommen im jeweiligen Datennetz gezogen werden.

Die Auswertung der applikations- und nutzerspezifischen Anteile fließen als Datenquellenmodell in ein Simulationsmodell für ein Datennetz ein. Damit ist es möglich, z.B. den Datendurchsatz oder die Verzögerung der Datenübertragung in einem Datennetz zu simulieren. Dies ist bei der Planung von Netzwerken hilfreich und kann z.B. Aussagen darüber geben, ab wie vielen Nutzern ein Netzwerk in bestimmten Parametern nicht mehr den geforderten Qualitätsansprüchen genügt.

Die Visualisierung des Kommunikationsaufkommens unterstützt die Auswertung der applikations- und nutzerspezifischen Anteile des Datenverkehrs dadurch, dass Zusammenhänge schneller erkannt werden, als das mit statistische Methoden möglich ist.

Um die applikations- und nutzerspezifischen Anteile des Datenverkehrs darstellen zu können, muss Klarheit über den Aufbau der Kommunikationsstruktur herrschen. Daher wurde in Kapitel 2 anhand des OSI-Schichtenmodells der Aufbau der Kommunikationsstruktur in paketorientierten Netzen erläutert. Es wurden die vier untersten Schichten des OSI-Modells nach Attributen untersucht, die eine parameterorientierte Darstellung des Datenverkehrs erlauben. Die Bitübertragungsschicht und die Sicherungsschicht stellen keine oder nur ungenügend verkehrsbeschreibende Attribute zu Verfügung, anhand derer eine Charakterisierung des Datenverkehrs möglich ist.

Die Untersuchung der verkehrsbeschreibenden Parameter der Vermittlungsschicht wurde im Kapitel 2.4 anhand der Protokolle IPv4 und IPv6 vorgenommen. Es wurden sieben Attribute dargestellt, anhand derer eine Klassifizierung der Layer-3-Datenverkehrs vorgenommen werden kann.

Das TCP-Protokoll als Übertragungsprotokoll der Transportschicht stellt vier Attribute bereit, die für eine Verkehrscharakterisierung genutzt werden können. Die daraus resultierenden Darstellungsmöglichkeiten sind in Kapitel 2.6 näher erläutert.

Um eine effiziente Visualisierung der aufgezeigten Darstellungsmöglichkeiten realisieren zu können, wurden in Kapitel 3 Visualisierungsverfahren auf ihre Relevanz hin untersucht. Dabei wurde der Analyseprozess der Visualisierung in drei Schritte (explorative Analyse, konfirmative Analyse und Präsentation) unterteilt. Da die Anwendung der Visualisierung im Rahmen dieser Arbeit nur den ersten Schritt des Analyseprozesses umfasst, wurde auf die konfirmative Analyse und Präsentation nicht näher eingegangen . Die Verfahren der explorativen Analyse sind in drei Klassen (zu visualisierender Datentyp, verwendete Visualisierungstechnik und verwendete Techniken für Interaktion und Verzerrung) eingeteilt worden. Jede Klasse wurde detailliert vorgestellt und es wurden ausführlich Beispiele aufgezeigt. Für die sieben Attribute der Vermittlungs- und die vier Attribute der Transportschicht wurden in Kapitel 3.8 anhand der Dimension und Wertebereiche der Attribute die effizientesten Visualisierungen ausgewählt und beschrieben.

Für zwei Darstellungsmöglichkeiten wurde ein Visualisierungstool mit MATLAB 7.0 entworfen. Die erste Visualisierung stellt den Anteil der Layer-3-Payload-Protokolle an der Gesamtübertragungsrate nach der Zeit dar, wobei nach ein und ausgehendem Datenfluss unterschieden wurde. Dies wurde mittels eines X-Y-Plots realisiert. (Abbildung 7.4) Durch eine einfache Modifikation des Tools in weiterführenden Arbeiten kann die Darstellung von Layer-4-Verkehr erreicht werden. Anstatt der Protokolle werden auf der Ordinate verschiedene TCP-Portbereiche abgebildet. Die zweite implementierte Darstellungsmöglichkeit ist eine Pixelvisualisierung, bei der die ein- und ausgehenden TCP-Ports auf der Abszisse bzw. Ordinate aufgetragen sind. Deren Schnittpunkte stellen in der Matrix eine Farbkodierung der Menge der übertragenen Daten dar. Mit Hilfe von Zooming- und Markierungstechniken wird eine Interaktion mit den Daten in der Darstellung erreicht. Diese Visualisierung lässt sich nach Modifikation für die Darstellung aller Attribute verwenden, bei denen eine Pixelvisualisierung vorgeschlagen wurde.

Für den Aufbau eines Messsystems für paketorientierte Daten wurde ein ähnliches System wie in [KMV03] beschrieben verwendet. Es wurde eine Performanceanalyse durchgeführt, die zum Ergebnis führte, dass das Messsystem den Anforderungen zur Messung des Datenverkehrs in 100 Mbit-Vollduplex-Ethernet-Netzwerk genügt. (Kapitel 4.2) Des Weiteren wurde eine Messung im Instituts-

netzwerk über den Zeitraum von einer Woche durchgeführt. Die Auswertung dieser Messung wird im Rahmen weiterführender Arbeiten in den Diplomarbeiten von Herrn Groth zum Thema "Analyse von IP Datenverkehr im Transport Layer" und von Herrn Kosubeck zum Thema "Vergleichende Betrachtungen statistischer Verfahren zur Beschreibung von Datenverkehr in IP Netzen" vorgenommen.

Um die gemessenen Daten für die nachfolgende Analyse in einer möglichst handhabbaren und flexiblen Form speichern zu können, wurde ein generisches, am IPFIX-Datenformat orientiertes, Datenspeicherformat entwickelt. Aus Speichereffizienzgründen wurden für IPv4 und IPv6 unterschiedliche Datenformate erstellt.

Die Messungen, die im Rahmen der erwähnten weiterführenden Diplomarbeiten getätigt wurden, sind mit dem im Rahmen dieser Arbeit erstellten Messsystem vorgenommen worden. Des Weiteren wird das entwickelte Visualisierungstool zur Analyse der Messdaten genutzt.

Literatur

[AC91] B. Alpern and L. Carter. Hyperbox. In Proc. Visualization '91, San Diego, CA, pages 133–139, 1991.

[ADLP95] V. Anupam, S. Dar, T. Leibfried, and E. Petajan. Dataspace: 3D visualization of large databases. In Proc. Int. Symp. on Information Visualization, Atlanta, GA, pages 82–88, 1995.

[AK02] J. Abello and J. Korn. Mgv: A system for visualizing massive multidigraphs. Transactions on Visualization and Computer Graphics, 2002.

[AKK96] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In Proc. Visualization 96, Hot Topic Session, San Francisco, CA, 1996.

[And72] D. F. Andrews. Plots of high-dimensional data. Biometrics, 29:125–136, 1972.

[AS94] C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA, pages 313–317, 1994.

[Asi85] D. Asimov. The grand tour: A tool for viewing multidimensional data. SIAM Journal of Science & Stat. Comp., 6:128–143, 1985.

[Ber82] J. Bertin. Graphische Darstellungen und die graphische Weiterverarbeitung der Information. de Gyters, Berlin, New York, 1982.

[BETT99] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. Graph Drawing. Prentice Hall, 1999.

[BH94] B. B. Bederson and J. D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In Proc. UIST, pages 17–26, 1994.

[BSC96] A. Buja, D. F. Swayne, and D. Cook. Interactive high-dimensional data visualization. Journal of Computational and Graphical Statistics, 5(1):78–99, 1996.

[BSP+93] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: The see-through interface. In Proc. SIGGRAPH '93, Anaheim, CA, pages 73–80, 1993.

[CCF97] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Ieee computer graphics and applications, special issue on information visualization. IEEE Journal Press, 17(4):42–51, July 1997.

[Che73] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. Journal Amer. Statistical Association, 68:361–368, 1973. 18

[Che99] C. Chen. Information Visualisation and Virtual Environments. Springer-Verlag, London, 1999.

[Cle93] W. S. Cleveland. Visualizing Data. AT&T Bell Laboratories, Murray Hill, NJ, Hobart Press, Summit NJ, 1993.

[CMS99] S. Card, J. Mackinlay, and B. Shneiderman. Readings in Information Visualization. Morgan Kaufmann, 1999.

[CWL96] D. B. Carr, E. J. Wegman, and Q. Luo. Explorn: Design considerations past and present. In Technical Report, No. 129, Center for Computational Statistics, George Mason University, 1996.

[Dod01] M. Dodge. Web visualization. http://www.cybergeography.org /atlas/geographic.html, Sep 2004.

[End04] Endace Measurement Systems. ERF Format Rev B. http://www.endace.com/support/EndaceRecordFormat.pdf, Dez 2004.

[EHJS01] J. Eagan, M. J. Harrold, J. A. Jones, and J. Stasko. Visually encoding program test information to find faults in software. In Technical Report, Georgia Institute of Technology, GIT-GVU-01-09, 2001.

[FB90] S. Feiner and C. Beshers. Visualizing n-dimensional virtual worlds with n-vision. Computer Graphics, 24(2):37–38, 1990.

[Fur86] G. Furnas. Generalized fisheye views. In Proc. Human Factors in Computing Systems CHI 86 Conf., Boston, MA, pages 18–23, 1986.

[Gos95] J. Gosling. Visualisierungtool für vergleichsorientierte Sortieralgorithmen. SortItem.java 1994-1995. http://cg.scs.carleton.ca/m̃orin/misc/sortalg/,2004.

[HBkc+01] B. Huffaker, A. Broido, k. claffy, M. Fomenkov, S. McCreary, D. Moore, and O. Jakubiec. Visualizing internet topology at a macroscopic scale. In http://www.caida.org/analysis/topology/as_core_network/, 2004.

[Hea95] M. Hearst. Tilebars: Visualization of term distribution information in full text information access. In Proc. of ACM Human Factors in Computing Systems Conf. (CHI'95), pages 59–66, 1995. 19

[Hol97] H. Holtkamp. Einführung in TCP/IP. Universität Bielefeld, http://www.rvs.uni-bielefeld.de/~heiko/tcpip/, 2004.

[HHNW02] S. Havre, B. Hetzler, L. Nowell, and P. Whitney. Themeriver: Visualizing thematic changes in large document collections. Transactions on Visualization and Computer Graphics, 2002.

[Hub85] P. J. Huber. The annals of statistics. Projection Pursuit, 13(2):435–474, 1985.

[ID90] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In Proc. Visualization 90, San Francisco, CA, pages 361–370, 1990.

[IPFIX04] IETF networking-group. IP Flow Information Export (ipfix). http://www.ietf.org/html.charters/ipfix-charter.html, Okt 2004.

[JS91] B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information. In Proc. Visualization '91 Conf, pages 284–291, 1991.

[Kei00] D. Keim. Designing pixel-oriented visualization techniques: Theory and applications. Transactions on Visualization and Computer Graphics, 6(1):59–78, Jan–Mar 2000.

[Kei01] D. Keim. Visual exploration of large databases. Communications of the ACM, 44(8):38–44, 2001.

[KH98] D. Keim and A. Herrmann. The gridfit approach: An efficient and effective approach to visualizing large amounts of spatial data. In Proc. Visualization 98, Research Triangle Park, NC, pages 181–189, 1998.

[KHDH02] D. A. Keim, M. C. Hao, U. Dayal, and M. Hsu. Pixel bar charts: A visualization technique for very large multi-attribute data sets. Information Visualization Journal, 1(1):1–14, Jan. 2002.

[KK94] D. A. Keim and H.-P. Kriegel. Visdb: Database exploration using multidimensional visualization. Computer Graphics & Applications, 6:40–49, Sept. 1994.

[KKA95] D. A. Keim, H.-P. Kriegel, and M. Ankerst. Recursive pattern: A technique for visualizing very large amounts of data. In Proc. Visualization 95, Atlanta, GA, pages 279–286, 1995.

[KMV03] T. Kessler, H-D. Melzer, T. Vergin. Analyse des Kommunikationsaufkommens in hochbitratigen Zugangsnetzen. In Proc. of 4th IuK-Tage Mecklenburg-Vorpommern '03, Rostock, Germany, 2003.

[LA94] Y. Leung and M. Apperley. A review and taxonomy of distortion-oriented presentation techniques. In Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA, pages 126–160, 1994.

[Lev91] H. Levkowitz. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In Proc. Visualization 91, San Diego, CA, pages 22–25, 1991.

[LRP95] J. Lamping, Rao R., and P. Pirolli. A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. In Proc. Human Factors in Computing Systems CHI 95 Conf., pages 401–408, 1995.

[LWW90] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In Proc. Visualization '90, San Francisco, CA, pages 230–239, 1990.

[MB95] T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3D hyperbolic space. In Proc. VRML '95 Symp, San Diego, CA, pages 33–38, 1995.

[MKS02] N. Lopez M. Kreuseler and H. Schumann. A scalable framework for information visualization. Transactions on Visualization and Computer Graphics, 2002. 20

[MRC91] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In Proc. Human Factors in Computing Systems CHI '91 Conf., New Orleans, LA, pages 173–179, 1991.

[NOR04] W. D. Norcott: IOzone Filesystem Benchmark. 2004. http://www.iozone.org/docs/IOzone_msword_98.pdf

[PF93] K. Perlin and D. Fox. Pad: An alternative approach to the computer interface. In Proc. SIGGRAPH, Anaheim, CA, pages 57–64, 1993.

[PG88] R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. In Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ, pages 514–519, 1988.

[RC94] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representation in an interactive focus+context visualization for tabular information. In Proc. Human Factors in Computing Systems CHI 94 Conf., Boston, MA, pages 318–322, 1994.

[SA82] R. Spence and M. Apperley. Data base navigation: An office environment for the professional. Behaviour and Information Technology, 1(1):43–54, 1982.

[SCB92] D. F. Swayne, D. Cook, and A. Buja. User's Manual for XGobi: A Dynamic Graphics Program for Data Analysis. Bellcore Technical Memorandum, 1992.

[SDZ+93] Schaffer, Doug, Zuo, Zhengping, Bartram, Lyn, Dill, John, Dubs, Shelli, Greenberg, Saul, and Roseman. Comparing fisheye and full-zoom techniques for navigation of hierarchically clustered networks. In Proc. Graphics Interface (GI '93), Toronto, Ontario, 1993, in: Canadian Information Processing Soc., Toronto, Ontario, Graphics Press, Cheshire, CT, pages 87–96, 1993.

[Shn92] B. Shneiderman. Tree visualization with treemaps: A 2D space-filling approach. ACM Transactions on Graphics, 11(1):92–99, 1992.

[Shn96] B. Shneiderman. The eye have it: A task by data type taxonomy for information visualizations. In Visual Languages, 1996.

[Shn99] B. Shneiderman. Dynamic queries, starfield displays, and the path to spotfire. In http://www.cs.umd.edu/hcil/spotfire, 1999. 21

[SM00] H. Schumann and W. Müller. Visualisierung: Grundlagen und allgemeine Methoden. Springer, 2000.

[Sma01] SmartMoney. Marketcap treemap visualization. http://www.smartmoney.com/marketmap/, 2004.

[Spe00] B. Spence. Information Visualization. Pearson Education Higher Education publishers, UK, 2000.

[Spo93] A. Spoerri. Infocrystal: A visual tool for information retrieval. In Proc. Visualization '93, San Jose, CA, pages 150–157, 1993.

[STH02] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. Transactions on Visualization and Computer Graphics, 2002.

[Tie91] L. Tierney. Lispstat: An object-orientated environment for statistical computing and dynamic graphics. In Wiley, New York, NY, 1991.

[War94] M. O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In Proc. Visualization 94, Washington, DC, pages 326–336, 1994.

[War00] C. Ware. Information Visualization: Perception for Design. Morgen Kaufman, 2000.

[Wis99] J. A. Wise. The ecological approach to text visualization. Journal of the American Society for Information Science, 50(13):1224–1233, 1999.

[WTP+95] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, Schur A., and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In Proc. Symp. on Information Visualization, Atlanta, GA, pages 51–58, 1995.

[WUT95] A. Wilhelm, A.R. Unwin, and M. Theus. Software for interactive statistical graphics - a review. In Proc. Int. Softstat 95 Conf., Heidelberg, Germany, 1995.

Erklärung

Ich erkläre, diese Arbeit selbstständig angefertigt und die benutzten Unterlagen vollständig angegeben zu haben.

Rostock, 15. Januar 2005

Udo Brunswig