Sichere Echtzeitübertragung in der Medizin-Telematik

Dissertation

der Fakultät für Informations- und Kognitionswissenschaften der
Eberhard-Karls-Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Dipl.-Inform. Kai Kamphenkel
aus Braunschweig

Tübingen
2009

Tag der mündlichen Qualifikation:	10.02.2010
Dekan:	Prof. Dr.-Ing. Oliver Kohlbacher
1. Berichterstatter:	Prof. Dr.-Ing. Georg Carle
2. Berichterstatter:	Prof. Dr. Andreas Schilling

Meinen lieben Eltern

Zusammenfassung

Ziel der Arbeit ist die Analyse von Telemedizin-Szenarien und die Entwicklung neuer technischer Ansätze zur Realisierung von sicherer Echtzeitübertragung in der Medizin.

In der im Vorfeld stattgefundenen Untersuchung am Klinikum in Tübingen hat sich das erweiterte Notfallszenario in Form der Expertenkonsultation unter Verwendung von präklinischer Sonographie als besonders interessant und technisch anspruchsvoll herausgestellt. Neben der technischen Umsetzung gilt es, auch die Integration in bestehende medizinische Workflows zu realisieren. Für den technischen Bereich hat sich herausgestellt, dass eine auf adaptiven Algorithmen beruhende Komponente, das Intelligente Netz (IN), in der Lage ist, einen performanten parallelen Datentransfer zu realisieren.

In einem Multipfad-SCTP-Szenario (MP-SCTP) wird die Multihoming-Eigenschaft des Stream Control Transmission Protocols (SCTP) genutzt, um Datenchunks auf verschiedene Ende-zu-Ende-Pfade zu verteilen. SCTP nutzt ursprünglich nur einen Pfad – den Primärpfad – für den Datentransfer, die weiteren Sekundärpfade werden hauptsächlich zur Ausfallsicherung verwendet. Die hieraus resultierenden technischen Probleme sind in der Literatur bereits umfangreich untersucht und größtenteils gelöst worden.

Die vorliegende Arbeit fokussiert auf die Pfadwahl in einem heterogenen Netzwerk, bei dem die Güte, insbesondere die Kapazitäten der einzelnen Pfade, im Vorfeld nicht bekannt ist. Das häufig zur Verteilung eingesetzte Loadsharing, also die gleichmäßige Verteilung der Daten auf alle zur Verfügung stehenden Kanäle, ist in einem heterogenen Umfeld nicht zur Pfadwahl geeignet.

Die Pfadwahl muss sich autonom und flexibel, insbesondere ohne A-priori-Informationen hinsichtlich der Dienstgüte der einzelnen Pfade, an die aktuelle Netzsituation anpassen. Hierfür ist die Beurteilung der Pfade hinsichtlich ihrer Güte notwendig. Diese Beurteilung kann effizient unter Verwendung von Algorithmen aus dem Bereich der adaptiven Algorithmen erfolgen. Grundlage für die Beurteilung bilden von SCTP bereitgestellte Parameter, welche die aktuelle Netzsituation jeweils von einem isolierten Standpunkt aus beschreiben. Da die Pfadwahl zum Zeitpunkt des physikalischen Einspeisens der Datenchunks auf den jeweiligen Pfad erfolgt, muss die Pfadwahl in der Transportschicht erfolgen.

Abschließend werden Überlegungen eingebracht, die es erlauben, das erfolgreiche Konzept der automatischen Pfadwahl auf die Wahl der notwendigen Sicherheitslösungen zu erweitern. Es wird ein Maß vorgestellt, welches einen Vergleich von verschiedenen Verschlüsselungsverfahren und deren Schlüssellängen erlaubt, falls der Datendurchsatz durch mangelnde Host-Performance nicht optimal ist.

Stichwörter Multipath-SCTP, Intelligentes Netz, heterogene Netze, Pfadwahl, SCTP, Sicherheitslösungen, Secure-SCTP, medizinische Workflows, Medizin-Telematik

Danksagung

Die vorliegende Dissertation ist im Rahmen einer Kooperation des Lehrstuhls für Rechnernetze und Internet der Universität Tübingen, der Radiologischen Universitätsklinik in Tübingen und der Siemens AG, Medical Solutiouns entstanden. An dieser Stelle möchte ich mich bei all denjenigen bedanken, die dazu beigetragen haben, dass ich diese Arbeit fertigstellen konnte.

Mein ganz persönlicher Dank gilt Herrn Prof. Dr. Georg Carle, ehemaliger Leiter des Lehrstuhls für Rechnernetze und Internet der Universität Tübingen und aktuell Leiter des Lehrstuhls für Netzarchitekturen und Netzdienste der Technischen Universität München, durch dessen Initiative mir die Möglichkeit zu dieser interessanten Arbeit gegeben wurde und der mir immer fordernd und fördernd zur Seite stand.

Herr Dr. Jens Bauer von der Radiologischen Universitätsklinik in Tübingen hat durch seine medizinische Sicht die Arbeit sehr bereichert, vielen Dank hierfür.

Seitens der Siemens AG möchte ich mich sehr bei Frau Dr. Susanne Laumann für die langjährige Unterstützung und der thematischen Ausrichtung der vorliegenden Arbeit, bei Herrn Dr. Markus Blank für die enge Zusammenarbeit am Imaging Science Institute in Tübingen und bei Herrn Thorsten Koopmann für die Unterstützung bei der Beendigung des Projekts bedanken.

Nicht zuletzt möchte ich mich bei meiner Frau Nataliya für ihre Geduld und Ihr Verständnis bedanken und mich bei meiner Tochter Alicia entschuldigen, dass sie Ihren Vater häufig vermissen musste.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Die Radiologische Universitätsklinik in Tübingen betreibt gemeinsam mit der Siemens AG ein Forschungs- und Entwicklungsinstitut, das Imaging-Science-Institute (ISI). Ziel ist die Forschung im Bereich der bildgebenden Diagnostik, wobei ein Schwerpunkt in der Behandlung von telemedizinischen Fragestellungen liegt.

Unter der Überschrift „Sichere Echtzeitübertragung in der Medizin-Telematik“ sollen Verfahren, Methoden und Protokolle dahingehend untersucht werden, den Transfer von medizinischen Bilddaten performant und sicherer zu gestalten. Es ist daran gedacht, ein Telemedizin-Portal für die Steuerung und Kontrolle des Datenflusses zu konstruieren, welches als übergeordnetes System in der Lage ist, die verschiedensten Anwendungen, Systeme und Institutionen miteinander zu verbinden.

Die Unterstützung von Ärzten durch IT-Technologien beschränkt sich fast ausschließlich auf die „klassische“ Datenverarbeitung. Es kann Bildmaterial gespeichert, archiviert und zur Befundung ausgelesen und bearbeitet werden. Hierfür stehen seit Jahren Verfahren und Methoden zur Verfügung.

In vielen Fällen erweist sich dieses Vorgehen als unzureichend, insbesondere bei Medizin-Szenarien, die neben einer räumlichen Trennung von Datenerfassung und Datenauswertung eine sofortige Verwendung der ermittelten Informationen voraussetzt. Die aufgenommenen Daten müssen in Echtzeit übertragen werden, damit eine unmittelbare Auswertung durch das medizinische Fachpersonal erfolgen kann. Die bisher auf diesem Gebiet verwendeten Techniken haben sich als unzureichend herausgestellt, sodass noch keine funktionsfähigen Verfahren im praktischen Einsatz sind.

Es sind neue und innovative Architekturen und Verfahren zu entwickeln, die den hohen Anforderungen an Hard- und Software solcher Szenarien gerecht werden. Ein besonders wichtiges Designmerkmal ist die Integration in bestehende Systeme und die gegebene Infrastruktur. Es wird derzeit in verschiedenen Bereichen an Medizin-Portalen und Medizin-Plattformen gearbeitet, die eine Kapselung sämtlicher Arbeitsabläufe in einer einheitlichen Umgebung ermöglichen. Die Übertragung der Ergebnisse auf das Medizin-Plattform-Projekt garantiert die Anwendbarkeit in zukünftigen Systemen.

Das Konflikt-Dreieck – Echtzeit – Sicherheit – Flexibilität

Ziel der vorliegenden Arbeit ist der Entwurf und das Design von neuen und innovativen Architekturen und Verfahren für Telemedizin-Szenarien mit dem Anspruch auf

Echtzeitübertragung von medizinischen Daten. Dabei sind die drei konkurrierenden Forderungen *Echtzeit, Sicherheit und Flexibilität bzw. Anpassbarkeit* zu berücksichtigen, wie sie im *Konfliktdreieck* in Abbildung 1.1 dargestellt sind.

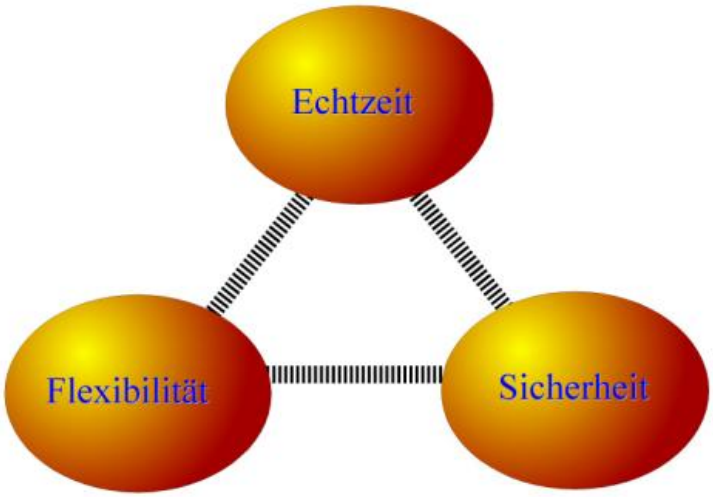


Abbildung 1.1: Das Konfliktdreieck als Basis für den Aufbau der Arbeit

Speziell in dieser Arbeit werden die Begriffe folgendermaßen ausgelegt: Die Erweiterung und Spezialisierung von bestehenden Netzprotokollen für die Echtzeitübertragung von Medizindaten entsprechen der Forderung nach *Echtzeit*. Die Erweiterung und Spezialisierung von bestehenden Netzwerkprotokollen für die sichere Datenübertragung auf Basis von kryptographischen Funktionen werden unter dem Gesichtspunkt *Sicherheit* betrachtet. Die dritte Forderung bildet den Schwerpunkt der vorliegenden Arbeit, wobei die Implementierung eines „intelligenten Netzwerks“ für die flexible, automatisierte und sichere Abwicklung von Echtzeitdatenströmen, also der Punkt *Anpassbarkeit bzw. Flexibilität*, im Vordergrund steht.

Jedem Konflikt ist ein eigener Teil der vorliegenden Arbeit gewidmet, die im Folgenden kurz vorgestellt werden; eine genaue Zuordnung der Themen zu den einzelnen Kapiteln und Abschnitten wird in Abschnitt 1.2 nachgeholt.

Echtzeit In der Medizin-Telematik werden eine Vielzahl von Szenarien betrachtet, in denen es nicht nur um die nachträgliche Auswertung, Befundung und Archivierung von medizinischen Daten geht, sondern die Daten müssen aktuell und in Echtzeit zur Verfügung stehen. Diese Szenarien stehen im Mittelpunkt dieser Arbeit. In Teil I dieser Arbeit werden zum einen Szenarien beschrieben, die bereits durch Informationstechnik unterstützt werden, zum anderen aber auch Szenarien betrachtet, die bisher nur theoretisch erörtert wurden.

In Abschnitt wird gezeigt, dass ein Bedarf an modernen Telemedizin-Anwendungen besteht. Es konnte ein Szenario mit akuter Praxisrelevanz speziell für das Universitätsklinikum Tübingen herausgearbeitet werden, nämlich die *mobile Sonographie*. Insbesondere in der Zusammenarbeit mit der Klinik der Universität Tübingen hat sich das Notfallszenario als besonders relevante Anwendung herauskristallisiert. Daher wird in weiten Teilen der vorliegenden Arbeit speziell auf dieses aktuelle Szenario verwiesen, welches auch in Zusammenarbeit mit der Firma CISCO in einem Basisprojekt thematisiert wurde.

Für die Praxis bedeutet dies eine Änderung im Standard-Ablauf bei der Versorgung von polytraumatisierten Verunfallten. Daher ergeben sich notwendige Änderungen in der Rettungskette und den zugehörigen medizinischen Algorithmen in der Prozesskette. Welche konkreten Änderungen notwendig sind, ist Thema von Abschnitt. Hierfür wurden neue Workflowaspekte herausgearbeitet und in die bestehende Architektur integriert.

Anpassbarkeit und Flexibilität Der zweite Teil der vorliegenden Arbeit setzt auf dem Transportprotokoll SCTP an, wobei hier speziell die Multi-Pfad-Übertragung untersucht wird. Standard-SCTP verfügt zwar über Mechanismen des Multihomings, also der Nutzung mehrerer IP-Adressen pro Endpunkt und somit auch die Verwendung von mehreren Pfaden zwischen den Endpunkten, diese sind aber grundsätzlich nicht für den parallelen Datentransport vorgesehen. Dies kann auch dazu führen, dass möglicherweise ein Pfad mit großer Bandbreite lediglich zur Ausfallsicherheit bereitgestellt wird und die Übertragung auf einem Pfad mit kleiner Bandbreite abgewickelt werden muss. SCTP bewertet die einzelnen Pfade nicht von sich aus. Die ausführende Applikation muss in einem solchen Fall aufgrund von zusätzlichen Informationen den „bestmöglichen“ Pfad als Primärpfad setzen.

Um auf die unterschiedlichen Netzsituationen auf den verschiedenen Pfaden reagieren zu können, muss ein flexibler Automatismus geschaffen werden, der die Daten bestmöglich auf die verwendeten Pfade verteilt. Für diese Problematik wurde in der Literatur bisher noch keine zufriedenstellende Lösung vorgestellt. In diesem Teil der Arbeit wird ein effektives und effizientes Verfahren erarbeitet, mit dem diese Lücke geschlossen wird.

Ein wesentlicher Aspekt ist die Tatsache, dass keine sogenannten A-Priori-Informationen über das Netzwerk vorliegen, das heißt, das Transportprotokoll kann auf keine Informationen wie beispielsweise eine vorgegebene Bandbreite eines Pfades zurückgreifen. Sämtliche Informationen müssen demnach aus den von SCTP bereitgestellten Parametern abgeleitet werden.

Die notwendigen Informationen über die aktuelle Leistungsfähigkeit eines Pfades wird dabei über SCTP-Parameter abgefragt, die von SCTP zur Laufzeit ständig

aktualisiert werden. Die Parameter liefern aber erst im Zusammenspiel aussagekräftige Ergebnisse. Für die Auswertung der SCTP-Parameter wird auf Algorithmen des Machine-Learnings zurückgegriffen, die es auf Basis einer vorangestellten Lernphase ermöglichen, die optimale Pfadwahl zu treffen.

Um die Ergebnisse unter möglichst vielen verschiedenen Netzkonstellationen zu überprüfen, wurde eine Teststellung installiert, wobei die adaptiven Algorithmen in Form eines „intelligenten Netzwerks“ in die Transportschicht, sprich: in den SCTP-Kern integriert wurde. Die Ergebnisse der Versuche wurden anschließend statistisch ausgewertet und bewertet.

Es konnte gezeigt werden, dass gerade bei Übertragung in Netzen, bei denen im Vorfeld keine Aussagen über die Leistungsfähigkeit der einzelnen Pfade gemacht werden kann, die adaptive Steuerung der Pfadwahl zu sehr guten Ergebnissen führt, d.h. sämtliche Pfade optimal ausgelastet werden können.

Sicherheit Der dritte Teil der vorliegenden Arbeit orientiert sich an der Forderung nach Sicherheit, die gerade im medizinischen Umfeld von großer Bedeutung ist. Die positiven Ergebnisse im Bereich der automatisierten Pfadwahl führten zu der Fragestellung, ob nicht auch durch adaptiv gesteuerte Verschlüsselung ein Performancegewinn ohne Verlust der Sicherheit erzielt werden kann.

Da es eine Vielzahl von Sicherheitslösungen auch im Zusammenspiel mit SCTP gibt, wurde untersucht, welcher Nutzen durch adaptive Verschlüsselung erzielt werden kann und welche bestehenden Sicherheitslösungen sich für den Aufbau eines IN für die Absicherung von Verbindungen eignen. Speziell für die Verwendung von Secure-SCTP wurde ein Maß definiert, mit dem es möglich wird, verschiedene Security-Suiten miteinander zu vergleichen.

Dies führt zu den im Folgenden angeführten Forschnungsfragen.

1.1 Forschungsfragen (Research Question)

Basierend auf den in dieser Einleitung vorgebrachten Vorüberlegungen können folgende Forschungsfragen formuliert werden, deren Beantwortung sich die vorliegende Arbeit zum Ziel gesetzt hat. Entsprechend der technischen wie auch medizinischen Ausrichtung dieser Arbeit, werden sowohl technische als auch medizinische Fragestellungen behandelt.

Im Teilbereich *Echtzeitanwendungen in der Telemedizin* (Teil I) werden folgende Forschungsfragen untersucht, wobei der Schwerpunkt auf der praktischen Umsetzung medizinischer Fragestellungen liegt.

1. *Welche Telemedizin-Szenarien werden von medizinischer Seite benötigt und akzeptiert?*

2. *Welche neuen technischen Innovationen werden benötigt, um das erweiterte Notfallszenario umzusetzen?*

3. *Welche Sicherheitsaspekte sind bei der Übertragung von personenbezogenen sensiblen Daten zu beachten?*

   Für die praktische Anwendung ist es unabdingbar, festzulegen, welche Szenarien tatsächlich im medizinischen Umfeld benötigt werden. Hierzu wurde eine ausführliche Untersuchung durchgeführt, wobei der technische Aspekt, speziell die Umsetzung in der Region Tübingen, Berücksichtigung gefunden hat. Hierfür wurden u.a. die möglichen Datenraten im mobilen Umfeld untersucht. Insbesondere die Befundbarkeit von ggf. bereits im Vorfeld komprimierten Daten stand im Mittelpunkt. Zudem wurde die Frage nach dem möglichen Transportprotokoll aufgeworfen, welches die notwendigen Eigenschaften aufweist, um die speziellen Anforderungen technisch und praktisch umzusetzen.

   Die konkrete Umsetzung, insbesondere unter Berücksichtigung der Fragestellung nach der Sicherheit, wurde in einem Gemeinschaftsprojekt mit der Firma CISCO erarbeitet.

4. *Welchen Einfluss hat die Technologie auf bestehende Arbeitsabläufe bzw. Workflows?*

   Aus medizinischer Sichtweise muss die Frage beantwortet werden, wie ein solches technisch orientiertes Szenario in den medizinischen Workflow eingebunden werden kann. Hierfür werden speziell auf das Notfallszenario abgestimmte Lösungen erarbeitet und in bestehende Systeme integriert.

Für den zweiten Teilbereich *Flexible und anpassbare Datenübertragung zur Realisierung von Telemedizin-Anwendungen* (Teil II) wurden folgende Fragestellungen vorgegeben, die auf die technische Realisierbarkeit spezieller Aspekte des erweiterten Notfallszenarios abstellen. Die Fragen orientieren sich am Transportprotokoll SCTP, welches sich aufgrund seiner flexiblen Ausrichtung für die Umsetzung des Szenarios als besonders geeignet herausgestellt hat.

1. *Können adaptive Verfahren und Methoden zur effektiven Pfadwahl mit SCTP genutzt werden?*

   Um diese Frage zu beantworten, wurden die von SCTP zur Verfügung gestellten Parameter dahingehend untersucht, ob und wie sie zur Auswertung der aktuellen Netzlast geeignet sind. Insbesondere galt es, herauszufinden, welche Parameter Rückschlüsse auf die aktuelle Güte einzelner Pfade bei paralleler Übertragung auf mehreren Pfaden in heterogenen Netzwerkstrukturen zulassen. Nachdem eine Erfolg versprechende Auswahl getroffen werden konnte, stand die Frage nach möglichen adaptiven Verfahren bzw. Klassifikationsverfahren im Mittelpunkt. Es

wurden verschiedene Klassifikationsverfahren theoretisch und praktisch herangezogen, wobei die Ergebnisse durch einen praktischen Testaufbau verifiziert wurden, sodass eine abschließende Bewertung der Leistungsfähigkeit der neuen Technik für die Pfadwahl erfolgen konnte. Bei der Auswertung der Ergebnisse wurde speziell auf verschiedene Netzcharakteristika eingegangen, um ein möglichst breites Spektrum an gegebenen Neztsituationen zu erfassen.

Im Teilabschnitt *Sicherheitsaspekte bei der Übertragung von Medizindaten* (Teil III) wurde speziell auf die Sicherheitsaspekte fokussiert und die praktische Realisierbarkeit der kryptographischen Behandlung von sensiblen medizinischen Informationen behandelt.

1. *Welche Penalties sind bei einem kryptographisch abgesicherten Datenversand zu berücksichtigen, insbesondere welchen Einfluss hat die Verschlüsselung auf den Datendurchsatz, und lassen sich diese Mankos durch adaptive Verschlüsselung egalisieren?*

Es hat sich während der praktischen Untersuchung der kryptographischen Absicherung von SCTP herausgestellt, dass mit *Secure-SCTP* eine besonders flexible und effektive Sicherheitsarchitektur besteht. Daher wurde in Abschnitt 11.1 eine ausführliche Beschreibung der Möglichkeiten von SCTP gegeben. Im folgenden Abschnitt 11.4 wurde kurz auf die existierenden Alternativen eingegangen. Basierend auf den Grundlagen konnte in Abschnitt 12 ein CTI (Chunk Throughput Index) erstellt werden, der als Entscheidungsgrundlage für adaptive Verschlüsselungssysteme verwendet werden kann.

1.2 Dokumentenstruktur

In Abbildung 1.2 ist die Struktur der Arbeit aus Gründen der Anschauung graphisch zusammengestellt.

Den Schwerpunkt der vorliegenden Arbeit bildet der technische Aspekt in Teil II, in dem ein neuartiges Konzept zur Pfadwahl im Multi-Pfad-Szenario erarbeitet und validiert wird. Nachdem in Abschnitt 5 die Problemstellung herausgearbeitet und bestehende Lösungsansätze beleuchtet wurden, werden in den folgenden zwei Abschnitten die Grundlagen für die folgende Ausarbeitung dargestellt.

Besonders wichtig erscheint die Zusammenstellung der Eigenschaften von SCTP sowie der praktische Ablauf einer Datenübertragung mittels SCTP, die in Abschnitt 6 ausführlich zusammengestellt sind. Der Ablauf sowie das Zusammenspiel der von SCTP bereitgestellten Parameter bilden die Grundlage für den Einsatz zur Pfadwahl und die Auswahl sowie für die Konfiguration der zugehörigen adaptiven Algorithmen. Es folgt eine Zusammenstellung möglicher adaptiver Verfahren in Abschnitt 7, die für die gesuchte Pfadwahl in die engere Auswahl gekommen sind.

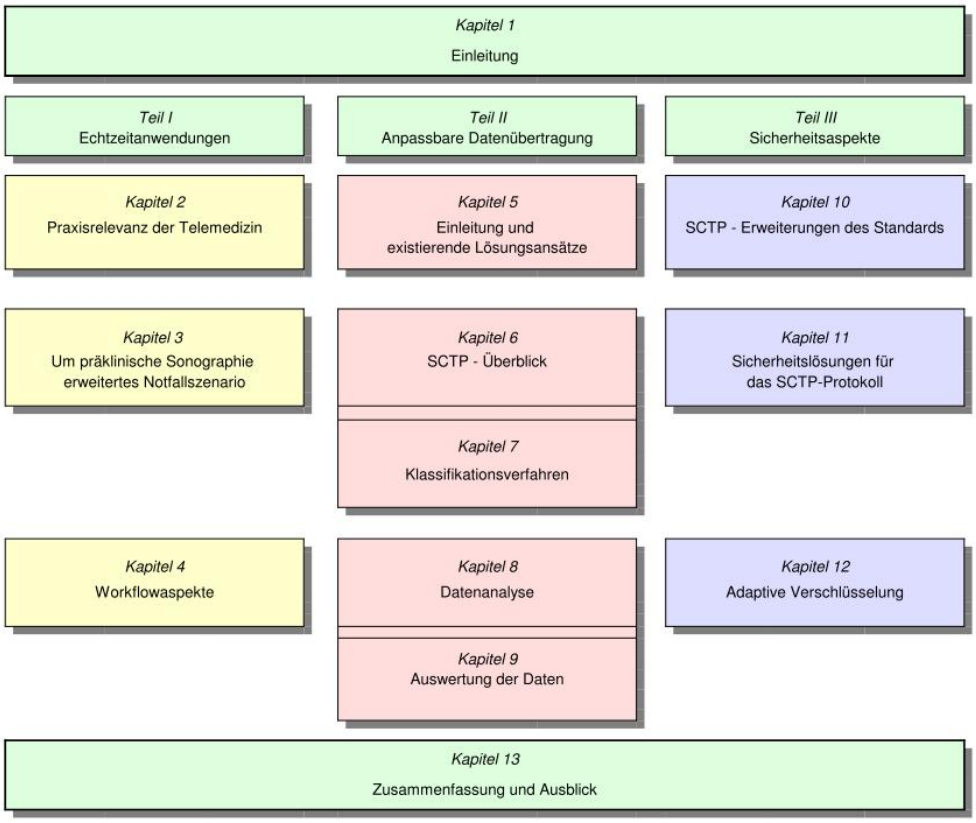


Abbildung 1.2: Der strukturelle Aufbau der Arbeit

Basierend auf diesen Grundlagen wird im Abschnitt 8 das neue Konzept zur Pfadwahl erst theoretisch erarbeitet und im Abschnitt 9 praktisch umgesetzt und evaluiert. In einer abschließenden Betrachtung werden die Ergebnisse der Testreihen zusammengefasst und ein abschließendes Urteil der Leistungsfähigkeit der adaptiven Pfadwahl gegeben. Die Auswertung der Ergebnisse hat gezeigt, dass sich mit dieser neuartigen Pfadwahlmethode neue Möglichkeiten für einen effektiven und effizienten Datentransfer unter Verwendung mehrerer Pfade eröffnet, sodass weitere wissenschaftliche Untersuchungen, basierend auf den hier vorgestellten Ergebnissen, sinnvoll erscheinen.

Der Teil I der vorliegenden Arbeit setzt auf den derzeit in der Medizin angewandten Szenarien auf, wobei der Schwerpunkt auf der Untersuchung bzw. Einbindung der Telematik in das Notfallszenario besteht. In Abschnitt 2 wird, basierend auf einer an der Universitätsklinik Tübingen durchgeführten Untersuchung, der Stand und die Notwendigkeit der Telemedizin erörtert. Als besonders interessant und technisch anspruchsvoll hat sich das erweiterte Notfallszenario herausgestellt, sodass der Abschnitt 3 die Möglichkeiten der mobilen Sonographie im Bereich der Telemedizin beleuchtet.

Ein wichtiger Aspekt ist die Integration von Informationstechnik in bestehende medizinische Abläufe. In Abschnitt 4 bestand daher die Notwendigkeit, in den bestehenden Workflow der Notfallmedizin einzugreifen und um den Bereich der präklinischen Sonographie zu erweitern. Der von dem Verfasser vorgestellte neue Workflow ermöglicht die Integration, ohne den Ablauf für das Rettungspersonal über Gebühr zu verändern.

Der abschließende Teil III der vorliegenden Arbeit fokussiert auf den Sichheitsaskpekt bei der Übertragung von nicht nur medizinischen sensiblen Daten. Die Sicherheit steht aber in direkter Konkurrenz zur Geschwindigkeit der Datenübertragung. Um einen Überblick über die derzeit für SCTP bestehenden Sicherheitslösungen zu erhalten, wurde in Kapitel 11 eine Übersicht über die bestehenden Sicherheitsarchitekturen gegeben. Zusätzlich wird in Kapitel 10 auf die Erweiterbarkeit von SCTP eingegangen, die einen entscheidenden Anteil an der Flexibilität von SCTP hat. Die Feststellung, dass es unter besonderen Umständen zu einer Verzögerung der Übertragung aufgrund mangelnder Host-Performance kommen kann, führt im abschließenden Kapitel 12 zur Definition eines Maßes zum Vergleich verschiedener Verschlüsselungsverfahren bzw. der zur Verfügung gestellten Schlüssellängen, sodass auch hier ein Einsatz von adaptiven Verfahren zur Auswahl des geeigneten Verfahrens möglich wird.

Teil I

Echtzeitanwendungen in der Telemedizin

Derzeit wird von den technischen Möglichkeiten, die die Telemedizin bietet, kaum Gebrauch gemacht. Das liegt zum einen an der nicht vorhandenen Infrastruktur, zum anderen aber auch an den unterschiedlichen, inkompatiblen Systemen, die eine herstellerübergreifende Bild- und Datenverarbeitung nicht zulassen. An Telemedizin-Anwendungen werden besonders hohe technische Anforderungen gestellt, um die Kundenwünsche adäquat umzusetzen. Hieraus ergibt sich das nächste Problem, nämlich die mangelnde Akzeptanz für die neuen Techniken beim medizinischen Personal, aber auch in der Bevölkerung, also dem *eigentlichen Kunden*, dem Patienten.

Kern der vorliegenden Dissertation ist die Erweiterung der technischen Möglichkeiten, um so die technische Infrastruktur für Anwendungen aus dem Bereich der Medizin-Telematik zu schaffen. Die Erweiterungen aus technischer Sicht werden schwerpunktmäßig im zweiten Teil der vorliegen Arbeit behandelt.

Der erste Teil betrachtet die Thematik aus medizinischer Sicht. Neben den grundsätzlichen Fragen, was unter Telemedizin überhaupt zu verstehen ist (vgl. Abschnitt 2.1) und welche Systeme bereits derzeit erfolgreich im Einsatz sind, werden auch die Telemedizin-Szenarien angesprochen, bei denen die technischen und wissenschaftlichen Voraussetzungen für eine vollständige praktische Umsetzung noch nicht gegeben sind. Hierbei haben sich solche Szenarien als besonders schwierig erwiesen, die ein größeres Aufkommen von Daten in *Echtzeit* verarbeiten müssen. Es wird sich zeigen, dass den *Echtzeitanwendungen* die Zukunft der Telemedizin gehört.

Nach einer kurzen Vorstellung der wissenschaftlich klassifizierten Medizin-Szenarien im Abschnitt 2.4 wird im Abschnitt 3.1 auf ein spezielles Szenario, nämlich auf das *erweiterte Notfallszenario* fokussiert. Dieses Szenario stellt hohe Anforderung an das Gesamtsystem, da neben Echtzeitfähigkeit auch die Ausfallsicherheit des Systems und die sichere Übertragung der personenbezogenen Patientendaten garantiert werden muss.

Ein Schwerpunkt bildet die in Abschnitt 3.2 behandelte *mobile Sonographie*, da durch den Einsatz der erst in den letzten Jahren zur Verfügung stehenden kleinen handlichen mobilen Geräte eine optimale Versorgung eines Verunfallten direkt am Ort des Geschehens erfolgen kann. Es gilt aber auch, Skepsis gegenüber der neuen Technik zu überwinden, sodass neben der technischen Realisierbarkeit auch immer wieder die Fragestellung, ob ein solches Szenario überhaupt sinnvoll und notwendig ist, in den Mittelpunkt rückt.

Es hat sich herausgestellt, dass neben den technischen Voraussetzungen, um ein Telemedizin-Szenario in die Fläche zu bringen, auch die *internen Abläufe* – Workflows – neu strukturiert bzw. angepasst werden müssen. Dies wurde zum Anlass genommen, im Abschnitt 4 die Erweiterung der bestehenden medizinischen Algorithmen um die präklinische Sonographie vorzunehmen. Auch greift der Einsatz von Informationstechnik in das Berufsbild des ausführenden Personals ein. Ein verwendetes technisches Gerät, wie beispielsweise ein mobiles Sonographiegerät, muss nicht nur korrekt bedient werden, sondern es müssen auch die Ergebnisse korrekt und vollständig interpretiert werden.

Dieser Teil der vorliegenden Arbeit schließt mit Fragen der grundsätzlichen technischen Realisierbarkeit. So werden in Abschnitt 3.4 Ergebnisse eines durchgeführten *Basisprojekts* vorgestellt. Für das Projekt wurde von verschiedenen Firmen technisches Equipment zur Verfügung gestellt, so dass die Verifikation der theoretischen Ansätze durch praktische Versuche unterstützt werden konnte. Der Abschnitt 3.3 schlägt den Bogen zum Netzwerk, speziell zum SCTP-Protokoll, welches in den folgenden Teilen der Arbeit als Grundlage zur Realisierung des gewählten Szenarios herangezogen wird.

2 Praxisrelevanz der Telemedizin

2.1 Telemedizin und Medizin-Telematik

Da die *Telemedizin* ein junges Fachgebiet darstellt, sind die Begriffe und Definitionen im wissenschaftlichen Umfeld noch nicht manifestiert. Dies hat zur Folge, dass viele Begriffe unterschiedlich definiert bzw. synonym verwendet werden. In diesem Abschnitt werden die wichtigen Begriffe *Telemedizin* und *Medizin-Telematik* eingeführt und so der Grundstein für die technische Betrachtung gelegt.

Der Begriff der *Telemedizin* wird nach [Horsch und Handels 2005] im deutschsprachigen Raum häufig als Synonym für das gesamte Fachgebiet angesehen, wobei die Gemeinsamkeit durch den Einsatz der Technik gegeben ist. Eine grundlegende Definition stammt von Field [Field 1996], der die Telemedizin wie folgt beschreibt:

Definition 2.1.1 (Telemedizin) *Unter* Telemedizin[1] *versteht man den Einsatz von Informations- und Kommunikationstechnik zur Unterstützung der Gesundheitsversorgung, wenn Nutzer und Anbieter räumlich getrennt sind.*

Neben dem Begriff der *Telemedizin* stößt man in der Literatur auf den Begriff der *Medizin-Telematik* oder allgemein der *Telematik im Gesundheitswesen*. Mit dem Begriff der Telematik wird auf den technischen Aspekt fokussiert und medizinische Anlagen, Bildverarbeitung, Signalverarbeitung, Kommunikationstechnik und vieles mehr in die Betrachtung einbezogen.

Der Begriff der *Telematik* ist selbst ein Kunstwort, welches sich nach [Krüger und Reschke 2002] aus den Begriffen Telekommunikation und Informatik zusammensetzt, wobei der Schwerpunkt auf dem Entwurf und der Realisierung der technischen Infrastruktur von Netzen, Netz-Diensten und der darauf aufbauenden Anwendungen liegt. Die *Medizin-Telematik* kann, wenn man diese Form der Beschreibung fortsetzt, als Schnittmenge von Telekommunikation, Informatik und dem Gesundheitswesen aufgefasst werden. Der Sprachgebrauch der vorliegenden Arbeit wird sich an dieser Definition orientieren, dennoch soll kurz die Sichtweise der WHO[2] vorgestellt werden. Die WHO hat sich als Sonderorganisation der Vereinten Nationen zur Aufgabe gemacht, den bestmöglichen Gesundheitszustand der Erdbevölkerung sicherzustellen.

[1] Telemedicine is the use of electronic information and communications technologies to provide and support health care when distance separates the participants.
[2] World Health Organization

Die WHO hat in ihrem Grundsatzpapier [WHO 97] eine eigene Definition von Medizin-Telematik vorgelegt, die in der Version von [Horsch und Handels 2005] eingeführt wird. Bei dieser Betrachtung werden neben den rein technischen Möglichkeiten auch die möglichen Anwendungsfelder der Medizin berücksichtigt.

Definition 2.1.2 (Medizin-Telematik) *Unter* Medizin-Telematik[3] *versteht man gesundheitsbezogene Aktivitäten, die unter Verwendung von Informations- und Kommunikationstechnologie zum Zweck globaler Gesundheitsförderung, Krankheitskontrolle und Krankenversorgung sowie für die Ausbildung, das Management und die Forschung auf dem Gebiet des Gesundheitswesens eingesetzt werden.*

Telemedizin – Strukturierung nach medizinischen Gesichtspunkten

Die Definition von Medizin-Telematik der WHO legt eine Strukturierung nach medizinischen Gesichtspunkten nah. Dabei lassen sich neben der *Patientenversorgung*, die in der Definition der WHO als Telemedizin betrachtet wird, weitere medizinische Anwendungsfelder bestimmen.

Die *Teleausbildung*[4] deckt den Bereich der Lehre ab. Die WHO versteht den Begriff sehr weitreichend, so ist zum einen die Aus- und Weiterbildung vom medizinischen Personal – vom Arzt bis zum Krankenpfleger – gemeint, zum anderen aber auch die Gesundheitsaufklärung der Bürger und Patienten.

Zudem wird die Telematik in der *medizinischen Forschung*[5] als eigenständiger Gesichtspunkt betrachtet. Auch hier wird der Begriff sehr allgemein verwendet, sodass sämtliche Forschung auf dem Gebiet der Medizin, die mit der Informationstechnik in Berührung kommt, unter diesem Punkt zusammengefasst wird. Ein besonderes Augenmerk wird auf die Durchführung von wissenschaftlichen Studien gelegt, die eine gezielte Kosten-Nutzen-Analyse von ausgewählten Telemedizin-Szenarien ermöglichen. Um den Einfluss von lokalen Parametern auf das Ergebnis der Studie gering zu halten, werden von der WHO multizentrische Studien bevorzugt, also Studien, die nicht nur in einem einzelnen, sondern in einer Vielzahl von Instituten bzw. Krankenhäusern durchgeführt werden. Soll die Studie länderübergreifende Resultate liefern, so wird zusätzlich die Auswahl von international verschiedenen Instituten gefordert.

Zudem wird das *Gesundheitsmanagement*[6] betrachtet. Auch dieser Gesichtspunkt wird von der WHO sehr weitreichend verstanden. So ist der Begriff des Managements nicht nur auf Planung, Überwachung und Überprüfung beschränkt, sondern umfasst auch die

[3] Definition nach WHO: Health telematics is a composite term for health-related activities, services and systems, carried out over a distance by means of information and communications technologies, for the pruposes of global health promotion, disease control and health care, as well as education, management, and research for health.
[4] tele-educatien for health
[5] telematics for health research
[6] telematics for health services management

Bewertung von Maßnahmen der Krankheitskontrolle und Gesundheitsversorgung sowie die Betrachtung von technischen und personellen Ressourcen, soweit sie durch moderne Kommunikationstechnik unterstützt wird.

Da diese Aufteilung sehr allgemeiner Natur ist, werden die einzelnen Teilbereiche in Form von medizinischen Anwendungsszenarien präzise spezifiziert, wobei jedes Szenario einer Grundstruktur nach der WHO zugeordnet werden kann. Die grundlegenden Szenarien sind in Abschnitt 2.4 zusammengestellt.

Telemedizin – Strukturierung nach technischen Gesichtspunkten

Jedes Anwendungsszenario stellt unterschiedliche Anforderungen an Mensch und Technik, was eine weitere Unterteilung nach technischen Gesichtspunkten nötig macht. Eine mögliche Strukturierung ist in [Wootton et al. 2006] gegeben, die eine Differenzierung nach den zwei Gesichtspunkten *Interaktion* und *Information* vorsieht.

Unter Information verstehen die Autoren die auszutauschenden bzw. über das Netz zu übertragenden Daten. Die in der Medizin üblichen Daten und Datenstrukturen werden in Abschnitt 3.4.3 thematisiert. Technisch gesehen erfolgt die Differenzierung zwischen bewegten Bildern, sprich: Videodaten, und einfachen Bilddaten. Betrachtet man moderne Szenarien, so kommen zusätzlich Text- und Sprachinformationen hinzu, die ebenfalls in Form von IP-Paketen über das Netz übertragen werden. Hier stellt sich die Frage, wie groß die zu übertragenden Daten sind – hierzu wurden im Vorfeld eigene Untersuchungen durchgeführt, die in Abschnitt 3.4.3 behandelt werden. Zudem muss geprüft werden, ob die derzeit existierende technische Infrastruktur darauf ausgelegt ist, diese Daten zeitnah zu transportieren.

Damit rückt der zweite Gliederungspunkt *Interaktion* ins Blickfeld, nämlich wie viel Zeit für die Übertragung zur Verfügung steht. Bei Umsetzung bestehender Abläufe spielt zumeist die Übertragungszeit eine untergeordnete Rolle. So ist es bei der Überbringung eines Arztbriefs (vgl. Beispiel 2.2.1) nicht kritisch, wenn der Brief mit einer größeren Verzögerung eintrifft, sodass beispielsweise der Versand per E-Mail in einem entsprechenden Telemedizin-Workflow möglich wäre. Ein solches Verfahren wird häufig auch als *Prerecorded* bezeichnet, da die Daten erst aufgezeichnet und im Anschluss als Ganzes gesendet werden.

Demgegenüber stehen Anwendungen, die die Übertragung der Daten in *Echtzeit* (engl.: realtime) voraussetzen. Werden beispielsweise bei einem kritischen Notfall, bei dem jede verstrichene Sekunde über Leben und Tod des Patienten entscheiden kann, die Informationen an ein Expertenteam zur Ferndiagnose gesendet, so darf es bei der Übertragung zu keinen Verzögerungen geschweige denn Ausfällen kommen. Die Daten können nicht erst aufgezeichnet werden, sondern müssen in Form eines Datenstroms (Streaming) übertragen werden. Echtzeitanwendungen stellen hohe Anforderungen an die verwende-

te Technik. So muss die Technik sicherstellen, dass auch bei geringen Bandbreiten die Übertragungsqualität ausreichend ist, um die medizinische Auswertung zu garantieren. Auch muss die Ausfallsicherheit garantiert werden, dies kann beispielsweise durch die Verwendung von Ausweichpfaden erreicht werden. Sollen hochauflösende Videodaten, die z.B. auf Basis von mobiler Sonographie erzeugt wurden, die im Abschnitt 3.2 im Rahmen des erweiterten Notfallszenarios eingeführt wird, in Form von Mobilkommunikation übertragen werden, stößt man schnell an die Grenzen des Machbaren. Zudem gibt es die zusätzliche Forderung der *sicheren Übertragung*, die speziell im dritten Teil der vorliegenden Arbeit thematisiert wird, die zusätzliche Anforderungen an die Echtzeitfähigkeit der Technik stellt.

Es kommt bei Echtzeitanwendungen zu einem *Zielkonflikt* zwischen *Echtzeit*, *Dienstgüte* und *Sicherheit*. Die Auflösung dieses Konflikts kann als Kernthema der vorliegenden Arbeit betrachtet werden. Von rechtlichen, organisatorischen und medizinischen Problemen sowie der ggf. bestehenden mangelnden Akzeptanz abgesehen, ist der aktuelle Stand der Technik nicht ausreichend, um sämtliche theoretisch konzipierten und als gewinnbringend und wichtig betrachteten Telemedizin-Szenarien auch in die Praxis umzusetzen. Mit dem in der vorliegenden Arbeit vorgestellten Konzept des *Intelligenten Netzes* wird eine Brücke zwischen den bestehenden technischen Möglichkeiten und den Anforderungen von Echtzeitanwendungen im medizinischen Umfeld gebaut, die es ermöglicht, auch unter schwierigen Bedingungen die Handlungsfähigkeit des medizinischen Personals sicherzustellen.

Jedes einzelne Szenario wird in Abschnitt 2.4 hinsichtlich *Interaktion* und *Information* betrachtet und bewertet.

2.2 Akzeptanz und Verbreitung von Telemedizin

Zwar hat die Telemedizin durch den rasanten Fortschritt der Computer- und Netztechnik gerade in den späten Achtziger, frühen Neunziger Jahren profitiert, doch wurden die hohen Erwartungen, die zu dieser Zeit an die Umsetzung in der Medizin gestellt wurden, nicht erfüllt.

Betrachtet man lediglich Szenarien, die nicht die hohen Anforderungen der Echtzeit bedingen, also mit Daten operieren, die im Vorfeld vollständig generiert und aufgezeichnet wurden, so kann das Argument der fehlenden Technologie nicht gestützt werden. So steht im Internet mit modernem digitalem Teilnehmeranschluss (DSL)[7] ein Übertragungsmedium, welches Übertragungsraten von bis zu 210 Mbit/s realisiert, ein ausgereiftes System zur Verfügung. Ein klassischer Filetransfer kann auch für große Datenmengen in akzeptabler Zeit durchgeführt werden. Zudem ist durch die hohe Verfügbarkeit von Internetdiensten, wie beispielsweise der *E-Mail*,[8] ein einfach zu installierendes, zu wartendes

[7] Digital Subscriber Line
[8] electronic mail – elektronische Post

und insbesondere einfach zu bedienendes Instrumentarium gegeben, welches einfach in den medizinischen Workflow integriert werden kann.

Somit deutet die mangelnde Akzeptanz auf organisatorische bzw. personelle Probleme hin. In [Wootton et al. 2006] wird auf die fehlenden gesicherten Informationen zur Kosteneffizienz der Telemedizin verwiesen. Dieses Argument kann nur noch bedingt anerkannt werden, da gerade in den letzten Jahren verstärkt Multizenter-Studien von anerkannten Instituten durchgeführt wurden, die die Notwendigkeit der Telemedizin deutlich hervorheben. Bevor auf Erkenntnisse einiger bekannter Studien eingegangen wird, soll anhand eines einfachen, jedem, der bereits als Patient mit Ärzten in Berührung gekommen ist, bekannten Szenarios die aktuelle Situation im Gesundheitssystem dargestellt werden.

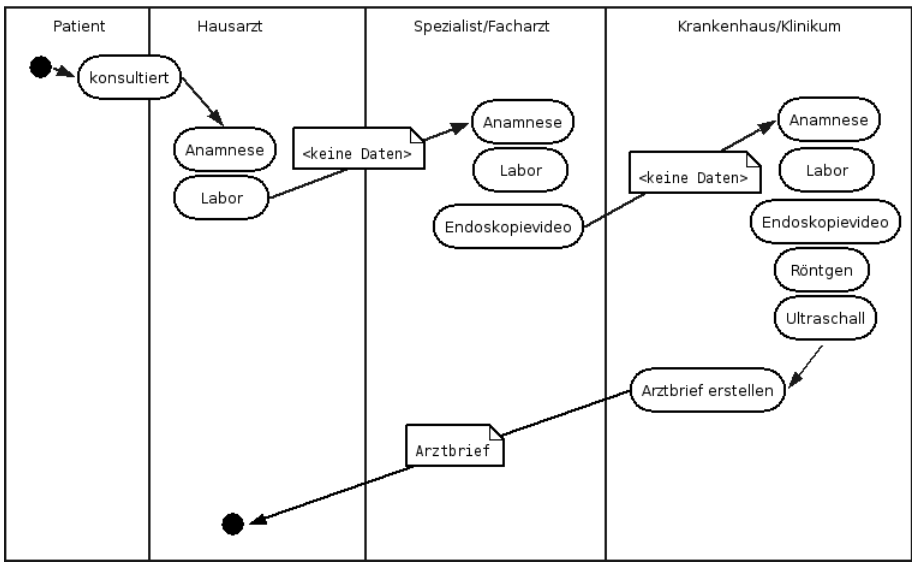


Abbildung 2.1: Beispiel für ungünstige Datenkommunikation, wie sie heute noch an der Tagesordnung ist.

Beispiel 2.2.1 (Prerecorded Medizin-Szenario – Arztbrief) *In der Arbeit von Dr. Pflügmeyer (vgl. [Pfllügmeyer 2001]) wurde dieser spezielle Ablauf analysiert, der hier kurz wiedergegeben wird. Der Datenfluss des heute noch üblichen Workflows ist in Abbildung 2.1 dargestellt. Der Patient konsultiert seinen Hausarzt, der die ersten Untersuchungen durchführt. Nachdem sein Part der Untersuchung abgeschlossen ist, wird der Patient an einen Facharzt überwiesen, der die erforderlichen weitergehenden Untersuchungen durchführt. Wird zusätzlich der Bedarf an Röntgen- und Ultraschalluntersuchungen festgestellt, erfolgt wiederum eine Überweisung, diesmal an das zuständige Klinikum. Hier werden die Untersuchungen komplett neu durchgeführt und ein* Arztbrief

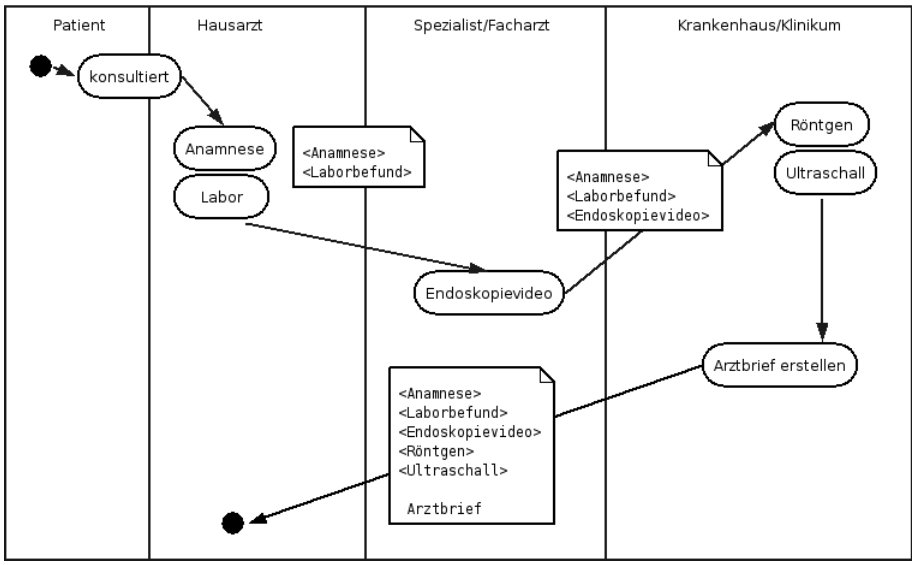


Abbildung 2.2: Verbesserter Datenfluss durch den Einsatz von elektronischem Datenaustausch

erstellt, der dem Hausarzt zugeleitet wird.

Die eigentlichen Informationen werden an jedem Punkt der Untersuchungskette durch wiederholte Untersuchungen gewonnen. Ein Datenaustausch zwischen den Instanzen findet nicht statt. Das Ergebnis wird in Form des Arztbriefs dem behandelnden Hausarzt zur Verfügung gestellt. Dies geschieht in den meisten Fällen in Form des klassischen Papierbriefs. Eine Übertragung über Datennetze ist in den seltensten Fällen gegeben.

Unter Verwendung von guter *Datenkommunikation kann der Aufwand und damit die Kosten für die Untersuchung drastisch gesenkt werden. Ein möglicher Ablauf ist in Abbildung 2.2 grafisch aufgearbeitet. Man erkennt, dass die folgenden Instanzen, der Facharzt, aber auch die Klinik, auf die bereits vom behandelnden Arzt gewonnenen Informationen zurückgreifen können. Die redundanten Untersuchungen können ausbleiben. Dies setzt den elektronischen Datenaustausch der jeweiligen Befunddaten voraus. Auch die Übergabe des Arztbriefs selber kann elektronisch erfolgen. Damit ist nicht nur eine Kostenreduzierung verbunden, sondern auch ein erheblicher Zeitvorteil zu erkennen.*

Die Gründe für die mangelnde Akzeptanz der Telemedizin sind vielfältig. Im Beispiel 2.2.1 scheitert die Umsetzung häufig an der Erstbeschaffung der notwendigen technischen Geräte, da kleine Arztpraxen die Kosten hierfür nicht tragen können oder wollen. Hinzu kommt, dass oft der Nutzen der Telemedizin nicht gesehen wird. Auch fehlen finanzielle Anreize für die Beteiligten.

In den Veröffentlichungen [Lacroix et al. 2002], [Jähn 2004] und [Wootton et al. 2006] wird ausführlich auf jedes einzelne Problem eingegangen. An dieser Stelle werden lediglich die wichtigsten kurz zusammengestellt. Ein Hauptproblem ist sicher die fehlende Eingliederung der telemedizinischen Leistungen in die Gebührenordnung. Zudem sind in vielen Fällen die rechtlichen Fragen nicht ausreichend geklärt, insbesondere Aufklärungspflichten und Arzthaftung bedingen gesetzliche Regelungen. Weiterhin wird – gerade auch von den Patienten – Datenmissbrauch und eine Entfremdung in der Patienten-Arzt-Beziehung befürchtet. Hier bedarf es zum einen Regelungen, die das Vertrauen in die Telemedizin herstellen, zum anderen aber Aufklärung über den Nutzen, der sich nicht nur in den Kosten niederschlägt, sondern sich auch durch eine verbesserte medizinische Versorgung ausdrückt.

2.3 Praxisrelevanz – Perspektiven der Telemedizin

Obwohl die Telemedizin sich im Vergleich zur Verwendung von IT in anderen Branchen noch in den Kinderschuhen befindet, sind die Perspektiven als sehr positiv zu beschreiben, was im Folgenden durch die Bewertung einiger exemplarischer Studien auf diesem Gebiet belegt werden soll.

So wurde bereits 1998 eine Studie von Berger & Partner im Auftrag des Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie und des Bundesministeriums für Gesundheit durchgeführt. Ziel der Studie mit dem Thema *Telematik im Gesundheitswesen* war die Ausarbeitung der *Perspektiven für die Telemedizin in Deutschland* [Berger 1998].

Als weitere Studie soll die speziell für die Teleradiologie entwickelte *ANARAD-Studie* einbezogen werden. Die ANARAD-Studie hat zum Ziel, eine möglichst breite und repräsentative Meinung von Radiologen zu verschiedenen Bereichen der Telemedizin zu erhalten. Deshalb wurden Fragebögen an die meisten deutschen Radiologen in Krankenhäusern und Praxen verschickt. Die Ergebnisse können aufgrund der hohen Beteiligung als repräsentativ angesehen werden. Eine Auswertung der Fragebögen wurde u.a. in dem Bericht [Horsch] vorgenommen, aus dem die zentralen Ergebnisse entnommen wurden.

Während die Anarad-Studie auf die Bedürfnisse und Probleme der Ärzte eingeht, fokussiert Berger auf den nationalen Gesamtkontext, der sämtliche Teilnehmer am Gesundheitssystem einbezieht.

Medizin-Telematik ganzheitlich betrachtet – Die Gesundheitsplattform

Nach Berger wird die Telemedizin die zukünftige Arbeitsumgebung eines Arztes entscheidend verändern. Neben Kostengründen werden auch medizinische Argumente für

den weiteren Ausbau der Telemedizin angeführt. Die Zukunft der Telemedizin beschreibt Berger mit folgenden euphorischen Worten:

> Die Lösung wesentlicher Probleme des modernen Gesundheitswesens – u.a. explodierende Informationsmengen, Qualitätsverbesserung, Kostendämpfung – wird durch den Einsatz moderner Informationstechnik erheblich erleichtert. *Telemedizin* wird nicht nur zur Lösung von Transparenzproblemen einen Beitrag leisten, sie wird auch die bestehenden Koordinierungs-, Integrations- und Vernetzungsprobleme minimieren und die Entscheidungs- und Planungsgrundlagen auf allen Ebenen verbessern.

Damit die Telemedizin diesen Anspruch erfüllen kann, fordert Berger eine ganzheitliche Betrachtung der Telemedizin und stellt fest, dass die derzeit gängige Praxis der Telemedizin auf Ebene einzelner Projekte nicht ausreichend sei. Der Vorteil der Ganzheitlichkeit kann mit den folgenden Worten der Berger-Studie beschrieben werden:

> Telematik, ganzheitlich betrachtet, zeigt auf, dass moderne Informations- und Kommunikationstechnologie sehr wohl geeignet ist, Lösungsansätze für Transparenz, Integration und Vernetzung zu bieten. Strebt man eine weitere Entwicklung der medizinischen Leistungsfähigkeit unter gleichzeitiger Kostenbegrenzung an, so ist dies kaum ohne den Einsatz telematischer Lösungen erreichbar.

Der Vorschlag, der sich aus der Berger-Studie abgeleitet, ist der Aufbau einer *Gesundheitsplattform*, die als einheitliches System die als Grundlage für zukünftige Telematikprojekte in der Medizin angesehen wird. Was wird in [Berger 1998] unter der Gesundheitsplattform verstanden?

Definition 2.3.1 (Gesundheitsplattform) *Unter der* Gesundheitsplattform *ist ein Informations- und Kommunikationssystem zur effizienten Bearbeitung zentraler Aufgaben des Gesundheitswesens zu verstehen, das die Teilnehmer am Gesundheitssystem miteinander verbindet bzw. vernetzt.*

Die Teilnehmer am Gesundheitssystem kann man anhand ihrer Aufgaben und Funktion im System in vier Gruppen einteilen. Angefangen bei den *Patienten* als Ausgangspunkt jeder medizinischen Versorgung. Demnach ist eine weitere Gruppe die der *Leistungserbringer*, also alle diejenigen, die direkt für die Versorgung des Patienten zuständig sind. Dies sind im Wesentlichen die Krankenhäuser, Arztpraxen, Labore, Apotheken sowie Pflege- und Notdienste. Neben diesen offensichtlichen Teilnehmern gibt es auch noch die *Gesundheitsverwaltung*, also die Krankenkassen, Rentenversicherungen, Ärztekammern, Ministerien etc., die sich hauptsächlich mit der Abrechnung der Leistungen und den rechtlichen Rahmenbedingungen der Medizin beschäftigen. Um den Ärzten die notwendige Infrastruktur und das medizinische Gerät bereitstellen zu können, bedarf es noch der Gruppe der *Anbieter*.

Die Gesundheitsplattform stellt den Austausch von Informationen der einzelnen Teilnehmer sicher, damit die Funktionen der einzelnen Teilnehmer, wie Versorgung, Qualitätssicherung, Abrechnung- und Erstattung, Planungs- und Entscheidungsfindung, Fort- und Weiterbildung etc., bestmöglich und IT-gestützt durchgeführt werden können.

Die Berger-Studie wurde bereits 1998 in Auftrag gegeben, sodass sich die Frage stellt, inwieweit die damaligen Visionen und Ideen bis zum jetzigen Zeitpunkt, also zehn Jahre später, verwirklicht werden konnten. Die Euphorie, die in der Studie heraufbeschworen wurde, konnte nicht in die Bevölkerung transportiert werden, sodass immer wieder Kritik an der Vernetzung der einzelnen Systeme aufkommt. Die Gefahr des *gläsernen Patienten* wird verstärkt von den betroffenen Patienten wahrgenommen. Insbesondere da das kritische Thema der *elektronischen Gesundheitskarte* in den Vordergrund gestellt wurde, die weniger auf die bessere bzw. schnellere Versorgung gerade in Notfällen abzielt, sondern vielfach als reines Instrumentarium zur Kostensenkung betrachtet wird. Ein aktueller Artikel, der am 10.09.2008 in der *Ärzte Zeitung* erschienen ist, (vgl. [Ärzte Zeitung 2008]) zeigt auf, dass es noch ein weiter Weg zur Vision der Gesundheitsplattform im Sinne der Berger-Studie ist.

Unabhängig von den Problemen bei der Gesundheitskarte sind auf anderen Gebieten der Medizin bereits Fortschritte in Richtung Gesundheitsplattform erzielt worden. In [Schug and Redders 2005] wird der Entwicklungsprozess seit 1999 betrachtet und Resümee gezogen. Die Betrachtung erfolgt aus Sicht der Bundesländer, denen aufgrund der föderalen Struktur der Bundesrepublik ein maßgeblicher Anteil an der Erstellung und Realisierung von Telemedizin-Projekten zufällt.

Grundsätzliche Meilensteine zur Gesundheitsplattform werden kurz zusammengestellt. Die Europäische Kommission hat mit dem Aktionsplan *eEurope* (vgl. [eeurope 2002]) die politische Initiative ergriffen, um die Informationsgesellschaft für *alle* zugänglich zu machen. Die Kommission versteht unter „alle“ alle Personen, Haushalte, Schulen, Unternehmen und Behörden. Der Aktionsplan bezieht sich dabei auf sämtliche Einsatzmöglichkeiten des Internets und der IT-Infrastruktur für die Gesellschaft. Ein Schwerpunktthema nimmt dabei die Telemedizin ein. Unter dem Stichwort *Gesundheitsfürsorge über das Netz* werden die Grundbegriffe der Medizinplattform, wie sie bereits von Berger und Partner in [Berger 1998] formuliert wurden, aufgegriffen. So wird von einem integrierten Gesundheitssystem gesprochen, das die Teilnehmer am Gesundheitssystem miteinander verbindet. Konkret wird die Anbindung von kleineren Krankenhäusern in ländlichen Gegenden angesprochen, aber auch auf die Gesundheitskarte und Informationsdienste wird bereits eingegangen. Ein solcher politische Aufbruch auf europäischer Ebene bleibt nicht ohne Einfluss auf die Politik der Mitgliedstaaten.

Die politische Diskussion der folgenden Jahre führt letztendlich zum *Gesetz zur Modernisierung der gesetzlichen Krankenversicherung*, kurz GKV-Modernisierungsgesetz – GMG, welches im Bundesgesetzblatt [GMG 2003] novelliert wurde und am 01 Januar 2004 in Kraft getreten ist. Das Gesetz legt den rechtlichen Grundstein für die Gesund-

heitsplattform, da erstmals wichtige Grundprinzipien und technische Grundsatzfragen festgelegt wurden. So wurden die Paragraphen §67 (Elektronische Kommunikation) und §68 (Finanzierung einer persönlichen elektronischen Gesundheitsakte) eingefügt.

Insbesondere §67 ist von Interesse, daher wird der Absatz (1) zitiert:

§67 *Zur Verbesserung der Qualität und Wirtschaftlichkeit der Versorgung soll die papiergebundene Kommunikation unter den Leistungserbringern so bald und so umfassend wie möglich durch die elektronische und maschinell verwertbare Übermittlung von Befunden, Diagnosen, Therapieempfehlungen und Behandlungsberichten, die sich auch für eine einrichtungsübergreifende fallbezogene Zusammenarbeit eignet, ersetzt werden.*

Somit wird nicht nur die Möglichkeit zum Einsatz der Medizin-Telematik gegeben, sondern sogar eine gesetzliche Verpflichtung verbunden.

Medizin-Telematik für teleradiologische Anwendungen

Die ANARAD-Studie hat einen mehr praktischen Ansatz. Es geht weniger um die Beschreibung der Gesamtsituation, sondern vielmehr um die Einschätzung derjenigen, die von dem Einsatz der neuen Technologie betroffen sind. Dies sind vor allem die Ärzte, die nicht nur die technischen Geräte bedienen müssen, sondern für die sich auch die Arbeitsabläufe und Vorgehensweisen ändern.

Ein für den weiteren Verlauf dieser Arbeit wichtiger Aspekt ist die Frage nach den Medizin-Telematik-Szenarien, die von den befragten Ärzten als besonders wichtig angesehen werden, insbesondere da im Abschnitt 3.1 auf ein spezielles Szenario eingegangen werden soll, das sich nicht nur von der praktischen, sondern auch von der wissenschaftlichen Seite als interessant herauskristallisiert hat.

Auf den Fragenkomplex – *Bedeutung von und Bedarf an teleradiologischen Anwendungen* – wurden zwei Telemedizin-Szenarien als besonders wichtig herausgestellt. In Abbildung 2.3 sind die Ergebnisse der Studie für die Frage nach relevanten Szenarien grafisch aufbereitet. Es handelt sich um einen Auszug aus der umfassenden Bewertung der Studie von Alexander Horsch [Horsch]. Aus allen Antworten wurden die statistischen Größen Median und oberes bzw. unteres Quartil bestimmt. Eine kurze Zusammenfassung der Bedeutung der mathematischen Begriffe kann in Abschnitt 8.5.1 nachgelesen werden. In der Abbildung steht das Quadrat für den jeweiligen Median der Antworten. Die entsprechenden Quantile werden entsprechend durch einen gefüllten Kreis repräsentiert.

Sehr wichtig erscheinen anhand der statistischen Daten die *Notfallkonsultation* und die *Expertenkonsultation* in schweren Fällen. Beide Szenarien wurden in der vom Verfasser durchgeführten internen Untersuchung der Praxisrelevanz von Telemedizin-Szenarien (vgl. [Kamphenkel]) genauer betrachtet, und es erscheint angebracht, bei einer praktischen Umsetzung diese Szenarien als vorrangig zu betrachten.

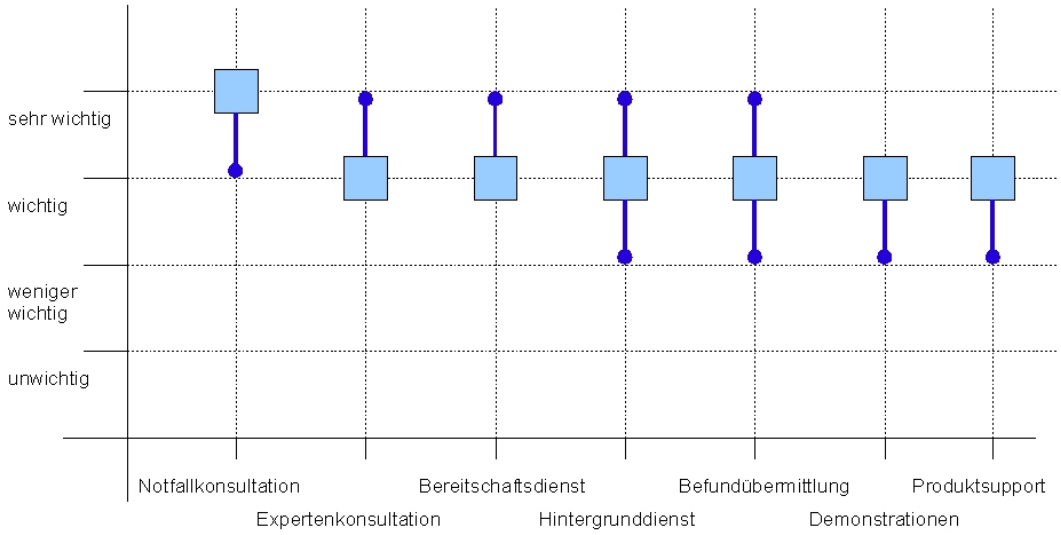


Abbildung 2.3: Bedeutung teleradiologischer Anwendungen – Die sieben Anwendungen mit den höchsten Bewertungen

Die Studie hat einen weiteren interessanten Aspekt herausgearbeitet, nämlich die Frage nach den Gefahren der Telemedizin. Die Auswertung der Antworten auf die Frage: „Welche Gefahren werden in der Teleradiologie gesehen?“ ist in Abbildung 2.4 zusammengestellt. In der Umfrage wurde von den befragten Radiologen der ungenügenden Datensicherheit viel Gefahrenpotenzial beigemessen. Daraus lässt sich die zentrale Bedeutung der Sicherheitsarchitektur erfassen, die im Zuge der Einführung einer IT-Infrastruktur eine herausgehobene Stellung einnehmen muss. Im dritten Teil dieser Arbeit wird speziell auf den Sicherheitsaspekt in der Telemedizin im Allgemeinen und in der Datenübertragung im Speziellen noch ausführlich eingegangen, sodass hier lediglich die Legitimation für die späteren Ausführungen hergeleitet werden soll.

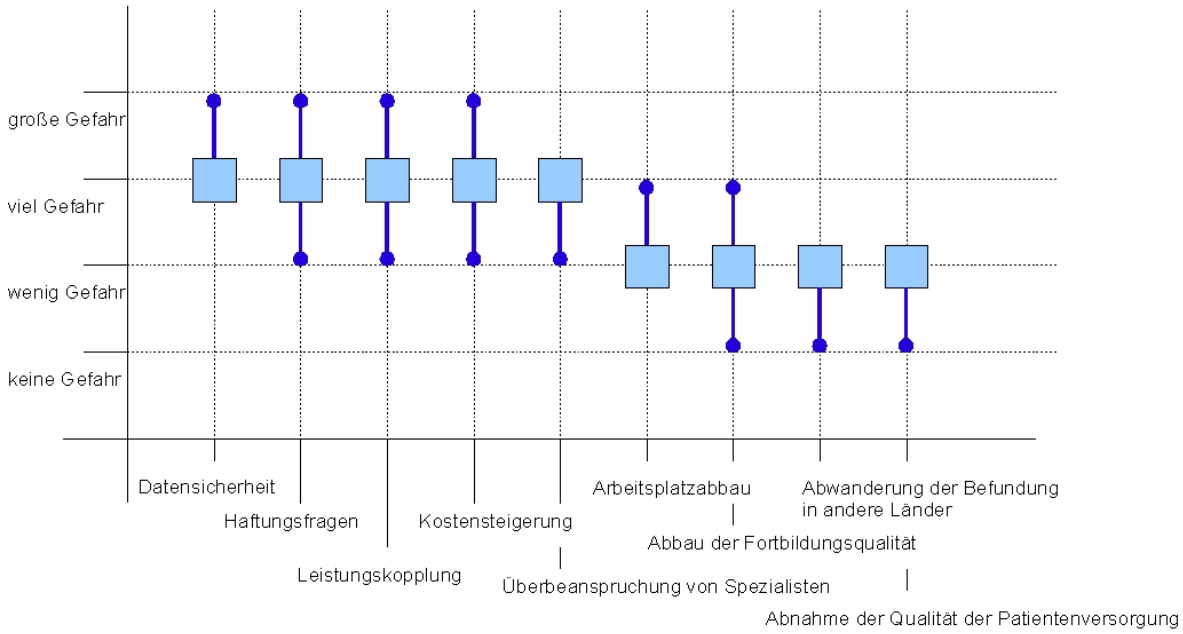


Abbildung 2.4: Welche Gefahren werden in der Verwendung von Teleradiologie gesehen? – Auswertung der ANARAD-Studie

2.4 Telemedizin-Szenarien – Ein Überblick

Für die Zusammenarbeit mit dem Klinikum Tübingen wurde ein spezielles Telemedizin-Szenario gesucht, das zum einen von großer medizinischer Bedeutung ist, zum anderen aber technisch und organisatorisch noch nicht abschließend behandelt wurde, sodass mit der Arbeit ein Beitrag zur Umsetzung in die Praxis gegeben werden kann. Neben Literaturrecherche wurde eine Befragung in bestimmten Abteilungen der Klinik durchgeführt, um so den Bedarf und die Wünsche des medizinischen Personals in Erfahrung zu bringen. Aufgrund dieser Voraussetzungen werden die klassischen Telemedizin-Szenarien kurz vorgestellt und auf Anwendbarkeit geprüft

In [Wootton et al. 2006] wird zwischen *allgemeinen telemedizinischen Diensten* und *Telemedizin-Szenarien* unterschieden, die in den beiden folgenden Teilabschnitten be-

trachtet werden.

2.4.1 Allgemeine telemedizinische Dienste

Unter *allgemeinen telemedizinischen Diensten* werden Anwendungen zusammengefasst, die den Prerecorded-Verfahren im Sinne von Abschnitt 2.1 zuzurechnen sind. Exemplarisch sei die *Elektronische Patientenakte*, das *elektronische Rezept*, der *elektronische Arztbrief* und die bereits erwähnte und in der öffentlichen Kritik stehende *Gesundheitskarte* angeführt. Die elektronische Patientenakte und das elektronische Rezept können als Schlüsseldienste verstanden werden, daher wird ihre Funktion kurz erläutert.

Elektronische Patientenakte

Die *elektronische Patientenakte* nimmt eine besondere Stellung unter den o.a. Diensten ein, da sie als Basisbaustein anderer Szenarien betrachtet werden kann. In [Märkle 2002] beschäftigen sich die Autoren mit der Standardisierung, geben aber auch eine sehr prägnante Definition des Begriffs:

Definition 2.4.1 (elektronische Patientenakte) *Unter der* Elektronische Patientenakte (engl.: electronical medical record) *(EPA) versteht man die Menge aller gesundheitsbezogenen Daten und Informationen über den einzelnen Patienten, die in elektronischer Form gespeichert sind.*

In der klassischen Terminologie wird häufig die Patientenakte rein als Sammlung von Daten verstanden, die für die Abrechnung benötigt werden. Der hier verwendete Begriff geht deutlich weiter, da hier sämtliche vorhandenen Daten eingeschlossen sind. Somit werden auch bereits Voruntersuchungen und Laboruntersuchungen in der Patientenakte dokumentiert. Hierbei ist man aufgrund der digitalen Speicherung der Daten nicht auf reine Textdaten beschränkt, sondern die moderne Patientenakte kann auch Bilder, Filme und sogar Audiokommentare enthalten. Man ist demnach in der Lage, die gesamte Krankheitshistorie inklusive der Begleiterkrankungen, Therapien und den Therapieverlauf vollständig und lückenlos zu erfassen.

Eine so geführte Akte wirft technische Fragen hinsichtlich der Realisierung, aber auch der sicherheitstechnischen Absicherung der kritischen personenbezogenen Daten auf. Hinzu kommen Fragen der Ablauforganisation, wenn die entsprechenden Daten von einer Institution an eine andere weitergegeben werden sollen.

Da diese Informationen über einen Patienten elementar für eine mögliche Behandlung sind, wird die EPA häufig auch bei Telemedizin-Szenarien mit Echtzeitcharakter benötigt. Handelt es sich um ein Szenario, das nicht nur aus technischer, sondern auch aus medizinischer Sicht zeitkritisch ist, sei es ein Notfall oder die Durchführung einer lebenserhaltenden Operation, so liefert die EPA zwar wichtige Informationen, die aber nicht im vollen Umfang zeitnah übertragen werden können. Die EPA ist darauf ausgelegt, Patientendaten über Jahrzehnte hinweg zu sammeln, dies führt zu einer fast unüberschaubaren

Datenmenge, sodass im Falle von zeitkritischen Datenübertragungen eine Selektion der wichtigen, sprich: für die spezielle Situation angepasste, Daten erfolgen muss.

Elektronische Rezept

Neben der EPA wird in [Wootton et al. 2006] das *elektronische Rezept* unter den allgemeinen Diensten geführt.

Definition 2.4.2 (elektronisches Rezept) *Unter einem* elektronischen Rezept (engl.: electronic prescription) *versteht man ein elektronisches Dokument, das Informationen über die Verordnung von Arzneimitteln enthält.*

Nach [Paland and Riepe 2005] ist das elektronische Rezept einer der Dienste, die sich nach kurzer Zeit nach der Einführung wirtschaftlich selber tragen können. Bei der Umsetzung sind viele meist kostspielige Medienbrüche zu überwinden. Derzeit werden nach Paland etwa 750 Millionen Rezepte pro Jahr ausgestellt, die im Schnitt 5 mal gesondert bearbeitet werden müssen. Neben der Kosteneinsparung wird die zuverlässige Arzneimittelausgabe an den Patienten als Vorteil gesehen.

Für die technische Realisierung des elektronischen Rezepts existieren zwei unterschiedliche Konzepte, die beide als Schlüsseltechnologie angesehen werden. Eine mögliche Realisierung bindet das elektronische Rezept als Anwendung an die elektronische Gesundheitskarte. Bei diesem Vorgehen würde der Arzt das Rezept in elektronischer Form auf der Gesundheitskarte des Patienten speichern, die dann vom Patienten persönlich zum Apotheker gebracht wird, der diese dann ausliest und die gewünschten Medikamente aushändigt. Ein solches Szenario lässt sich mit wenig organisatorischem und finanziellem Aufwand realisieren, sodass aufgrund des hohen Kosteneinsparpotenzials dieser Lösung sich die Gesundheitskarte ohne große Widerstände etablieren lässt.

In [Berger 1998] wird eine Serverlösung für das elektronische Rezept als zweite technische Form der Realisierung vorgestellt. Berger versteht das elektronische Rezept als „Schuhlöffelfunktion“[9], man kann auch sagen als Hintertür, um eine generelle Infrastruktur und Kommunikationsplattform zu etablieren. Auch hier wird Bezug auf das hohe Kosteneinsparpotenzial genommen. Bei diesem Verfahren würde der Arzt das Rezept direkt über eine Datenleitung an die Apotheke versenden. Berger argumentiert, dass ein solches Szenario bereits eine einheitliche IT-Infrastruktur voraussetzt, die dann für weitere Telemedizindienste genutzt werden kann.

Unabhängig, welche Lösung sich in der Praxis durchsetzt, nimmt das elektronische Rezept eine Schlüsselposition für den weiteren Aufbau der Medizin-Telematik ein. Da es sich als schwierig herausgestellt hat, die nach Berger geforderte Infrastruktur herzustellen (vgl. auch das Arztbrief-Beispiel 2.2.1 und die darauf basierenden Folgerungen), ist

[9] Schuhlöffelfunktion – Originalwortlaut Berger

nach aktuellem Stand die Einführung des elektronischen Rezepts auf Basis der Gesundheitskarte am wahrscheinlichsten.

2.4.2 Echtzeitszenarien der Medizin-Telematik

Die in diesem Abschnitt beschriebenen Szenarien haben eins gemeinsam, nämlich die Echtzeitdarstellung von medizinischen Bild- und Videodaten. Unterschieden wird lediglich in der „Dringlichkeit“ der Echtzeitübertragung. Bevor auf die Unterschiede eingegangen wird, werden die einzelnen Szenarien kurz vorgestellt.

Telekonsultation und Teleradiologie

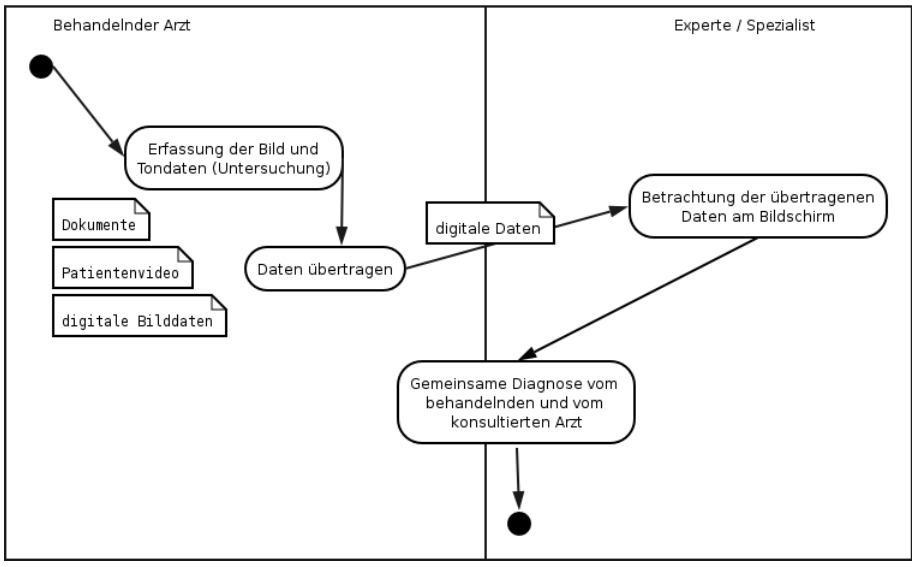


Abbildung 2.5: Arbeitsablauf der Telekonsultation

In Abbildung 2.5 ist der schematische Ablauf einer Telekonsultation beschrieben. Eine mögliche Definition kann folgendermaßen[10] angegeben werden:

Definition 2.4.3 (Telekonsultation) *Unter* Telekonsultation *versteht man die interaktive, kooperative Diagnostik des behandelnden Arztes und einem räumlich entfernten Experten. Die notwendigen Informationen – vornehmlich die elektronische Patientenakte – werden parallel zum Gespräch über das Netz ausgetauscht und bearbeitet, mit dem Ziel, eine gemeinsame Diagnose zu stellen.*

Ein wesentliches Merkmal der *Telekonsultation* ist die Interaktion. Beide Teilnehmer, also der behandelnde Arzt sowie auch der Experte, müssen gleichzeitig Zugriff auf alle

[10] In Anlehnung an [Berger 1998]

benötigten Daten haben. Je nach Umfang der bereits durchgeführten Untersuchungen stellt die Forderung der Echtzeit bei der Übertragung eine Herausforderung an die Technik dar.

Soll die Konsultation in Form einer sogenannten *Fallkonferenz* stattfinden, d.h. es nehmen mehrere beteiligte Ärzte an der Konsultation teil, sind nicht sämtliche Daten der elektronischen Patientenakte zur Bewertung des vorliegenden Falls notwendig. Hier wird bei der derzeit praktizierten Variante ohne Einsatz von Telemedizin im Vorfeld ein *virtueller Fall* (s.a. [Kamphenkel]) erstellt, der nur die relevanten Informationen enthält. Eine entsprechende Software sollte somit nicht nur die Übertragung der Informationen in ihrer Gesamtheit ermöglichen, sondern die Ärzte bei der Erstellung des *virtuellen Falls* unterstützen.

Als weiteres Problem wurde in [Kamphenkel] auf die fehlende Möglichkeit der Archivierung der Daten hingewiesen. Bei der Besprechung eines Falls ist möglicherweise nicht nur das Endergebnis entscheidend, sondern auch der Weg, wie es zu diesem Ergebnis gekommen ist. Für eine spätere Überprüfung, falls es zu Problemen bei der weiteren Behandlung kommen sollte, wäre die genaue Herleitung des Entscheidungsprozesses von Bedeutung. Es wird aufgezeigt, dass ein entsprechendes System zusätzlich die Möglichkeit der Versionierung der Daten eröffnen sollte.

Eine besondere Form der Telekonsultation ist das häufig als *Second-Opinion* oder auch *Expertenkonsultation* bezeichnete Szenario. Beim Second-Opinion-Szenario geht man davon aus, dass ein Arzt bereits eine Diagnose gestellt hat, diese aber durch eine zweite Expertenmeinung bestätigt haben möchte. In diesem speziellen Fall können sämtliche bereits erfassten Daten im Vorfeld zum Experten per Datenaustausch gesendet werden. Bei der eigentlichen Konferenz wird lediglich eine Beurteilung basierend auf der „Aktenlage“ durchgeführt.

Eine weitere Spezialanwendung der Telekonsultation stellt die *Teleradiologie* dar, deren grundsätzlicher schematischer Ablauf in Abbildung 2.6 dargestellt ist.

Definition 2.4.4 (Teleradiologie) *Unter* Teleradiologie *versteht man die Befundung von radiologischen Bilddaten – CT- Röntgen- oder MR-Bildmaterial – durch einen räumlich entfernten Radiologen und die damit verbundene Übertragung der Daten.*

Um den Bogen zur ANARAD-Studie zu schlagen, sei auf Abbildung 2.3 verwiesen. Die Expertenkonsultation wird von den befragten Radiologen als besonders bedeutend hervorgehoben, sodass es sich hierbei nicht nur um ein theoretisches Konzept handelt, sonder auch um ein Konzept von erheblichem praktischem Nutzen.

Ein Problem bei der *Teleradiologie* stellt der Umfang der Bilddaten dar, da aus rechtlichen Gründen nur eine verlustfreie Kompression der Daten durchgeführt wird. Falls für die Bewertung der Daten keine Kommunikation benötigt wird, können die Daten auch

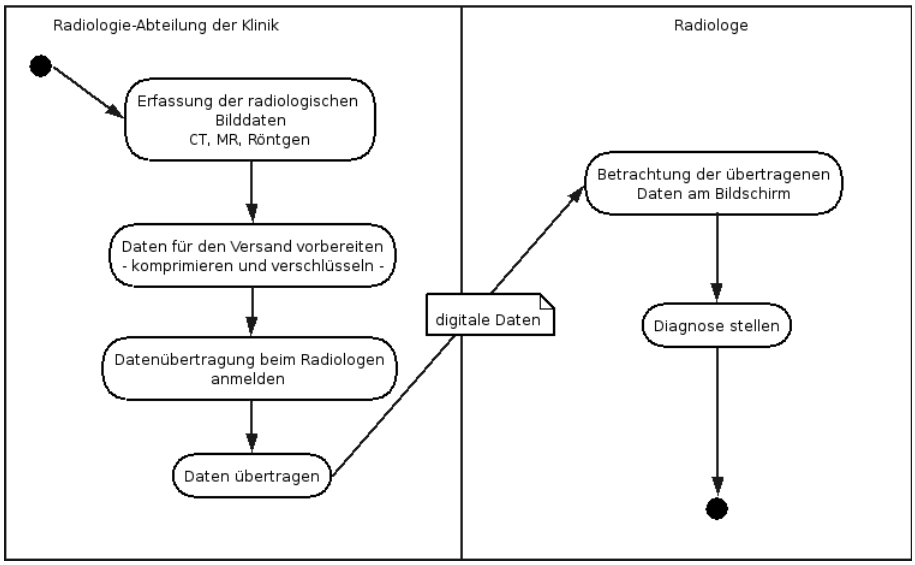


Abbildung 2.6: Arbeitsablauf der Teleradiologie

im Vorfeld über das Netz übertragen werden. In diesem Fall führt der Experte die Befundung anhand der aufgezeichneten Bilder aus. Dies entspricht weitestgehend dem bereits jetzt praktizierten Vorgehen, nur dass die Daten nicht per Fahrdienst transportiert werden. Der Zeitvorteil liegt klar auf der Hand, aber auch Mehrfachuntersuchungen können vermieden werden, da das aktuelle Datenmaterial zeitnah im vollen Umfang vorliegt.

Die bisher angesprochenen Szenarien bedingen zwar die Echtzeit, falls es aber zu Problemen bei der Datenübertragung kommt, ist der Schaden, der entsteht, höchstens finanzieller Natur. Anders verhält es sich bei den folgenden Szenarien. Wird beispielsweise die Medizin-Telematik an einem Unfallort eingesetzt, um eine erste Diagnose zu stellen und somit eine Aussage über die Schwere der Verletzungen zu erhalten (vgl. das Notfallszenario in Abschnitt 3.1), kann ein Fehler im System über Leben und Tod des Verunfallten entscheiden. Ähnlich verhält es sich bei dem folgenden Szenario, welches aufgrund der Brisanz der Ausführung noch am wenigsten praktiziert wird. Bei der *Telechirurgie* übernimmt ein Roboter die eigentliche Operation, der von einem entfernt befindlichen Chirurgen geführt wird.

Der schematische Ablauf der Telechirurgie ist in Abbildung 2.7 dargestellt. Die Akzeptanz dieses Szenarios in der Bevölkerung ist lt. [Berger 1998] nicht gegeben. Der Mensch neigt dazu, einem Arzt aus Fleisch und Blut eher das Vertrauen zu schenken als der Technik, egal wie gut sie auch sei. Betrachtet man den Einsatz der Medizin-Telematik außerhalb der zivilen Nutzung, so bietet dieses Szenario interessante Ansätze in der Militär-Medizin. In [Wootton et al. 2006] wird das Szenario derart ausgelegt, dass ein mobiler Roboter auf dem Schlachtfeld die Rolle der Klinik übernehmen kann.

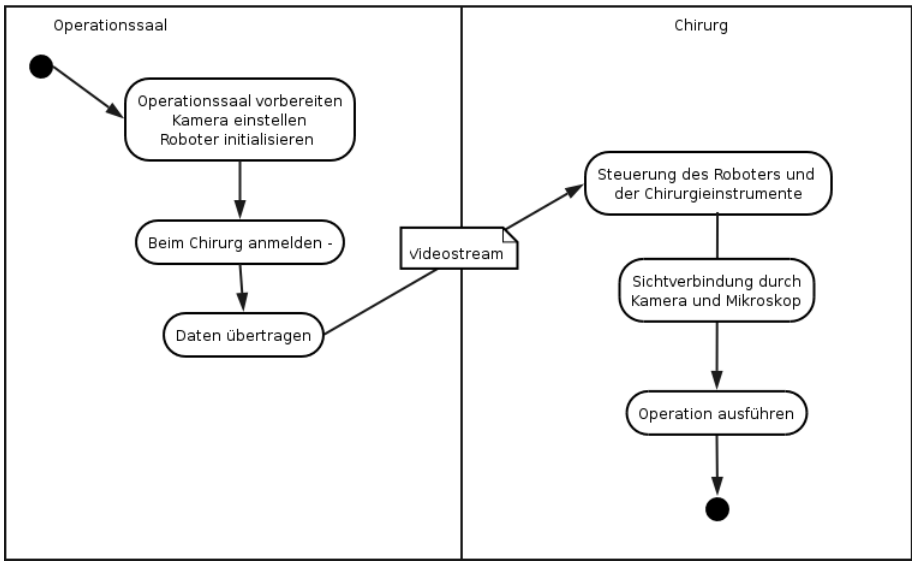


Abbildung 2.7: Arbeitsablauf der Telechirurgie

Beim als *Battle-Field-Surgey* bezeichneten Szenario kann ein sich in sicherer Entfernung befindlicher Arzt einen chirurgischen Eingriff an verwundeten Soldaten durchführen.

Unabhängig ob das Szenario im militärischen oder im zivilen Bereich genutzt wird, ein Ausfall der Übertragung kann schwerwiegende Folgen für den Patienten haben. Daher ist bei solchen Szenarien ein besonderes Augenmerk auf die Ausfallsicherheit zu werfen. Eine Möglichkeit, diesem Sicherheitsanspruch gerecht zu werden, wird in Abschnitt 3.3 auf Basis des SCTP-Protokolls aufgezeigt.

3 Um präklinische Sonographie erweitertes Notfallszenario

Ein wesentliches Szenario wurde im einführenden Abschnitt 2.1 ausgespart, und zwar das *Notfallszenario*. Dem Notfallszenario wird in der vorliegenden Arbeit ein besonderes Augenmerk geschenkt, da es sich in der im Vorfeld stattgefundenen Untersuchung zur Praxisrelevanz (vgl. [Kamphenkel]) von den beteiligten Ärzten und Fachpersonal als wünschenswert und praktisch relevant herausgestellt hat. Diese Erkenntnis findet sich auch in der ANARAD-Studie (vgl. Abschnitt 2.3) wieder, bei der die Notfallkonsultation, wie aus der Analyse der Abbildung 2.3 hervorgeht, als eine der für die Telemedizin bedeutenden Anwendungen bewertet wurde.

Darüber hinaus ist die Entwicklung der technischen Geräte in der Medizin an dem Stand angekommen, an dem große technische Klinikgerätschaften durch Einsatz von Mikroelektronik für den mobilen Einsatz gefertigt werden können. So ist man mittlerweile in der Lage, die komplette sonographische Untersuchung mit kleinen, mobilen präklinischen Sonographiegeräten durchzuführen. Diese Geräte sind bisher nicht flächendeckend im Einsatz, sondern werden derzeit lediglich im Rahmen von Studien und Testläufen eingesetzt. Soll die mobile Sonographie in den Telemedizinworkflow (vgl. Abschnitt 4) eingebunden werden, müssen die generierten Videodaten zusätzlich über das Netz übertragen werden.

Das Notfallszenario, wie es in Abschnitt 3.1 beschrieben wird, stellt demnach auch für die technische Umsetzung eine Herausforderung dar, da die derzeitigen Möglichkeiten nicht geeignet sind, das Szenario in voller Breite abzudecken. Die vorliegende Arbeit hat zum Ziel, diese wissenschaftliche „Lücke“ zu schließen und durch neue Konzepte und Techniken das erweiterte Notfallszenario in der Notfallmedizin zu etablieren.

3.1 Notfallszenario

Das *Notfallszenario* beschreibt den Einsatz der Medizin-Telematik am Unfallort und die notwendige Kommunikation mit der Rettungsleitstelle bzw. dem ausgewählten Zielkrankenhaus. Hierbei werden bereits am Unfallort die Weichen für die folgende notfallmedizinische Versorgung des Verunfallten gestellt. Wesentlich ist der schnelle und kompetente Ablauf der Versorgung des Verunfallten, da der Faktor Zeit die Überlebenswahrscheinlichkeit des Patienten erheblich beeinflusst.

Definition 3.1.1 (Notfallszenario) *Unter dem Begriff* Notfallszenario *versteht man die elektronische Datenerfassung am Unfallort mit mobiler Übertragung der Daten an*

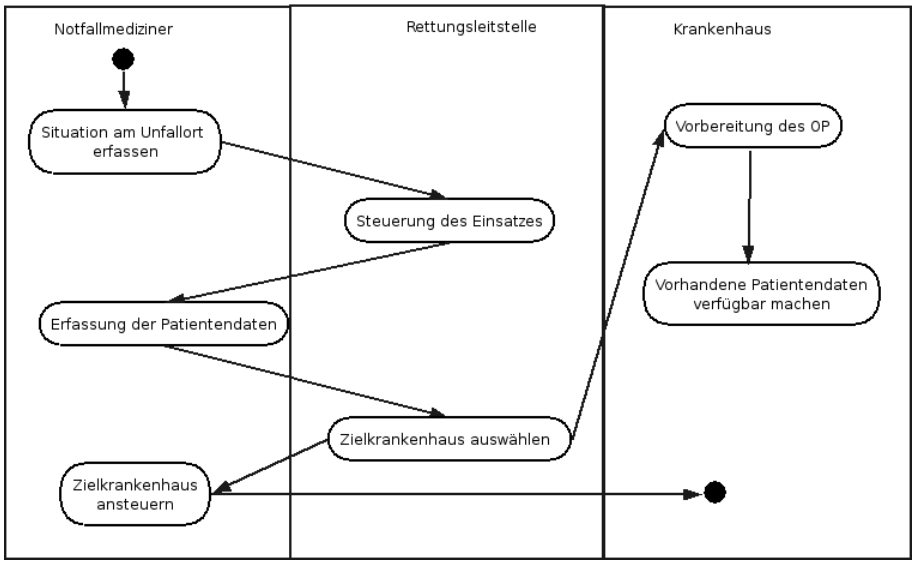


Abbildung 3.1: Arbeitsablauf im Notfallszenario

die Rettungsleitstelle bzw. das so optimal gewählte Zielkrankenhaus.

Auch für das Notfallszenario wird in Abbildung 3.1 der Ablauf schematisch dargestellt. Derzeit werden keine weitreichenden Untersuchungen am Unfallort durchgeführt. Erst nach Einlieferung in das Unfallkrankenhaus wird ein vollständiger Befund erstellt und die notwendigen Maßnahmen ergriffen. Betrachtet man das Szenario in seiner einfachsten Form, können lediglich Patientendaten übertragen werden. Da diese zusätzliche Information der Rettungsleitstelle kaum neue Informationen zur Wahl des Zielkrankenhauses liefert, hat sich das Szenario in dieser Form nicht durchgesetzt bzw. findet kaum Zuspruch unter dem Fachpersonal, seien es die Rettungskräfte oder die Ärzte in der Unfallklinik.

Anders verhält es sich, wenn zusätzliche medizinische Informationen über den Zustand des Verunfallten am Unfallort bereitstehen, die bereits im Vorfeld zur Begutachtung an ein mögliches Zielkrankenhaus gesendet werden können. Dies führt zum *erweiterten Notfallszenario*.

Erweitertes Notfallszenario

Der konkrete Ablauf bei der Versorgung von Patienten am Unfallort ist im Abschnitt 4 auf Basis von Algorithmen dargestellt, insbesondere wird dort die in Europa bevorzugte Strategie des *Stay-and-Play*, wie sie in Definition 4.2.3 festgelegt wurde, fokussiert. Bei Stay-and-Play geht man davon aus, dass durch die direkte Versorgung des Patienten am Unfallort die Überlebenswahrscheinlichkeit des Patienten erhöht werden kann.

Beim Stay-and-Play werden am Unfallort bereits Untersuchungen vorgenommen, die bei der klassischen Scoop-and-Run-Strategie, wie sie in Definition 4.2.2 gegeben ist, erst beim Zielkrankenhaus vorgesehen sind. Die Stay-and-Play-Strategie ermöglicht demnach auch, weiterführende Untersuchungen durchzuführen, die durch den Einsatz von mobilen medizinischen Geräten ermöglicht werden. Dies führt zur folgenden Definition des *erweiterten Notfallszenarios*:

Definition 3.1.2 (erweitertes Notfallszenario) *Unter dem Begriff* erweitertes Notfallszenario *versteht man die elektronische Datenerfassung mittels mobiler bildgebender Verfahren am Unfallort mit mobiler Übertragung der Daten an ein speziell vorbereitetes Krankenhaus, welches aufgrund der übermittelten Daten einen Befund über den Zustand des Patienten stellt und so den weiteren Ablauf der Rettungskette beeinflusst.*

Der Unterschied zum Notfallszenario nach Definition 3.1.1 besteht darin, dass die Verantwortung bzw. die Entscheidung des weiteren Vorgehens vom Rettungspersonal auf ein Expertenteam verlagert werden kann. Ein großes Problem beim Einsatz von mobilen medizinischen Geräten besteht darin, dass die korrekte Benutzung und Befundung der so erzeugten Bilddaten gewährleistet werden muss. Da das Personal in einem Rettungswagen im Normalfall keine spezielle Ausbildung auf diesem Gebiet besitzt, wird die Bewertung von einem Experten benötigt. Dieser kann durch das erweiterte Szenario herangezogen werden.

Welche bildgebenden Verfahren eignen sich für die Untersuchung am Unfallort bzw. liefern wichtige Informationen, die für die Auswahl des Zielkrankenhauses entscheidend sind? Ein bildgebendes Verfahren wird ohne Einsatz von Telemedizin in Teilbereichen bereits erfolgreich eingesetzt, nämlich die präklinische Sonographie. Im Folgenden wird das erweiterte Notfallszenario unter Anwendung der mobilen Sonographie als Referenzszenario beschrieben. Anhand dieses Referenzszenarios werden die Probleme und mögliche Problemlösungen des erweiterten Notfallszenarios deutlich.

3.2 Präklinische Sonographie

Die Sonographie ist ein etabliertes bildgebendes Verfahren im stationären Bereich, welches sich gerade bei der Versorgung von Notfallpatienten im Schockraum zur Diagnose bewährt hat. In Abschnitt 4 wird das Polytraumamanagement auch aus Schockraumsicht (vgl. den Schockraumalgorithmus in Abbildung 4.3) besprochen. An dieser Stelle bleibt festzuhalten, dass die Thorax und die Abdomen-Sonographie zur Diagnose, insbesondere um festzustellen, ob freie Flüssigkeit in den Abdomen oder in den Hohlräumen des Thorax vorliegt, ein wesentlicher und wichtiger Aspekt des Schockraumalgorithmus ist.

Die stationäre Sonographie hat sich bewährt, u.a. auch, weil es ein als risikoarm bewertetes Verfahren ist und so auf aufwendige Strahlenschutzmaßnahmen verzichtet werden

kann. Die Sonographie kann nichtinvasiv, schmerzlos und strahlenexpositionsfrei durchgeführt werden. Hinzu kommt der Zeitfaktor, da die Sonographie schnell durchführbar ist und die hohe Verfügbarkeit im stationären Bereich, da jede Klinik mit den entsprechenden medizinischen Geräten ausgestattet ist. Zudem ist die Echtzeit als enorm positiv zu bewerten, da die Sonden in Echtzeit kontrolliert werden können und somit auch die gewünschten Schnittbilder in Echtzeit vorliegen. Betrachtet man die Sonographie unter wirtschaftlichen Aspekten, bleibt festzuhalten, dass die Anschaffungs- bzw. die Betriebskosten im direkten Vergleich zu anderen bildgebenden medizinischen Verfahren gering sind.

Zeitgewinn – höhere Überlebenswahrscheinlichkeit durch präklinische Sonographie

Die stationäre Sonographie hat einen festen Platz in der Diagnostik einer Klinik. Die mobile Sonographie für die notärztliche Versorgung von Unfallpatienten ist dagegen noch in der Aufbauphase. Es ist lediglich ein kleiner Teil des Rettungspersonals mit entsprechenden Geräten ausgestattet, insbesondere wenn die Geräte im Versuchsstadium oder im Rahmen einer wissenschaftlichen Studie Verwendung finden. Welchen Vorteil verspricht man sich vom Einsatz der präklinischen Sonographie?

Der Zeitgewinn bei der Behandlung von Schwerverletzten ist von besonderer Bedeutung. In Abbildung 3.2 ist das derzeitige Vorgehen bei einem Notfalleinsatz, dem sogenannten Ist-Prozess, im Vergleich mit dem von uns angestrebten Vorgehen, dem Soll-Prozess, wie er in [Kamphenkel et al. 2006] aufgestellt wurde, dargestellt. Medizinisch kann der Zeitgewinn folgendermaßen begründet werden:

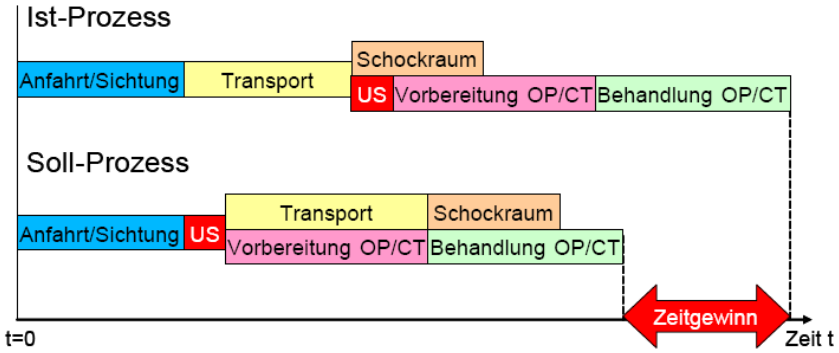


Abbildung 3.2: Prozessveränderung durch mobile Sonographie

Bei der Behandlung polytraumatisierter Unfallopfer nimmt nach [Walcher 2003] die abdominelle Verletzung in der präklinischen und frühen klinischen Versorgung einen zentralen Stellenwert ein, da abdominelle Massenblutungen neben der Schädel-Hirn-Verletzung

die wesentliche Todesursache darstellen.

Dabei nimmt die Sonographie bezüglich der Diagnostik des Abdominaltraumas mit intraperitonealer Blutung eine zentrale Rolle im Schockraum und in der Notaufnahme ein [McGahan et al. 2002]. Bisher weist die Diagnosesicherheit der Notärzte bezüglich der Erkennung von abdominellen Verletzungen ohne Einsatz der sonographischen Untersuchungstechnik erhebliche Fehlerquoten auf [Lechleuthner et al. 1994]. Mit der Entwicklung und Verfügbarkeit mobiler Ultraschallgeräte und Anwendung der präklinischen Sonographie kann die intraabdominelle Blutung mit einer hohen Sicherheit bereits am Unfallort festgestellt werden, was im positiven Fall wesentliche therapeutische und logistische Veränderungen des präklinischen Managements nach sich zieht. Insbesondere hat bei einer abdominellen Massenblutung der Patiententransport absolute Priorität, während erweiterte therapeutische Maßnahmen am Unfallort relativiert werden müssen [Walcher 2003].

Die verstrichene Zeit bis zur definitiven operativen Blutstillung spielt die entscheidende Rolle in der präklinischen und frühen klinischen Phase [Clarke et al. 2002]. Eine frühzeitige Diagnose von Art und Ausmaß abdomineller Verletzungen bereits am Unfallort mit Übermittlung der Befunde und Diagnosen an das Zielkrankenhaus ermöglicht dort die Einleitung weiterführender diagnostischer (z.B. Computertomographie) und therapeutischer (z.B. Operation) Maßnahmen und die Organisation des erforderlichen medizinischen Fachpersonals (Anästhesie, OP-Team, Radiologie).

Auch die Auswahl des Zielkrankenhauses an sich kann entsprechend dem Verletzungsmuster vorgenommen werden. Insgesamt wird durch die Vermeidung von logistischen Fehlentscheidungen während der präklinischen Phase nach [Walcher 2003] eine deutliche, mitunter lebensrettende Verkürzung der Prozessketten erreicht. Dies entspricht dem gesuchten Zeitgewinn, wie er in Abbildung 3.2 dargestellt ist.

Die Deutsche Rettungsflugwacht entschied sich deshalb bereits Anfang 2004 für den Einsatz von mobilen Ultraschallgeräten in zunächst fünf ihrer Rettungshubschrauber. In [drf] wird der Feldversuch der Rettungsflugwacht betrachtet, wobei jeder Rettungsassistent eine spezielle Zusatzausbildung absolvieren musste, da die Anwendung des technischen Geräts nicht trivial ist.

Dies wurde auch von der Arbeitsgruppe von Walcher (s.a. [Walcher 2003]) erkannt, die als wesentliches Problem beim präklinischen Einsatz der Sonographie die Ausbildung und das Training von Notärzten und Rettungsassistenten in der Anwendung dieser Technik identifiziert haben. Zwar wurden spezielle Ausbildungskonzepte für die in der Notfallversorgung involvierten Ärzte erarbeitet – in [Frezza et al. 1999], [Hoffmann et al. 2002], [Ma et al. 1995] und [Nast-Kolb et al. 1993] sind diese u.a. ausführlich dargestellt – und eine steile Lernkurve bei den Absolventen entsprechender Kurse beobachtet; dennoch ist die Gefahr, dass unerfahrene Untersucher lange Zeit am Unfallort benötigen, um einen unsicheren Befund zu erheben und daraus ggf. falsche Konsequenzen

ableiten, nicht von der Hand zu weisen. Dies wird sogar vom Hauptbefürworter dieser Technik in [Walcher 2003] bestätigt.

Der Einsatz präklinischer Telesonographie zum Zielkrankenhaus bietet hier entscheidende methodische Vorteile. So ist es für ein fachlich hoch qualifiziertes Expertenteam, das auch die weiteren diagnostischen und therapeutischen Schritte in der Klinik betreuen wird, im Zielkrankenhaus möglich, die Untersuchung „live“ mitzuverfolgen. Durch die Kontrolle der Schnittführung in Echtzeit können ergänzende Schnittbilder bzw. Organdarstellungen von Remote angefordert und beurteilt werden. Die Qualifikation des Untersuchers vor Ort kann damit auf die rein technische Durchführung der Untersuchung begrenzt werden.

Abbildung 3.3: Modell der präklinischen Sonographie mit Anbindung über Datenleitung an ein speziell vorbereitetes Krankenhaus

Das hier beschriebene erweiterte Notfallszenario unter Verwendung der präklinischen Sonographie ist im Modell in Abbildung 3.3 anschaulich abgebildet. Am Modell kann man die benötigten Komponenten bereits ablesen und den einzelnen Teilbereichen zuordnen. Wenn es zu einem Unfall gekommen ist, fahren der Rettungswagen und der Notarzt getrennt zur Unfallstelle. Der Notarzt würde die notwendige Sonographie durchführen und parallel die Daten an das Klinikum übertragen, wo das Expertenteam die Befundung durchführt. Anhand des Befunds ist das Rettungsteam in der Lage, eine Entscheidung hinsichtlich weiterer Maßnahmen am Unfallort bzw. Wahl des Zielkrankenhauses zu tref-

fen.

Der FAST-Algorithmus – Standardisierte Methode der Ultraschalluntersuchung

Dr. R. Adams Cowley wird häufig als Vater der modernen Polytraumaversorgung bezeichnet. In seinem Artikel [Cowley 1976] schreibt er der unzureichenden bzw. nicht vorhandenen Blutzirkulation und den damit einhergehenden Veränderungen im Körper als häufigste Todesursache bei Schockpatienten zu. Er prägte den Begriff der *golden hour of shock*, mit dem er zum Ausdruck bringt, dass die Zeit den wesentlichen Faktor bei der Behandlung von traumatisierten Patienten darstellt. Unter der Voraussetzung, dass die Blutung innerhalb der ersten Stunde nach Eintritt des Schocks gestoppt werden kann, können die meisten Patienten stabilisiert und damit gerettet werden. Die heute praktizierte Versorgung von Notfallpatienten orientiert sich an dieser Maxime, sodass die Zeit, die für eine erste Untersuchung eines Verunfallten benötigt wird, über Leben und Tod entscheiden kann.

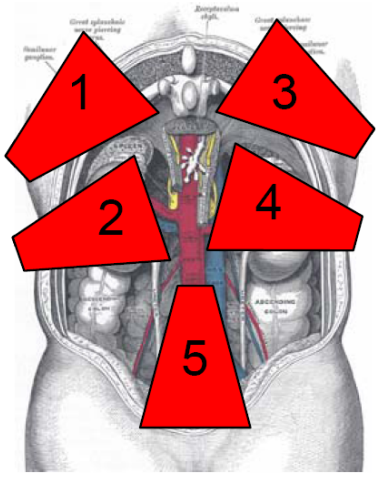


Abbildung 3.4: Schnittebenen beim FAST-Algorithmus – Focused assessment with sonography for trauma

Für die Sonographie steht mit dem FAST-Algorithmus eine schnelle und zuverlässige Methode zur Verfügung, um frühzeitig intraabdominelle oder intrathorakale Blutungen festzustellen. Die folgende Beschreibung orientiert sich an [Walcher et al. 2002], weitere Quellen sind u.a. [Busch 2006], [Tayal et al. 2004] und das Kapitel 3 aus [Gaspari et al. 2005]. Mit dem FAST-Algorithmus ist der behandelnde Arzt in der Lage, in nur zwei Minuten sämtliche Schnittbilder zu erstellen, mit der die Verdachtsdiagnose erhärtet werden kann.

Definition 3.2.1 (FAST) *Der* FAST-Algorithmus – Focused assessment with sonography for trauma (FAST) – *ist eine Methode, mit der man innerhalb von wenigen Minuten*

die wesentlichen Bereiche des Körpers, drei Bereiche der Abdomen und ein Bereich des Thorax, unter Verwendung von Ultraschall auf freie Flüssigkeit untersuchen kann.

Ziel der Untersuchung ist es, festzustellen, ob sich freie Flüssigkeit, in der Regel handelt es sich um Blut, in den Abdomen angesammelt hat. Im Ultraschallbild kann dies anhand der eindeutigen Schwarzfärbung der Flüssigkeit mit hoher Wahrscheinlichkeit festgestellt werden. In Abbildung 3.4 sind die verschiedenen Schnittebenen dargestellt. Mit dieser Methode können die Niere, die Leber, der Morrison-Pouch, das Zwerchfell, die Pleura, die Milz, der Koller-Pouch und der Douglasraum untersucht werden. Durch eine Pericarduntersuchung wird auch der Herzbeutel begutachtet. Mit nur fünf Schnittbildern kann das Vorkommen von freier Flüssigkeit zuverlässig bestimmt werden. Es werden hierfür fünf Standardeinstellungen des Schallkopfs verwendet, die in Abbildung 3.4 durchnummeriert sind. Nach [Walcher 2003] ist die Reihenfolge aus praktischen Erwägungen einzuhalten.

Die Anwendung des FAST-Algorithmus bildet somit eine Grundlage für den Einsatz der mobilen Sonographie im erweiterten Notfallszenario. Ein Beispiel zur Anwendung des FAST-Algorithmus folgt in Abschnitt 3.4.2, in dem auf die Realisierbarkeit des erweiterten Notfallszenarios fokussiert wird.

Prospektive Studien – Bestätigung der präklinischen Sonographie

Die präklinische Sonographie ist eine junge Disziplin, da die Miniaturisierung der technischen Geräte erst in den letzten Jahren brauchbare Ergebnisse geliefert hat. Mittlerweile stehen kleine, handliche Sonographiegeräte zur Verfügung, die Bilder in hoher Qualität und Auflösung produzieren und zudem einfach zu handhaben sind.

Die präklinische Sonographie ist somit aus technischer Sicht anwendbar. Es stellt sich die Frage, ob es aus medizinischer Sicht sinnvoll ist, auf die präklinische Sonographie zurückzugreifen. Sinnvoll heißt in diesem Fall: Kann durch die zusätzliche Anwendung am Unfallort die Überlebenswahrscheinlichkeit des Verunfallten signifikant gesteigert werden? Diese Frage wird in der Literatur kontrovers diskutiert, wobei der Nutzen durch eine Vielzahl von Studien bereits nachgewiesen wurde.

Die Studie, die Dr. Walcher in [Walcher et al.] in Zusammenarbeit mit der deutschen Luftrettung durchgeführt hat, wurde bereits erwähnt. Die Ergebnisse der Studie werden kurz zusammengefasst. Die Untersuchungen selber wurden dabei von einem nicht an der eigentlichen Notfallversorgung teilnehmenden Untersucher durchgeführt. Insgesamt wurde die Ultraschalluntersuchung bei 61 Personen durchgeführt, bei denen intraabdominelle Verletzungen angenommen wurden. Die Untersuchungsdauer lag mit 2,8 Minuten im Schnitt etwas über den für den FAST-Algorithmus in der Literatur angegebenen zwei Minuten (vgl. Abschnitt 3.2). Bei 25,2 Prozent der untersuchten 61 Personen, also 16 Patienten, wurde freie Flüssigkeit gefunden. Bei 7 von den 16 Patienten konnte ein massiver und bei 9 Patienten ein moderater pathologischer Befund gestellt werden. In [Walcher et al.] wurde als Ergebnis festgehalten, dass sich die präklinische Sonographie als sichere

Methode bewährt hat.

Bleibt die Frage nach dem Nutzen oder anders ausgedrückt, welche Maßnahmen aufgrund der Feststellung, dass freie Flüssigkeit vorliegt, ergriffen wurden, die ohne die Untersuchung nicht durchgeführt worden wären. Betrachtet man alle Einsätze des Rettungspersonals, so wurde die Wahl des Zielkrankenhauses bei 21 Prozent der untersuchten Fälle beeinflusst. Somit konnten durch die zusätzliche Untersuchung bei jedem fünften Einsatz wichtige Erkenntnisse gewonnen werden, die den weiteren Ablauf der Versorgung entscheidend und nachhaltig veränderten.

Aufgrund dieser positiven Ergebnisse wurde von derselben Arbeitsgruppe eine weiterführende Studie durchgeführt, die auf einen Zeitraum von 12 Monaten ausgelegt war. Als Resümee dieser Studie kann festgehalten werden, dass die Ergebnisse bestätigt werden konnten und die präklinische Sonographie als echte Ergänzung zur Versorgung des Patienten am Unfallort geeignet ist.

Zu dem gleichen Ergebnis kommt eine Studie, die am Trauma-Center in Kalifornien durchgeführt wurde. Das Anliegen der Studie war nach [McGahan et al. 2002] die Bestimmung der Genauigkeit, die beim Auffinden von Hemoperitoneum und massiver Organverletzungen, welche durch stumpfes Trauma verursacht wurden, erreicht werden kann. Innerhalb von drei Jahren wurden 3264 Patienten notfallsonographisch untersucht. Zur Bewertung der Ergebnisse dieser Untersuchungen wurden alle Patienten mit intraabdomnialen Verletzungen identifiziert und die Befunde mittels CT und den operativen (maßgeblischen) Befunden und klinischen Ergebnissen verglichen. Die Studie belegt, dass die Sonographie zur Bewertung von Patienten mit Verletzungen durch stumpfe Traumata hoch präzise und spezifisch erfolgt.

Als letzte Studie soll die Studie der norwegischen Air-Ambulance (vgl. [Busch 2006]) angeführt werden. In Norwegen wurde bei dieser dreimonatigen, prospektiven, weitsichtigen Studie die Sonographie beim präklinischen Patientenmanagement eingesetzt. Als Ergebnis der Studie wurde festgehalten, dass die präkinische Sonographie, wenn sie durch einen kompetenten Untersucher angewendet wird, von hohem diagnostischem und therapeutischem Nutzen ist. Der angewendete FAST-Algorithmus (vgl. Definition 3.2.1) führte zu keiner Verzögerung des Patientenmanagements, sodass dem Faktor Zeit der präklinischen Sonographie nichts entgegensteht.

3.3 Technische Lösungsansätze aus medizinischer Sicht

Grundlage für den Einsatz von mobiler Sonographie im erweiterten Notfallszenario ist eine ausgereifte Technik, die es ermöglicht, verschiedene Datenströme effizient und sicher zwischen Notarztwagen und Krankenhaus zu übertragen. Im Bereich der Notfallmedizin sind neben den umfangreichen Videodaten aus der mobilen Sonographie weiterhin Sprachdaten, wie sie bei der Kommunikation von Notfallarzt am Unfallort und einem

Experten im Klinikum anfallen, und ggf. auch Textdaten, falls Informationen aus der Patientenakte übermittelt werden sollen, zu übertragen. Um einen kostengünstigen flächendeckenden Einsatz gewährleisten zu können, muss auf einer bestehenden Infrastruktur aufgebaut werden.

Im Einzelnen heißt das, dass die bestehenden IP-Netzstrukturen verwendet werden müssen. Hierbei wird als Transportprotokoll derzeit hauptsächlich auf TCP und UDP gesetzt, wobei beide Protokolle in die Jahre gekommen sind und für die hier angenommene Komplexität der Übertragung nicht geeignet erscheinen.

Als Alternative bietet sich das relativ neue *Stream-Control-Transmission-Protokoll* oder kurz SCTP an, wie es in [Stewart and Xie 2001] spezifiziert ist. Das SCTP-Protokoll ist auf Grund seiner modernen Architektur sowie seiner Flexibilität und Erweiterbarkeit besonders geeignet, die hohen Anforderungen zu erfüllen. Bereits existierende bzw. in der Planung befindliche Erweiterungen machen SCTP für unterschiedliche Anwendungsbereiche mit konkurrierenden Anforderungen an das Transportprotokoll interessant. TCP und UDP sind etablierte Protokolle in der Transportschicht, die nicht ohne Weiteres durch einen neuen Standard abgelöst und somit parallel zu SCTP weiter betrieben werden können. Die Vorteile von SCTP gegenüber den älteren Protokollen können so bereits heute genutzt werden.

Die Verwendung von SCTP zieht sich wie ein roter Faden durch die gesamte Arbeit, daher werden das Protokoll sowie die möglichen Erweiterungen in Abschnitt 6 ausführlich aus technischer Sicht behandelt. In diesem Unterabschnitt werden einige Vorteile herausgestellt, die sich direkt in den fachlichen Kontext einbeziehen lassen und somit als technisches Plug-In für das Notfallszenario betrachtet werden können.

Für multimediale Echtzeit-Anwendungen wird ein zumindest teilweiser verbindungsloser Aufbau der Verbindung benötigt. Kommt es zum Verlust von IP-Paketen, so bringt die Neuübertragung im Falle einer Echtzeitanwendung keinen Nutzen, da die enthaltenen Videoinformationen nicht mehr aktuell sind und vom Empfänger nicht mehr genutzt werden können. Wenn auf die Neuübertragung von verloren gegangenen Pakten verzichtet wird, spricht man auch von einem *teilgesicherten Transport*, der im SCTP-Standard aber nicht vorgesehen ist. Um einen teilgesicherten Transport auch unter Verwendung von SCTP zu ermöglichen, muss auf Erweiterungen zurückgegriffen werden. Eine mögliche Erweiterung ist als PR-SCTP bekannt, welches in Abschnitt 10.2 beschrieben wird. SCTP kann somit unter Verwendung dieser Erweiterung zur Übertragung von medizinischen Video-Echtzeitdatenströmen verwendet werden.

Von SCTP werden mit *Multihoming* und *Multistreaming* zwei grundsätzlich neue Konzepte in der Transportschicht realisiert, die in dieser Form weder von *TCP* noch von *UDP* realisert sind. Diese Eigenschaften ermöglichen ein neuartiges Design der technischen Infrastruktur im erweiterten Notfallszenario.

Multihoming zur Verbesserung der Ausfallsicherheit

Multihoming ist eine Möglichkeit von SCTP, auf Verzögerungen im Netz zu reagieren, indem der Einbau von Redundanzen in der Netzwerkschicht die möglichen Pfade vom Sender zum Empfänger erhöht. Jedem Endpunkt einer SCTP-Verbindung können mehrere IP-Adressen zugewiesen sein. Die vollständige Definition 5.1.1 des Begriffs *Multihoming* wird in Abschnitt 5.1 nachgereicht. Im zweiten Teil der Arbeit wird die Multihoming-Eigenschaft von SCTP dazu verwendet, sämtliche Pfade zur Datenübertragung zu nutzen, um so das Gesamtvolumen der Übertragung zu erhöhen. Dies entspricht einer Erweiterung des grundsätzlichen Konzepts des Multihomings und führt zur Definition 5.1.2 der *konkurrierenden Multipfadübertragung*. Im SCTP-Standard wird die Multihoming-Eigenschaft lediglich zur Erhöhung der Ausfallsicherung verwendet.

Beim *Multihoming* wird eine Verbindung zwischen zwei Endpunkten als *einfacher Pfad* bezeichnet. Ein *herausgehobener Pfad*, der die Hauptlast der Übertragung trägt, heißt *primärer Pfad* oder *Primärpfad*. Eine Verbindung wird unter SCTP als *Assoziation* bezeichnet. Einer Assoziation können demnach mehrere einfache Pfade zugeordnet werden, wobei ein Pfad als Primärpfad ausgewiesen wird. Die weiteren Pfade werden als *Sekundärpfade* bezeichnet. Mit diesen einfachen Begriffen kann das Multihoming vollständig beschrieben werden.

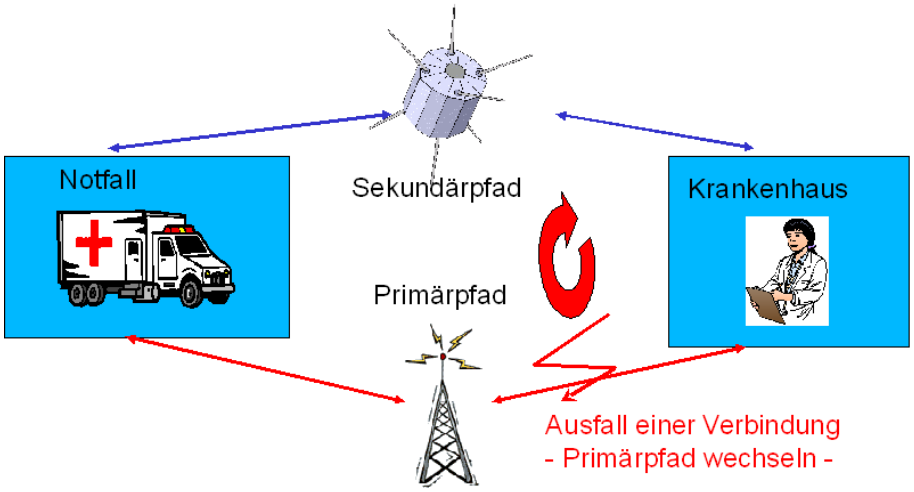


Abbildung 3.5: Erweitertes Notfallszenario – Ausfallsicherheit wird durch Verwendung von mehreren Pfaden ermöglicht.

In der Grundkonfiguration werden die zusätzlichen Sekundärpfade nur zur Neuübertragung von verloren gegangenen Paketen genutzt, während auf dem Primärpfad die eigentliche Datenübertragung stattfindet. Da während des Betriebs Informationen über den Zustand der einzelnen Pfade bzw. ihrer Verfügbarkeit ermittelt werden, kann durch

diesen Mechanismus ein ausgereiftes Fehlermanagement und eine erweiterte Fehlererkennung hergeleitet werden. Auch der komplette Ausfall einer Ende-zu-Ende-Verbindung führt daher nicht zum Abbruch der Gesamtverbindung. Speziell für den Einsatz im Bereich der mobilen Sonographie kann somit maximale Übertragungssicherheit erreicht werden.

In Abbildung 3.5 ist der klassische Anwendungsfall im Notfallszenario dargestellt, wobei zwei Pfade (in der Abbildung rot und blau) für die Assoziation bereitgestellt wurden. Der rote Pfad wurde als Primärpfad gewählt, während der blaue Pfad als Sekundärpfad als Ausweichpfad zur Verfügung steht. Es kann sichergestellt werden, dass bei einem Ausfall einer Verbindung zwischen Notarztwagen und Krankenhaus die Gesamtverbindung aufrechterhalten bleibt, da beim Ausfall der Primärpfad gewechselt wird. Im Beispiel wird der blaue Pfad zum Primärpfad, und die Übertragung kann fortgesetzt werden. Damit kann die Diagnose ohne Einwirken des Notarztes trotz Ausfall der Verbindung fortgeführt werden.

Multistreaming zur Vermeidung von Verzögerungen im Datentransfer

In einem klassischen TCP-Netzwerk können Übertragungsfehler zu Verzögerungen für nachfolgende Pakete führen. Beim Entwurf von SCTP wurde diese Problematik erkannt und durch verschiedene Verfahren der Durchsatz deutlich verbessert. Eine Möglichkeit besteht in der Trennung der Reihenfolgensicherung einzelner Pakete von der Forderung nach einem zuverlässigen Transport der Daten. Dazu werden zusätzlich logische Übertragungskanäle zwischen Sender und Empfänger ausgehandelt, die sogenannten *Streams*. Die formale Definition des hier eingeführten Begriffs wird im Abschnitt 6, speziell in Definition 6.1.5, nachgeholt.

Im Normalfall wird von einer reihenfolgen-gesicherten Zustellung ausgegangen. Die Aufteilung des Datenstroms in mehrere Teilströme ermöglicht es, die o.a. Verzögerung zu verringern, da jeder Teilstrom einzeln betrachtet wird. Geht ein Nachrichtenpaket verloren, so muss im Falle einer zuverlässigen Übertragung das entsprechende Paket erneut gesendet werden, was sich wiederum auf den Versand der nachfolgenden Pakete auswirkt, die ihrerseits nicht übertragen werden können. Werden dahingegen verschiedene Streams zur Übertragung genutzt, blockiert ein fehlerhaft übertragenes Paket lediglich den ihm zugeordneten Stream, und die Übertragung der Daten auf den anderen Streams kann ohne Störung fortgesetzt werden.

Dieses als *Multistreaming* (vgl. auch Abschnitt 6.5.1 für die technische Betrachtung) bekanntes Verfahren ermöglicht es weiterhin komplexe Daten auf kleinere strukturelle Teildaten aufzuspalten und in getrennte Streams zu übertragen. Jeder einzelne Stream verfügt über einen eigenen Auslieferungsmechanismus, sodass ein blockierter Stream keinen negativen Einfluss auf die anderen Streams nimmt. Eintreffende IP-Pakete eines nicht blockierten Streams können somit ausgeliefert werden, auch wenn Pakete eines

anderen Streams bereits überfällig sind. In [Ladha und Amer 2003a] wurde diese Eigenschaft von SCTP beispielsweise dazu verwendet, einen effizienten Filetransfer über SCTP zur realisieren.

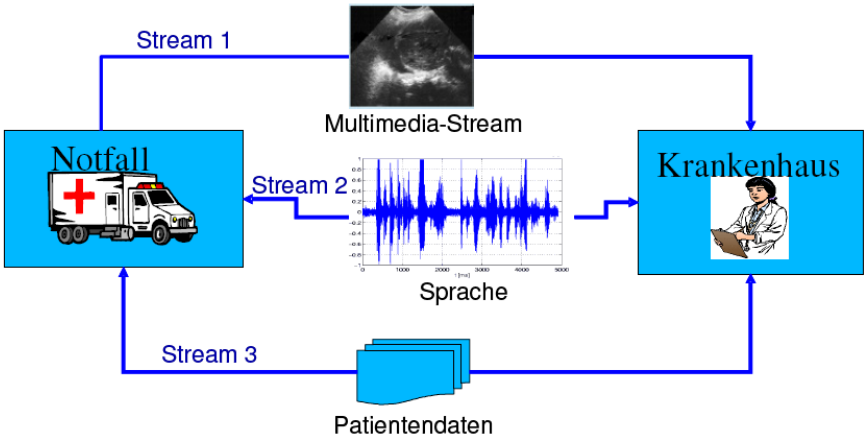


Abbildung 3.6: Erweitertes Notfallszenario – Verwendung von verschiedenen Streams unterschiedlicher Zuverlässigkeitsklassen

Für das erweiterte Notfallszenario, wie es in Abbildung 3.6 dargestellt ist, können durch den Einsatz von Multistreaming Verzögerungen im Datentransfer vermieden oder zumindest verringert werden. Die Informationen, die für die Fernbefundung notwendig sind, können in verschiedene Zuverlässigkeitsklassen eingeteilt werden. So gibt es Videodaten, die, wie im vorherigen Abschnitt erläutert, mittels PR-SCTP, also im teilgesicherten Modus übertragen, werden können. Ähnliches gilt für Sprachinformationen, da diese ebenfalls in Echtzeit übertragen werden müssen. Anders sieht es bei den Textinformationen aus. Hier kommt es darauf an, dass die Daten vollständig und korrekt beim Empfänger ankommen, sodass sich ein gesicherter Transport anbietet. Jedem verwendeten Stream wird eine Zuverlässigkeitsklasse zugeordnet.

Zum einen stört eine gestörte Übertragung der Videodaten nicht die Übertragung der Sprachinformationen, da diese nicht blockiert werden, zum anderen kann die Patientenakte im gesicherten Modus übertragen werden, ohne damit die Video- und Sprachinformation zu beeinflussen.

Ein weiterer Vorteil liegt in der Möglichkeit, jeden Stream gesondert kryptographisch zu behandeln. Sollen beispielsweise die Patientendaten verschlüsselt und authentifiziert übertragen werden, da hier sicherheitsrelevante Personendaten übertragen werden, kann speziell für diesen Stream die notwendige Verschlüsselung eingestellt werden. Das Telefongespräch zwischen Notarzt und Klinikum kann, falls hier die Verschlüsselung nicht

zwingend notwendig ist, ggf. ohne kryptographische Behandlung versendet werden. Jeder Stream kann somit einzeln konfiguriert werden. Die kryptographische Behandlung von komplex strukturierten Daten und die nur teilweise Verschlüsselung von sogenannten Kerninformationen, die zu einer deutlich effizienteren Verschlüsselung der Gesamtinformation führt, wird ausführlich in Teil III dieser Arbeit besprochen.

3.4 Basisprojekt zur Funkübertragung bewegter Ultraschallbilder

Die theoretischen Untersuchungen von Telemedizin-Szenarien motivierten zu einer Untersuchung hinsichtlich der technischen Machbarkeit und einer möglichen praktischen Realisierung. Als Grundlage wurde aufgrund des großen Zuspruchs von Ärzten und Klinikpersonal das im vorherigen Unterabschnitt besprochene *erweiterte Notfallszenario*, wie es in Definition 3.1.2 eingeführt wurde, gewählt.

Die Untersuchung hatte zum Ziel, auf Basis existierender Hard- und Softwarekomponenten die notwendige Infrastruktur zu schaffen bzw. im Laborumfeld nah der Realität abzubilden. Dieses Kapitel gibt einen kurzen Abriss über durchgeführte Versuche, Aufbauten und Ergebnisse, die sich aus der Zusammenarbeit der Projektteilnehmer ergeben haben.

3.4.1 Beschreibung des Vorhabens

Das Projekt hat zur Aufgabe, die Fernübertragung von bewegten sonographischen Bilddaten mittels bestehender Komponenten zunächst in einer Testumgebung umzusetzen und anschließend zu analysieren, inwieweit eine Fernbefundung der übertragenen Daten aus medizinischer Sicht durchführbar ist.

Voraussetzungen

Das Szenario soll technisch auf eine Testumgebung abgebildet werden. Die hierfür zu beschreibenden Teilbereiche werden in Einzelteststellungen aufgeteilt.

Definition 3.4.1 (Einzelteststellung) *Unter dem Begriff der* Einzelteststellung *werden die zu simulierenden Teststellungen zusammengefasst, die einen Teilbereich des realen Szenarios abdecken. Für das erweiterte Notfallszenario werden die Einzelteststellungen*

- *Unfallstelle,*
- *Übertragungsmedien und*
- *Expertenteam im Klinikum*

benötigt.

Durch die Zerlegung des Geseamtszenarios in Einzelteststellungen ist man in der Lage, die einzelnen Teilbereiche unabhängig voneinander zu betrachten und die Ergebnisse in einem Gesamtergebnis zusammenzufassen. Jede Einzelteststellung stellt andere Anforderungen an Hard- und Software.

Abbildung 3.7: Testaufbau mit den Kernkomponenten

Die Kernkomponenten sind in Abbildung 3.7 zusammengestellt. Um die einzelnen Geräte tatsächlich im Rettungswagen bzw. in der Klinik zu installieren, werden weitere technische Komponenten benötigt. Im Ausblick auf Seite 52 wird die Teststellung „vollständig“ im Sinne der praktischen Realisierbarkeit dargestellt. Insbesondere werden zusätzliche Komponenten für die sichere Übertragung sowie für den speziellen Einsatz von UMTS beschrieben.

3.4.2 Unfallstelle

Für die Abbildung der *Unfallstelle* wird ein mobiles Ultraschallgerät benötigt, welches die zu befundenden und zu übertragenden Daten erzeugt. Die Daten werden dann per WLAN an den Rettungswagen übermittelt. Im Rettungswagen ist ein Modul zur Übertragung der Daten installiert. Für die Untersuchung stand ein mobiles Sonographiegerät der Firma *Sonosite* zur Verfügung, das zusätzlich über die notwendigen Videoausgänge verfügt, sodass die generierten Ultraschalldaten direkt vom Sonographiegerät ausgelesen werden können.

In der Literatur kursieren aufgrund der verschiedenen technischen Geräte und Hersteller unterschiedliche Größenangaben für in digitaler Form vorliegendes medizinisches Bildmaterial. In [Lehmenn et al. 2005] werden die in der Medizin üblichen Bild- und Videodaten basierend auf den heute geltenden Standards betrachtet. Der Artikel liefert eine gelungene Zusammenstellung hinsichtlich Speicherbedarf, Kompression und der damit verbundenen Datenübertragung.

Im Vergleich zu anderen bildgebenden Verfahren schneidet die Sonographie in puncto Bildgröße sehr gut ab. Ein Röntgenbild des Thorax wird mit 4000 x 4000 Bildpunkten bei einem Wertebereich von 10 Bit pro Pixel angegeben, während eine sonographische Abbildung mit 256 x 256 Bildpunkten bei einer Tiefe von 8 Bit pro Pixel auskommt.

Anhand des zur Verfügung stehenden Geräts konnten konkrete Werte ermittelt werden, die als Grundlage für die weitere Betrachtung bzw. Diskussion verwendet werden.

Anwendung des FAST-Algorithmus

Für eine realistische Abschätzung der benötigten Bandbreite wurden Testmessungen mit dem zur Verfügung gestellten mobilen Sonographiegerät durchgeführt. Bei der Durchführung der Untersuchung wurde zunächst eine vollständige FAST-Untersuchung nach Definition 3.2.1 durchgeführt, welche als Referenz dient.

Das FAST-Protokoll besteht aus fünf Schnittebenen im Bereich des Thorax und Abdomens, wobei nach freier Flüssigkeit und nach Rupturen einzelner Organe gesucht wird. Optional kann eine Echokardiographie durchgeführt werden. Die einzelnen Schnittebenen sind im Folgenden kurz zusammengefasst.

1. Schnittebene: Lateral-diaphragmaler Längsschnitt an der rechten Körperhälfte, wobei der Pleuraraum auf Hämatothorax und Pleuraerguss untersucht wird. Weiterhin wird subphrenisch nach freier abdomineller Flüssigkeit gesucht.

2. Schnittebene: Lateral-kaudaler Längsschnitt rechts. Subhepatisch wird auf freie abdominelle Flüssigkeit und Blutung bei Leberruptur untersucht. Freie abdominelle Flüssigkeit im Morison-Pouch und retroperitoneale Blutung wird diagnostiziert.

3. Schnittebene: Lateral-diaphragmaler Längsschnitt an der linken Körperhälfte. Der Pleuraraum und subphrenische Strukturen werden wie in der ersten Schnittebene betrachtet. Perisplenisch wird auf den Verdacht auf subkapsuläres Milzhämatom und Blutung bei Milzruptur untersucht.

4. Schnittebene: Lateral-kaudaler Längsschnitt links. Freie abdominelle Flüssigkeit im Koller-Pouch und retroperitoneale Blutung wird diagnostiziert.

5. Schnittebene: Medianer Unterbauch (quer/längs): Retro- und paravesikal wird auf freie Flüssigkeit untersucht.

6. Epigastrium (kardial-gerichtet): Verdachtsdiagnose auf Perikarderguss und Hämatoperikard.

Das Protokoll konnte in einem Zeitraum von 3 bis 5 Minuten durchgeführt werden.

3.4.3 Erzeugen des Datenstreams und Befundung

Das mobile Sonographiegerät verfügt über einen Composite-Video-Ausgang, der genutzt wurde, um die Videodaten an den Streamingserver weiterzuleiten. Das Sonographiegerät stellt auf dem lokalen Bildschirm neben den eigentlichen Sonodaten ein sogenanntes Overlay mit zusätzlichen Informationen für den schallenden Arzt dar. Bei der Übertragung werden sämtliche Bildinformationen, die auf dem Bildschirm des Sonographiegeräts

dargestellt werden, für die Übertragung genutzt. Es kommt also zu keiner Unterscheidung zwischen den Sonographiedaten, die von der Sonde aufgenommen werden, und dem zusätzlichen Overlay.

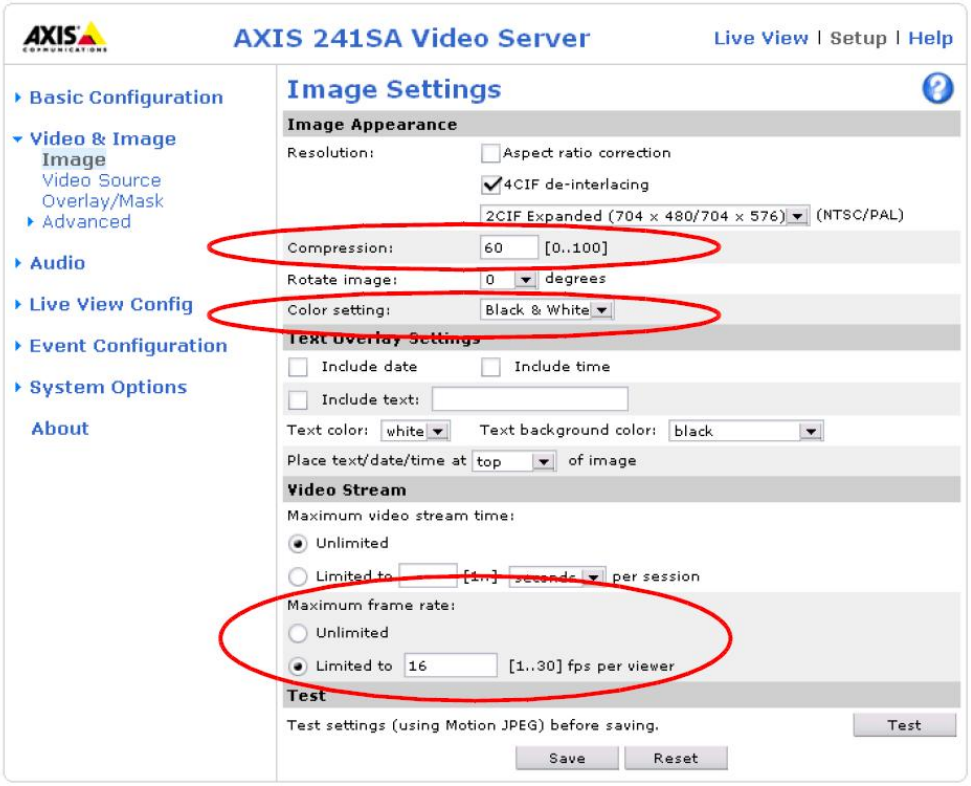


Abbildung 3.8: Konfiguration des Streaming-Servers

Danach wurde jeweils der zweite Abschnitt des FAST-Algorithmus genutzt, um die erforderliche Qualität der zu befundenden Daten zu ermitteln. Der Streaming-Server bearbeitet und überträgt die Daten an einen Rechner über Ethernet. Eine Echtzeitübertragung der Videodaten erfordert die Umsetzung in Form eines Videostreams. Aufgrund der gegebenen Voraussetzung, dass das Rettungspersonal am Unfallort nicht für die Untersuchung mit Ultraschallgeräten ausgebildet ist, erfordert die konkrete Untersuchung möglicherweise das Eingreifen des Expertenteams. Auch diese Bedingung kann nur erfüllt werden, wenn die Übertragung in Echtzeit erfolgt, d.h. die Daten als Stream übertragen werden.

Das Übertragungsmedium wird im Kernsystem, wie es in Abbildung 3.7 dargestellt ist, durch einen Streaming-Server der Firma *Axis* nachgebildet. Der Streaming-Server ist in der Lage, die vom Sonographiegerät generierten Videodaten in verschiedene Videostreams abzubilden. Dabei kann neben dem zu verwendenden Format auch die Qualität

gezielt gesteuert werden.

An dem Zielrechner werden die Daten sichtbar gemacht und aufgezeichnet, sodass das Ergebnis auf Befundbarkeit überprüft werden kann. Die Überprüfung der eingehenden Daten entspricht der Einzelteststellung „Expertenteam“ und wird von einem fachkundigen Arzt repräsentiert. Welche Parameter für die Übertragung variiert werden können, kann aus Abbildung 3.8 entnommen werden, die speziell auf die konfigurierbaren Parameter des verwendeten Streaming-Servers abstellt.

Aufgrund der Struktur der Daten wurde auf eine Kodierung mittels Motion-JPEG zurückgegriffen. Bei Motion-JPEG handelt es sich um ein Kompressionsverfahren für Videodaten. Anders als beim MPEG, bei dem Änderungen von Bild zu Bild berücksichtigt werden, wird beim Motion-JPEG jedes Einzelbild mittels JPEG komprimiert. Die Übertragung erfolgt bei einer konstanten Auflösung 704 x 476 (PAL). Da jedes Bild einzeln kodiert wird, ist es möglich, die Framerate sehr klein zu halten, da es keine Abhängigkeiten zwischen den einzelnen aufeinander folgenden Bildern gibt.

Dabei wurde in kleinen Schritten zunächst die Kompression variiert. In Abbildung 3.8 kann man die möglichen Einstellungen des Streaming-Servers hinsichtlich Kompression erkennen. Es kann ein Motion-JPEG-Kompressionsfaktor von 0 bis 100 gewählt werden. Hier kann als Resultat festgehalten werden, dass bei einem Kompressionsfaktor von 10 das Bild noch über genügend Detailauflösung verfügt, um eine Befundung vorzunehmen.

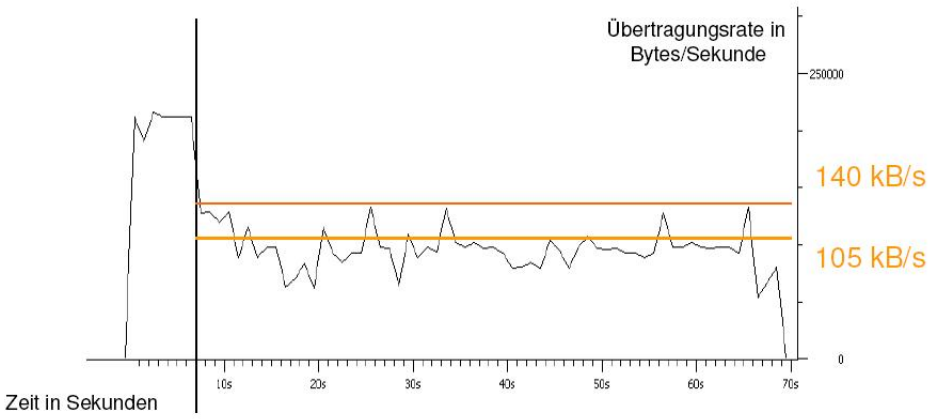


Abbildung 3.9: Übertragung einer FAST-Untersuchung bei einer Motion-JPEG-Komprimierung von 10 und einer Framerate von 10 Bildern pro Sekunde

Als zweiter Parameter wurde die Framerate ebenfalls zunächst unabhängig von ande-

ren Parametern betrachtet. Die Framerate gibt an, wie viele Einzelbilder pro Sekunde übertragen werden. Hierbei ist festzuhalten, dass für die Dokumentation eines positiven Befundes ein einzelnes Bild bereits ausreichend sein kann. Auf diesem Bild müssen allerdings sämtliche Merkmale, die für den Befund gesorgt haben, klar erkennbar sein.

Um den Befund zu erstellen, werden bedeutend mehr Bilder benötigt, da die gesuchten Merkmale aus den Einzelbildern extrahiert werden müssen. Hierfür muss die Umgebung der zu befundenden Stelle unter verschiedenen Ansichten beurteilt werden. Dies geschieht, indem der Schallkopf über den zu untersuchenden Bereich geführt wird. Eine Framerate von 10 Bildern pro Sekunde wurde als akzeptabel festgehalten.

Abschließend wurden die Grenzwerte der beiden Parameter gleichzeitig bei der Untersuchung angewandt und festgestellt, dass bei einer gleichzeitigen Anwendung die Bildqualität im Vergleich zu den Einzelanwendungen nicht in Mitleidenschaft gezogen wird. Die grafische Aufarbeitung der „minimalen“ befundbaren Übertragung ist in Abbildung 3.9 aufgetragen.

An der Grafik lässt sich der praktische Versuchsaufbau ablesen. So wurde die Sonde zuerst mit Ultraschall-Gel bestrichen, wodurch das Rauschen bei Beginn der Schallung zu erklären ist. Nach dieser Einschwingphase konnte eine konstante Übertragung durchgeführt werden. Im Durchschnitt ist eine Bandbreite von 105 kB/s ausreichend. Sollen auch die auftretenden Spitzen berücksichtigt werden, muss eine Bandbreite von mindestens 140 kB/s zur Verfügung stehen. Basierend auf diesen Richtwerten kann die Untersuchung des notwendigen bzw. zur Verfügung stehenden Übertragungsmediums durchgeführt werden.

Übertragungsmedium

Ein weiteres wichtiges Kriterium für die Simulation des Übertragungsmediums ist sicherlich das verwendete Netz und die zur Verfügung stehende Bandbreite. Als Übertragungsmedien zwischen dem Rettungswagen und der Klinik bieten sich UMTS-Netze bzw. eine Übertragung per Satellit an. Benötigt wird hier entsprechende Hardware und eine Netzfreischaltung bzw. eine Emulation der Übertragungsmedien. Die Infrastruktur der Klinik wird durch entsprechende Netzwerkgeräte modelliert.

Netzschwankungen und Netzausfälle können durch Softwaretools, wie den in Abbildung 3.7 vorgesehene Netzwerkemulator NistNet, emuliert werden.

UMTS (Universal Mobile Telecommunication System) ist ein Mobilfunkstandard der dritten Generation. Es gibt unterschiedliche Verfahren zur Übertragung der Daten. Das in Deutschland am weitesten verbreitete ist das FDD-Verfahren (Frequency Division Duplex). Bei diesem Frequenzmultiplex-Verfahren findet die Kommunikation zwischen Endgerät und Basisstation in zwei verschiedenen Frequenzbereichen statt. Die Kommunikation von der Basisstation zum Endgerät wird als *Downlink* bezeichnet. Die Gegen-

richtung, sprich die Kommunikation vom Endgerät zur Basisstation wird über den sogenannten *Uplink* realisiert. Bei FDD stehen jeweils ein Frequenzbereich für den Uplink und für den Downlink zur Verfügung. Als maximale Übertragungsraten sind 384 kBit/s im Downlink und 128 kBit/s im Uplink-Bereich angegeben.

In Release 6 des UMTS-Standards sind Erweiterungen vorgesehen, die die Übertragungsraten erheblich vergrößern. So sind mit *HSDPA* (High Speed Downlink Packet Access) unter optimalen Bedingungen Downlink-Raten von bis zu 14,4 MBit/s und mit *HSUPA* (High Speed Uplink Packet Access) Uplink-Raten von zunächst 1,4 MBit/s und später von bis zu 5,8MBit/s zu erzielen. Die Einführung einer HSDPA mit einer Übertragungsrate von maximal 7,2 MBit/s soll ab dem Jahre 2007 beginnen.

Ein weiteres Verfahren zur Übertragung der Daten ist das *TDD*-Verfahren (Time Division Duplex). Dieses Zeitmultiplex-Verfahren erreicht Downlink-Raten von bis zu 1920 kBit/s. Da in Deutschland zur Zeit nur ein Netzbetreiber dieses Verfahren anbietet – und dies auch nur in einigen Großstädten – wird dieses Verfahren hier nicht weiter betrachtet.

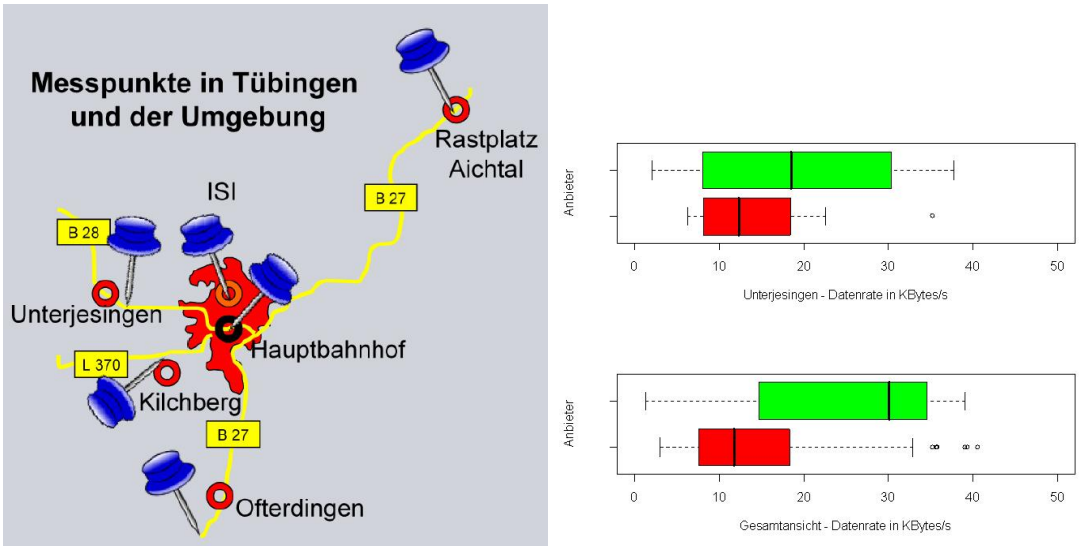
Feldversuch – UMTS-Abdeckung in der Region Tübingen

Es wurde ein Feldversuch mit dem Ziel, die heutige UMTS Abdeckung in der Region um Tübingen zu ermitteln, durchgeführt. Hierzu wurde an den Bundesstraßen, welche durch Tübingen führen, jeweils dieselben Testmessungen vorgenommen. Ein Großteil der Notfälle ereignen sich im Straßenverkehr, sodass für einen praktischen Einsatz des vorgestellten Szenarios eine Abdeckung auf den Schnell- und Bundesstraßen als erforderlich angesehen wird. Welche Messpunkte konkret gewählt wurden, kann an Abbildung 3.10a abgelesen werden.

Für die Messung wurde auf ein Standardtool zurückgegriffen. Mit *ttcp*, einem freien Tool, welches unter [Muss] erhältlich ist, kann der Durchsatz einer TCP/IP-Verbindung ermittelt werden. Hierzu wird gemessen, wie lange ein großer Block an Daten benötigt, um vollständig über die Testleitung übertragen zu werden. Das Ergebnis wird häufig als *Bulk-Transfer-Kapazität* des Endpunkts bezeichnet.

Für die Messung standen zwei UMTS-Karten verschiedener Anbieter zur Verfügung, sodass neben der reinen Netzabdeckung auch überprüft werden konnte, inwieweit eine vom Hersteller unabhängige Aussage möglich ist.

Eine wichtige Kenngröße ist die Datenrate beim Uplink, da sie Aufschluss darüber geben kann, ob die Sonographiedaten, deren Größe im vorherigen Teilabschnitt abgeschätzt wurde, überhaupt per UMTS übertragen werden können. An jedem in Abbildung 3.10a vermerkten Messpunkt wurde deshalb dieselbe Datenmenge an einen zentralen Server, der das Klinikum repräsentiert, gesendet. Als zentraler Punkt wurde das ISI-Tübingen gewählt.



(a) Messpunkte in Tübingen und Umgebung

(b) Datenraten

Abbildung 3.10: Feldversuch – UMTS-Abdeckung

Das ttcp-Tool besteht aus zwei Komponenten, aus einem Server, zu dem die Daten gesendet werden, und einem Client, von dem der Transport ausgeht. Der Server wurde im ISI installiert und gestartet. Für die Übertragung wurden die Länge der einzelnen Datenbulks fest vorgegeben, die Anzahl der pro Sekunde zu sendenden Bulks wurde dem entgegen langsam sukzessiv erhöht, um an die *Grenzen* des Systems zu stoßen.

Aus den Versuchsreihen wurden zwei repräsentative Auswertungen ausgewählt und in Abbildung 3.10b in Form eines Boxplots (vgl. Abschnitt 8.5.1) dargestellt. Die obere Grafik zeigt die erzielten Datenraten beim Messpunkt *Unterjesingen*, während die untere Grafik sämtliche Messwerte als Grundlage berücksichtigt. Die Grafiken wurden skaliert, da es zu einigen einzelnen Ausreißern mit teilweise hohen Werten gekommen ist. Als Farben wurde den Anbietern Vodafon, die Farbe Rot, und E-Plus, die Farbe Grün, zugewiesen.

Als Ergebnis für die Betrachtung der möglichen Datenrate kann Folgendes festgehalten werden. Eine Datenrate von 35 KByte pro Sekunde sollte erzielt werden können, wobei im Versuch die Vodafon-Karte deutlich niedrigere Ergebnisse geliefert hat. Von einer so hohen Datenrate kann aber nicht zwingend ausgegangen werden, es wurden auch Übertragungen durchgeführt, bei denen ein Wert von 20 KByte/s nicht übertroffen werden konnte.

Es bleibt festzuhalten, dass eine Übertragung mit lediglich einer UMTS-Karte nicht ausreicht, um eine Übertragung der Sonographiedaten in ausreichender Qualität zu gewähr-

leisten. Eine Alternative wäre die parallele Übertragung mit mehreren Karten, wobei dann eine Synchronisation der Daten beim Empfänger erfolgen muss. Einen Beitrag hierzu könnte das im zweiten Abschnitt entwickelte Multi-Pfad-Szenario für SCTP leisten. An dieser Stelle sei aber auch auf den neuen Standard HSUDA verwiesen, der den notwendigen Durchsatz bereitstellen wird.

Eine weitere Fragestellung bezog sich auf die Herstellerunabhängigkeit. Es wurden lediglich zwei verschiedene Anbieter im Versuch berücksichtigt, allerdings sind die stark voneinander abweichenden Ergebnisse verwunderlich. Man kann feststellen, dass die Übertragung über die Vodafon-Karte zwar deutlich konstanter abgelaufen ist, aber die durchschnittliche Datenrate nicht an das Konkurrenzprodukt heranreichte. Dieses Ergebnis ist natürlich nicht repräsentativ und wird sicherlich in anderen Gegenden mit umgekehrten Vorzeichen festzustellen sein.

Für Tübingen besteht eine im Verhältnis für die Größe der Stadt gute Netzabdeckung. Es wurden für den Feldversuch nur Stationen angefahren, die vom Hersteller zumindest als erreichbar klassifiziert waren. Allerdings konnte am Messpunkt *Kollberg* mit der E-Plus-Karte aufgrund der nicht ausreichenden Signalstärke keine Verbindung hergestellt werden. Ein solcher Ausfall wäre im Falle eines echten Notfalls nicht akzeptabel, da nur bei einem konsequent funktionierenden System mit der Akzeptanz des medizinischen Personals gerechnet werden kann.

Es bleibt festzuhalten, dass ein Ausbau der UMTS-Netze für professionelle Anwendungen unbedingt erforderlich ist.

3.4.4 Ausblick

Ein Ausbau der UMTS-Netze für professionelle Anwendungen ist unbedingt erforderlich. In der Übergangszeit könnte hier die Möglichkeit des parallelen Versands über mehrere Pfade Abhilfe schaffen. Durch die Kombination von verschiedenen UMTS-Anbietern könnte das Problem der Ressourcenteilung zwischen den einzelnen Nutzern reduziert werden. Welche Vorteile sich daraus ergeben, hängt von sehr vielen Faktoren ab, die eine gesonderte Untersuchung erfordert.

Als Ergebnis der vorherigen Unterabschnitte kann festgehalten werden, dass unter bestimmten Bedingungen eine Realisierung des erweiterten Notfallszenarios unter Anwendung bestehender technischer Möglichkeiten grundsätzlich bereits heute möglich ist. Der komplette Testaufbau des Basisprojekts ist in Abbildung 3.7 zusammengestellt.

Mit den zusätzlichen Komponenten ist man in der Lage, die UMTS-Sendeeinheit, die mehrere verschiedene UMTS-Karten aufnehmen kann, fest und witterungsunabhängig am Rettungswagen zu befestigen. Über die Router kann auch die geforderte Verschlüsselung der Daten garantiert werden, da hier bereits herstellerseitig gängige Kryptosysteme eingebaut wurden.

Der bestehende Testaufbau wurde auch für andere Arbeiten genutzt. So wurden in der Diplomarbeit [Roll 2008], basierend auf dem bestehenden Testnetz, Untersuchungen zur sicheren Übertragung von Multimedia-Daten durchgeführt.

4 Workflowaspekte

Hochkomplexe Behandlungskonzepte wie die *Polytraumaversorguung* im Allgemeinen und die *präklinische Polytraumaversorgung* im Speziellen unterliegen festen Regeln. Die Abläufe sind vorgeschrieben und werden trainiert, sodass bei einem zeitkritischen Notfall der Ablauf routiniert erfolgen kann. Um die präklinische Sonographie erfolgreich in diesen Ablauf zu integrieren, müssen Prozessveränderungen angegeben und umgesetzt werden.

Medizinische Prozesse oder auch *medizinische Workflows* werden üblicherweise in Form von Algorithmen beschrieben, sodass im folgenden Abschnitt die Veränderungen der Prozessabläufe durch Anwendung der *präklinischen Sonographie* erarbeitet werden. Hierfür werden im Abschnitt 4.1 die notwendigen Grundlagen besprochen und die notwendigen Begriffe definiert.

4.1 Grundlagen medizinischer Algorithmen

Der ursprünglich aus der Mathematik bzw. Informatik stammende Begriff des Algorithmus wird u.a. in [Kanz et al. 2002] auf die Anwendung in der Medizin spezialisiert.

Definition 4.1.1 (Medizinischer Algorithmus) *Ein* Algorithmus *im medizinischen Sinn ist eine festgelegte Anweisung zur systematischen Behandlung von Patienten, wobei die schrittweise Anordnung der Entscheidungsknoten prioritätenorientiert erfolgt. Der Algorithmus legt den Zeitpunkt und die logische Abfolge der Behandlung fest.*

Der Begriff des Algorithmus wird in der Medizin auch für die Darstellung der Abläufe in Form von Prosatext verwendet. In [Waydhas et al. 1997] werden die Vor- und Nachteile der Prosadarstellung gegen die aus der Informatik übliche Diagramm-Repräsentation abgegrenzt. Für die folgende Integration der präklinischen Sonographie in die bestehenden Arbeitsabläufe wird zur Beschreibung auf die grafische Darstellung in Form von Strukturdiagrammen zurückgegriffen.

Algorithmen kommen nach [Waydhas et al. 1997] im medizinsischen Umfeld in drei unterschiedlichen Bereichen zum Einsatz. Ein Bereich wird durch die Lehre abgedeckt. *Algorithmen in der Lehre* ermöglichen es dem Lernenden, das vermittelte Grundwissen zu ordnen und zu strukturieren und so das konkrete medizinische Handeln zu vereinfachen. Ein weiterer wichtiger Aspekt ist die Anwendung von *Algorithmen zur Prozessevaluierung und zum Qualitätsmanagement*. Algorithmen dienen hierbei als Grundlage

für die Bewertung der *Prozessqualität*. Die Vorteile des Qualitätsmanagements auf Basis von Algorithmen wurden bereits in den Sechziger Jahren (vgl. [Donabedian 1966]) hervorgehoben.

Der *Sollwert*, also das erwartete Handeln, wird durch den Algorithmus eindeutig wiedergegeben. Dies entspricht dem erwarteten Verlauf der Behandlung. Der tatsächliche Verlauf oder auch *Istwert* kann direkt an den Verzweigungspunkten verglichen werden. In [Buchholz et al. 1994] konnte anhand der Qualitätsanalyse der Schockraumbehandlung gezeigt werden, wie Managementprobleme Einfluss auf die Behandlung haben. Durch die Auswertung der Ist- und Sollwerte kann eine mögliche Ursache des Problems gefunden werden, was wiederum die Grundlage für die Lösungsfindung darstellt.

Diese Algorithmenaspekte werden in der voliegenden Arbeit nicht weiter verfolgt. Im Mittelpunkt steht die Verwendung des Algroithmenbegriffs unter dem Gesichtspunkt *Algorithmen als Handlungsleitlinien*. Hierfür werden Algorithmen standardisiert und als vorgeschriebene Handlungsleitlinien durch Institutionen festgelegt. Im Folgenden wird die Spezifikation der *Deutschen Gesellschaft für Unfallchirurgie*, hier die Arbeitsgemeinschaft Notfallmedizin und der *Deutschen Gesellschaft für Chirurgie* speziell die Arbeitsgemeinschaft Notfall- und Intensivmedizin verwendet, da hier das präklinische Polytrauma Management angesiedelt ist.

Im folgenden Abschnitt 4.2 wird speziell auf die Spezialisierung der Algorithmen für traumatologische Notfälle eingegangen. Vorher werden die Konzepte der medizinischen Algorithmen kurz dargestellt.

Die *Charakteristika von Algorithmen* können nach [Lackner und Kanz 1996] wie folgt zusammengefasst werden:

Algorithmen

- *bilden* einheitliche Behandlungsleitlinien
- *gestatten* begründete Abweichungen
- *zerlegen* komplexe Probleme in Einzelschritte
- *zeigen* einen strukturierten Lösungsweg auf
- *vermitteln* trotz Zeitdruck Sicherheit
- *machen* Behandlungsabläufe transparent

Die Entwicklung von Algorithmen kann in fünf Phasen erfolgen. In der Phase I wird der Prozessablauf analysiert und das bzw. die Probleme identifiziert. Nachdem das Problem lokalisiert und beschrieben wurde, kann in Phase II die Entwicklung von Handlungsleitlinien vorgenommen werden. Hierzu müssen Maßnahmen, Prioritäten, Kriterien und

Handlungsabläufe festgelegt und beschrieben werden. Diese Phase mündet in der Formulierung eines Algorithmus.

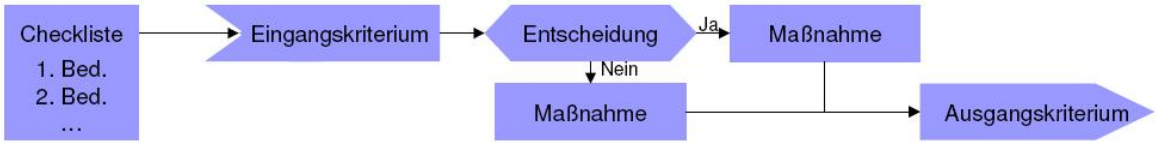


Abbildung 4.1: Strukturelemente von medizinischen Algorithmen

Zur Formulierung werden Strukturelemente ähnlich wie in der Informatik üblich verwendet. Eine Übersicht in Form eines Beispieldiagramms ist in Abbildung 4.1 dargestellt, an dem neben den einzelnen Strukturelementen auch ihre Funktion im Diagramm ersichtlich wird.

Den Einstieg in ein Algorithmus-Diagramm erfolgt immer über eine *Checkliste*. Eine Checkliste ist eine Sammlung von Bedingungen, die erfüllt sein müssen, um in einen Teilalgorithmus einsteigen zu dürfen. Somit dienen Checklisten dazu, die Übersichtlichkeit auch bei umfangreichen unübersichtlichen Abläufen zu gewährleisten. Die Granularität der Bedingungen ist dabei divergent, sodass neben umgangssprachlichen Bezeichnungen wie z.B. „Sturz aus mehr als 5 m Höhe“ auch wissenschaftliche Termini wie „Glascow Coma Scale <= 10“ verwendet werden. Es müssen für die Verwendung des Teilalgorithmus nicht sämtliche Bedingungen erfüllt sein, es reicht, wenn eine Bedingung zutrifft.

Je nach Zeitrahmen unterscheidet man zwischen *Arbeitsdiagnosen* und Verdachtsdiagnosen. Je nach Dringlichkeit wird der Ablauf in verschiedene *Zeitphasen* bzw. *Zeitrahmen* eingeteilt. Jede Zeitphase wird nach der Nomenklatur des internationalen Merkwortalphabets gezählt und trägt den unterschiedlichen therapeutischen Prioritäten Rechnung. So gibt es die Phasen Alpha, Bravo, Charlie etc. Die jeweilige Diagnose dient als *Eingangskriterium*. Der weitere Verlauf des Algorithmus ist durch eine Aneinanderreihung von *Entscheidungen* und *Maßnahmen* gegeben. Ein *Ausgangskriterium* bildet immer den Abschluss eines Teilalgorithmus, welches den weiteren Handlungsbedarf festlegt. Dies kann wiederum ein eingeschobener Teilalgorithmus, wie beispielsweise die Reanimation, aber auch die Überführung in eine neue Gesamtsituation, wie beispielsweise die Überführung in den Schockraum, darstellen.

Durch eine solche Strukturierung können Maßnahmen, Prioritäten, Kriterien und der Handlungsablauf vollständig abgebildet werden. Nach der Formulierung eines Algorithmus erfolgt in der Phase III die klinische Umsetzung. Den Abschluss bildet die Phase IV, in der durch kontinuierliche Validierung und Revision des Algorithmus das Qualitätsmanagement einsetzt.

4.2 Versorgung bei Polytrauma

Die Versorgung bei Polytrauma kann in zwei Szenarien eingeteilt werden. In der Schockraum-Phase befindet sich der Patient bereits im Klinikum und kann im Schockraum behandelt werden. In Teilabschnitt 4.2.2 wird der hierfür standardmäßig verwendete Algorithmus erläutert. Als weiteres Standard-Szenario liegt die Versorgung am Unfallort vor. Der Algorithmus bei solchen traumatologischen Notfällen wird in Abschnitt 4.2.1 zusammengestellt. Die Integration der mobilen Sonographie, wie sie bereits in Abschnitt 3.2 erläutert wurde, wird dann in Abschnitt 4.2.4 durchgeführt.

4.2.1 Präklinisches Polytrauma Management – Teil I

Nach [Beck et al. 2002] ist das erste und entscheidende Glied in der Rettungskette die schnelle und kompetente Versorgung des Verunfallten am Unfallort. Beck differenziert nicht nach der Schwere der Verletzungen, sondern hält die Versorgung am Unfallort sowohl für Einzelverletzungen als auch für Polytraumatisierte für erforderlich. Dies führt zum Begriff des Polytraumas.

Definition 4.2.1 (Polytrauma) *Nach [Ziegenfuss 1998] versteht man unter einem* Polytrauma *Verletzungen mehrerer Körperregionen oder Organe, von denen mindestens eine oder die Kombination mehrerer lebensbedrohlich sind.*

Dieser Begriff ist noch sehr vage, kann aber über verschiedene Scoringsysteme präzisiert werden. Zwei Scoringsysteme haben sich in der Literatur durchgesetzt. Der *Revised-Trauma Score* oder kurz *RTS* und das *Injury Severity Score* abgeküzt als *ISS*. Das RTS erfasst die Auswirkungen des Traumas aufgrund von Störungen der wichtigsten physiologischen Systeme. Das ISS dagegen ist anatomisch orientiert. Der Schweregrad der Gesamtverletzung wird dabei auf Basis der drei am schwersten verletzten Körperregionen abgeschätzt. Für jede einzelne Körperregion und Schwere der Verletzung wird nach einem Punktsystem die Gesamtverletzung bestimmt. Mit dem ISS kann der Begriff des Polytraumas erweitert werden. Ein Polytrauma liegt demnach dann vor, wenn der ISS bei einem Wert größer als 15 liegt.

Als besonders schwerwiegend wird die Verletzung der Abdomen gewichtet, wodurch das *Abdominaltrauma* als besonders schwerwiegend klassifiziert wird. Gerade bei diesen schwerwiegenden Fälle setzt die präklinische Sonographie an. Im klassischen Algorithmus steht daher bei einem Abdominaltrauma der dringliche Abtransport in ein entsprechend ausgerüstetes Krankenhaus im Vordergrund (vgl. [Ziegenfuss 1998]).

Im präklinischen Polytrauma-Management greift man auf die jahrelangen Erfahrungen zurück, die man durch Anwendung der Schockraumphase gewonnen hat. Daher wird als Einschub kurz auf die Algorithmen im Schockraum eingegangen.

4.2.2 Schockraum-Algorithmus

Bevor auf die Verwendung der präklinischen Sonographie im Speziellen eingegangen werden kann, wird der klassische Algorithmus zur präklinischen Traumaversorgung besprochen. Die Zeitphasen wurden im einleitenden Abschnitt 4.1 bereits zur Gliederung der therapeutischen Prioritäten eingeführt. Für die präklinische Traumaversorgung sind vier verschiedene Prioritäten, also vier verschiedene Zeitphasen, definiert.

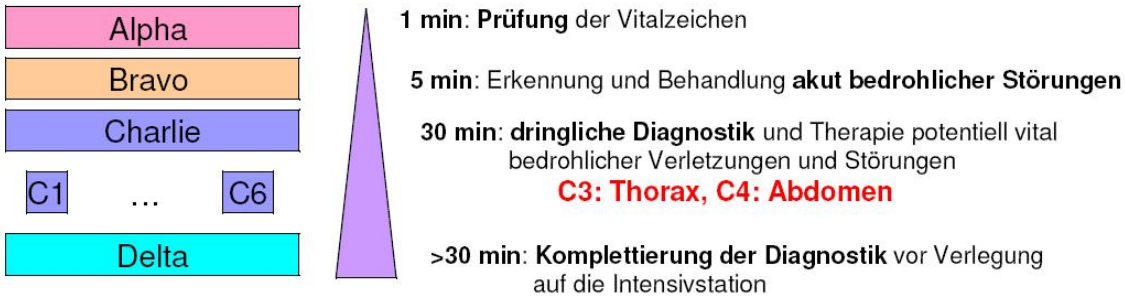


Abbildung 4.2: Schockraum-Algorithmus der Polytraumaversorgung – Zeitphasen

Die verschiedenen Zeitphasen für die Behandlung von polytraumatisierten Patienten im Schockraum sind in Abbildung 4.2[1] schematisch dargestellt. Für die erste Phase *ALPHA* ist lediglich eine Minute vorgesehen. Hier geht es darum, die *Vitalzeichen* oder Lebenszeichen des Patienten zu prüfen. Die grundlegenden Vitalzeichen sind das Bewusstsein, die Atmung sowie der Kreislauf. Sollten die Vitalzeichen fehlen, ist sofort zu handeln. Vorgeschlagene Maßnahmen: ABC-Reanimation, Atemwege freimachen, Beatmung, Sauerstofftherapie, Kreislauftherapie etc.

Die zweite Phase *BRAVO* ist auf ca. 5 Minuten ausgelegt und hat zum Ziel, die Behandlung und Erkennung von akut vital bedrohlichen Störungen einzuleiten. In [Waydhas et al. 1997] werden zu den akuten vital bedrohlichen Störungen die Instabilität der Halswirbelsäule, massive externe Blutungen, Hypoxämie – Sauerstoffmangel im Blut – und Schock gezählt.

Die folgende Phase *CHARLIE* ist mit 30 Minuten und sechs parallel verlaufenden Algorithmen besonders komplex. In dieser Phase gilt es, den sogenannten *Check-Up* durchzuführen, also die notwendige und dringliche Diagnostik und Therapie von potenziell vital bedrohlichen Verletzungen. Die bereits erwähnten 6 Teilalgorithmen können mit folgenden Schlüsselworten beschrieben werden:

C1 Atmung

[1] nach [Waydhas et al. 1997]

C2 Kreislauf

C3 Thorax

C4 Abdomen

C5 ZNS und Schädel

C6 Bewegungsapparat

Die Algorithmen C3 und C4 sind für unser Szenario von besonderer Bedeutung, da hier auf den Einsatz der Sonographie zwingend zurückgegriffen werden muss. Die Radiologie führt in dieser Phase zwingend folgende Untersuchungen aus: CT des Schädels, CT-HWS, CT des Thorax, CT der Abdomen, CT des Beckens. Diese Bereiche gilt es, beim Einsatz der präklinischen Sonographie zu berücksichtigen.

In der Phase *DELTA* wird die Diagnostik vor der Verlegung auf die Intensivstation komplettiert. Diese Phase ist zeitlich nicht terminiert.

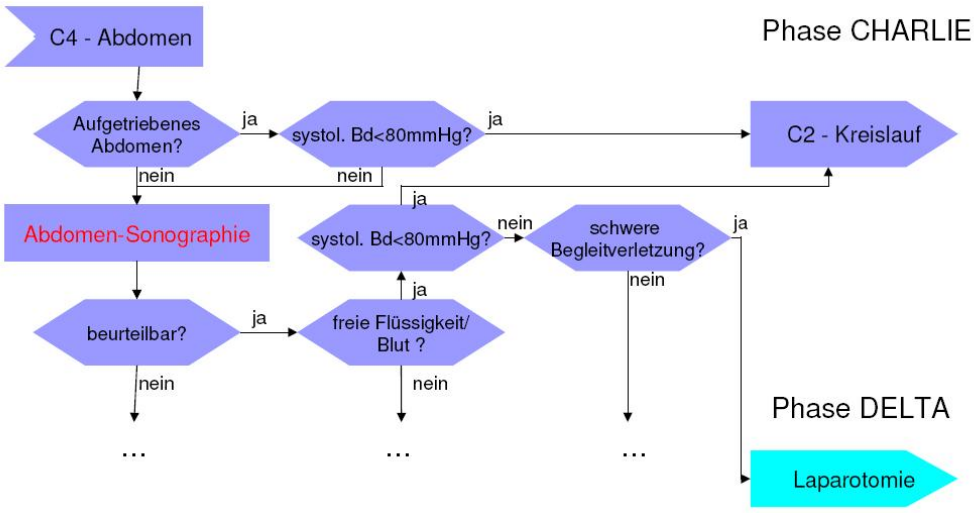


Abbildung 4.3: Zeitphase Charlie und Delta – Schockraumalgorithmus (verkürzte Darstellung)

Die Abbildung 4.3[2] stellt einen Auszug aus dem Algorithmus C4-Abdomen der Phase *CHARLIE* dar.

Die wichtige Abdomen-Sonographie ist hervorgehoben. Ausgehend von dieser Betrachtung kann die präklinische Phase wieder aufgegriffen werden.

[2]Quelle: Deutsche Gesellschaft für Unfallchirurgie – Arbeitsgemeinschaft Notfallmedizin und Deutsche Gesellschaft für Chirurgie – Arbeitsgemeinschaft Notfall- und Intensivmedizin

4.2.3 Präklinisches Polytrauma Management – Teil II

Über den Umfang der direkt an der Unfallstelle durchgeführten Maßnahmen besteht keine einheitliche Meinung, daher haben sich zwei gegensätzliche Strategien entwickelt. In Nordamerika wird nach [Beck et al. 2002] die sogenannte *Scoop-and-Run-Strategie*, in [Peters and Runggaldier 2005] auch als *Load-and-Go-Strategie* bezeichnet, bevorzugt eingesetzt.

Definition 4.2.2 (Scoop-and-Run-Strategie) *Unter der* Scoop-and-Run-Strategie *versteht man den schnellen Transport ohne aufwendige ärztliche Primärversorgung.*

In Europa hingegen wird die alternative *Stay-and-Play-Strategie* als die effektivere betrachtet.

Definition 4.2.3 (Stay-and-Play-Strategie) *Bei der* Stay-and-Play-Strategie *erfolgt die primäre Stabilisierung des Patienten am Unfallort. Die mitunter länger dauernde Versorgung des Patienten berücksichtigt nahezu jede therapeutische Variante.*

Die Stay-and-Play-Strategie erfordert vom Rettungspersonal ein hohes Maß an Kompetenz, sodass die Erstversorgung schnell und fehlerlos durchgeführt werden kann. Speziell für dieses Szenario wird der Notarzt durch immer bessere und vor allem kleinere Technik unterstützt. So fallen die Geräte zur mobilen Sonographie in diese Kategorie, die aber derzeit aus den im Abschnitt 2.2 ausführlich betrachteten Gründen noch nicht zum Einsatz kommt bzw. sich noch in der Testphase befindet. Bei der weiteren Betrachtung wird daher auf die Stay-and-Play-Strategie abgestellt.

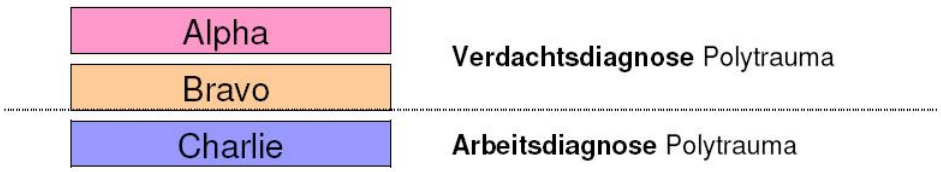


Abbildung 4.4: Präklinisches Polytrauma-Management – Zeitphasen

Das präklinische Polytrauma-Management, wie es in Abbildung 4.4 abgebildet ist, zerfällt in die drei Zeitphasen *ALPHA*, *BRAVO* und *CHARLIE*. Nach [Kanz et al. 2002] wird das präklinische Polytrauma-Management als Behandlungskonzept beschrieben, das nicht nur den Gesamtprozess definiert, sondern auch die einzelnen Entscheidungen und Versorgungsschritte berücksichtigt.

In Abbildung 4.4 werden die Phasen *ALPHA* und *BRAVO* unter dem Einstiegspunkt *Verdachtsdiagnose Polytrauma* zusammengefasst. Die Verdachtsdiagnose Polytrauma umfasst die lebensrettenden Sofortmaßnahmen am Unfallort. Der Maßnahmenkatalog enthält

daher u.a. Maskenbeatmung, Notfallthoraxdekompression, Sauerstoffabgabe etc.

Zur Veranschaulichung werden die verwendeten Checklisten angegeben, die konkreten Algorithmen können in [Kanz et al. 2002] nachgelesen werden. Für diese Phasen stehen drei Checklisten zur Verfügung. Die einführende Checkliste mit der Bezeichnung *Unfallmechanismus* stellt auf die Ursache der Verletzung ab. Folgende Punkte sind erfasst:

- Sturz aus mehr als 5 m Höhe
- Explosionsverletzungen
- Herausschleudern aus dem Fahrzeug
- Einklemmung und Verschüttung
- Verkehrsunfall eines Fußgängers oder Fahrradfahrers,
- Motorrad- oder Kraftfahrzeugunfälle mit höherer Geschwindigkeit
- Tod des Beifahrers

Dies bildet den Einstieg in die Verdachtsdiagnose Polytrauma. Die zweite Checkliste greift nach erfolgreicher Abarbeitung des ersten Teilalgorithmus. Hier wird bei Vorliegen eines Spannungspneumothorax die Dekompression des Pleuraraumes durchgeführt. Die Checkliste sieht folgendermaßen aus:

- fehlende Atemgeräusch und zusätzliches Vorliegen von
  - gestauten Halsvenen
  - einer schweren Dyspnoe
  - eines hohen Beatmungsdrucks
  - einer Atemfrequenz von < 10 oder $> 29/min$ oder eines systolischen Blutdrucks $< 80\ mmHg$
  - Herzrhythmusstörungen und EKG-Veränderungen

Als dritter Teilschritt wird die Checkliste „Dringliche Intubation“ abgearbeitet. Damit ist die Phase *BRAVO* abgeschlossen.

In einem zweiten Abschnitt werden funktionserhaltende Maßnahmen eingeleitet. Dies ist Aufgabe der Phase *CHARLIE*, die mit der *Arbeitsdiagnose Verletzungsmuster* einsetzt. Der Einsatz der mobilen Sonographie kann erst nach Abschluss der Phasen *ALPHA* und *BRAVO* erfolgen, wenn sämtliche lebensrettenden Sofortmaßnahmen abgeschlossen sind. Ein Auschnitt aus dem Algorithmus für das präklinische Polytrauma-Management in der Phase *CHARLIE* ist in Abbildung 4.5 zu sehen. Als Ausgangskriterium wird auf den Schockraum verwiesen, der wiederum einen Abtransport in die Klinik voraussetzt.

Im folgenden Abschnitt 4.2.4 wird der hier kurz beschriebene Algorithmus der Phase *CHARLIE* in Hinblick auf die präklinische Sonographie untersucht.

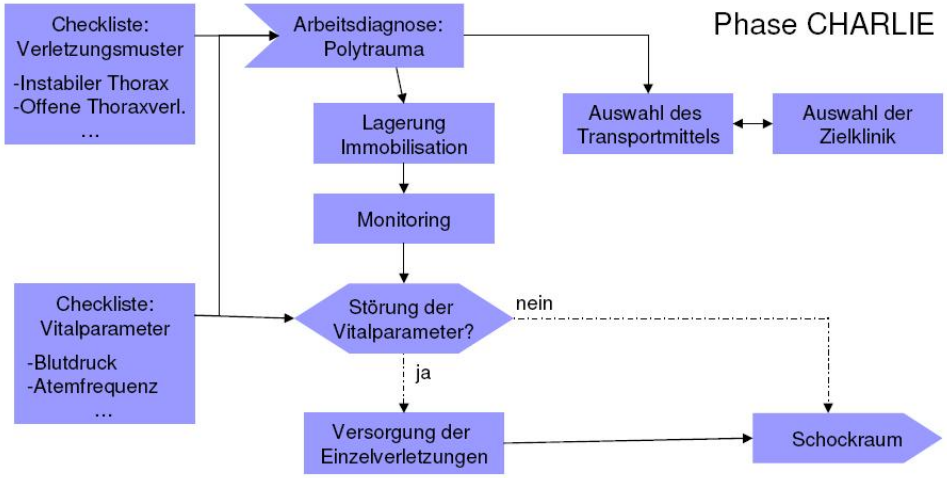


Abbildung 4.5: Zeitphase Charlie – präklinisches Polytrauma Management (verkürzte Darstellung)

4.2.4 Erweitertes präklinisches Polytrauma Management

Nachdem die bestehenden und anerkannten Algorithmen für den Schockraum und das präklinische Polytrauma Management dargestellt wurden, kann in diesem Abschnitt auf die Erweiterung der Algorithmen eingegangen werden. Durch den Einsatz von mobiler Sonographie, wie er in unserem Telemedizin-Szenario vorgesehen ist, sind folgende Änderungen im Ablauf denkbar.

Eine Möglichkeit der Integration der mobilen Sonographie in den Algorithmus des präklinischen Polytrauma-Managements ist in Abbildung 4.6 abgebildet. Ausgehend von der Frage. „Was soll durch die bereits im Vorfeld durchgeführte Sonographie erreicht werden?“, kann die Erweiterung beschrieben werden.

Das interdisziplinäre Polytraumamanagement wurde u.a. in einer mehrteiligen Artikelserie der Zeitschrift *Notfall- und Rettungsmedizin* untersucht. Der Teil 2 der Serie (vgl. ([Culemann et al. 2003]) untersucht die Klinikaufnahme vital bedrohter traumatisierter Patienten. Die Autoren gelangen zu der Auffassung, dass die Wahl des geeigneten Krankenhauses einen entscheidenden Einfluss auf die Überlebenschance des Patienten hat. Es wird ausgeführt, dass aus der Darstellung der Diagnostik- und Behandlungsabläufe bei der Aufnahme von polytraumatisierten Patienten, wie sie im o.a. Artikel ausführlich besprochen wurden, diese sich rückwirkend auf die Rettungskette auswirken und sich so Kriterien für die Zuweisung der Patienten ergeben. Die folgende Behandlung im Schockraum des Klinikums muss demnach bereits bei der notärztlichen Versorgung am Unfallort berücksichtigt werden.

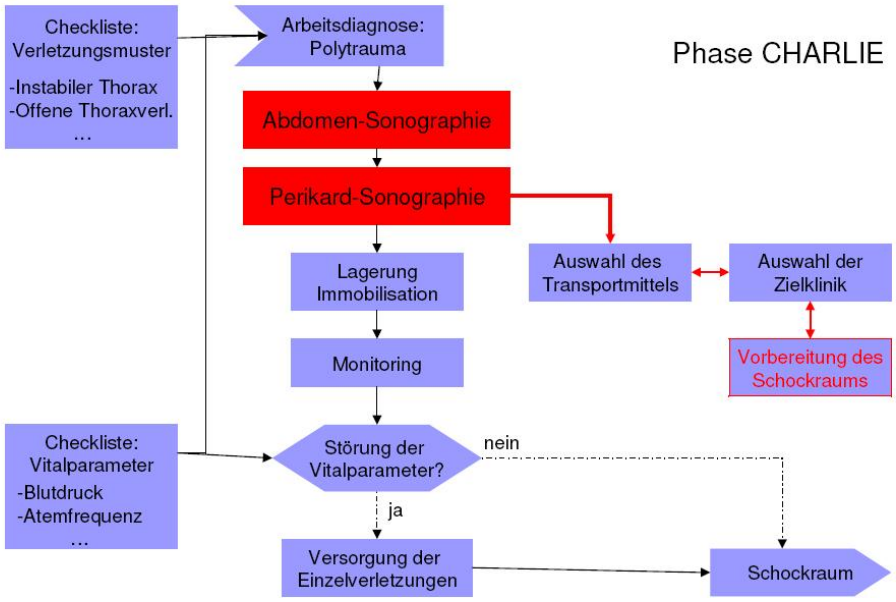


Abbildung 4.6: Erweitertes präklinisches Polytrauma Management – Integration der mobilen Sonographie

Der Notarzt vor Ort muss somit die Entscheidung treffen, welches Krankenhaus für die weitere Versorgung des Patienten geeignet ist. Nicht jedes Krankenhaus verfügt über die notwendige Ausstattung, um vital bedrohte polytraumatisierte Patienten zu behandeln. Welche Voraussetzungen ein Krankenhaus erfüllen muss, ist auch vielfach untersucht worden. Eine Übersicht gibt Prof. Dr. Haas in [Haas 1997], der auch die spezielle traumatologische Infrastruktur und Logistik bewertet. Daraus resultiert die direkte Beziehung zwischen der Verletzungsschwere des Patienten und die Wahl des Zielkrankenhauses.

In [Haas 1997] wird in Bezugnahme zu den bestehenden Algorithmen die These aufgestellt, dass der Transport eines polytraumatisierten Patienten auch über weite Entfernungen in Kauf genommen werden muss, da die Versorgung nur in einem geeigneten Krankenhaus vorgenommen werden kann. Auch die Verlegung in ein besser ausgerüstetes Krankenhaus nach Aufnahme in ein „normales“ Krankenhaus wird als mögliche Lösung vorgestellt. Greift man diese Probleme auf, so kann durch die bereits am Unfallort durchgeführte Sonographie ein entscheidender Vorteil bei der Patientenversorgung erzielt werden, da durch das frühzeitige Erkennen von freier abdominaler Flüssigkeit nicht nur direkt die geeignete Klinik angesteuert werden kann, sondern auch die nachträgliche Verlegung des Patienten mit all seinen Nachteilen entfällt.

Die gemachten Ausführungen finden sich im Algorithmus der Abbildung 4.6 als *Auswahl der Zielklinik* und im Vorgriff auf die Einlieferung in die Zielklinik als *Vorbereitung des*

Schockraums wieder. In diesem Zusammenhang findet auch die Auswahl des geeigneten Transportmittels statt. Hier ist zu prüfen, ob die Schwere des Notfalls ggf. den Einsatz des Rettungshubschraubers erfordert oder der „klassische“ Transport per Rettungswagen ausreichend erscheint.

In [Ziegenfuss 1998] wird die Dringlichkeit des Transports basierend auf dem Zustand des Patienten, eingehend erläutert. Entscheidend ist die Frage, ob ein *Abdominaltrauma* vorliegt, da starke Blutungen der inneren Organe eine der häufigsten Ursachen für einen hämorrhagischen Schock nach einem Polytrauma darstellen. Eine sichere Diagnose kann mittels Ultraschalluntersuchung erfolgen. Bei der Abdomen-Sonographie wird der Bauchraum *geschallt*, sodass die Leber, die Gallenblase, die Milz, die Nieren und die Lymphknoten dargestellt werden können. Zur schnellen Überprüfung auf freie Flüssigkeit kann der FAST-Algorithmus (vgl. 3.2) zum Einsatz kommen. Unabhängig ob die Untersuchung von geschultem Personal durchgeführt wird oder die Übertragung der Daten in Form einer Expertenkonsultation mit einer Spezialklinik erfolgt, kann, durch den im Algorithmus als *Abdomen-Sonographie* bezeichneten Schritt, eine Aussage über freie intraabdominale Flüssigkeit gemacht werden. Wird freie Flüssigkeit diagnostiziert, ist die Einweisung in eine besonders eingerichtete Klinik notwendig. Weiterhin kann im Vorfeld mittels Sonographie geprüft werden, ob ein Perikard-Erguss, also eine Flüssigkeitsansammlung im Herzbeutel, vorliegt. Der notwendige Schritt ist im Algorithmus als *Perikard-Sonographie* ausgezeichnet. Nach Feststellung kann ein geeignetes Krankenhaus ausgewählt und informiert werden. Durch die frühzeitige Information kann das Zielkrankenhaus die Vorbereitungen für die OP treffen.

Durch die hier eingeführten Erweiterungen kann die präklinische Sonographie in die Rettungskette integriert werden.

Teil II

Flexible und anpassbare Datenübertragung zur Realisierung von Telemedizin-Anwendungen

5 Einleitung

5.1 Einleitung und existierende Lösungsansätze

Bisher wurde das IN im Zusammenhang mit der sicheren Übertragung von Daten verwendet. Die zeitlich früheren Veröffentlichungen zu diesem speziellen Thema werden gesondert in Teil III dieser Arbeit unter dem Sicherheitsaspekt im Allgemeinen und für personenbezogene Medizindaten im Speziellen abgehandelt. Die Möglichkeit, über maschinelles Lernen auf die Datenübertragung in der Transportschicht Einfluss zu nehmen, kann aber auch in anderen Bereichen gewinnbringend eingesetzt werden.

Die besonderen Eigenschaften von SCTP (vgl. Abschnitt 6) machen dieses Transportprotokoll für Nicht-Standard-Anwendungen interessant. So wurde in [Ladha und Amer 2003b] ein FTP-Protokoll aufgesetzt, welches unter Verwendung der Multi-Streaming-Eigenschaft von SCTP (vgl. Abschnitt 6.5.1) einen performanten Datentransfer ermöglicht.

Ein weites Feld an Anwendungen eröffnet die *Multihoming-Eigenschaft* von SCTP. Multihoming wurde von Braden in [Braden 1989] eingeführt und folgendermaßen definiert:

Definition 5.1.1 (Multihoming) *Ein Kommunikationsendpunkt hat die Eigenschaft des* Multihoming[1]*, falls er über mehrere IP-Adressen angesprochen werden kann.*

Häufig erreicht man die Multihoming-Eigenschaft, indem einem Rechner physikalisch mehrere Netzkarten zugeordnet werden. Eine spezielle Form des Multihoming wird bei SCTP mit dem Ziel verwendet, Ausfallsicherheit und Zuverlässigkeit zu erreichen. Ausfallsicherheit wird erreicht, indem bei Ausfall eines Pfades automatisch auf einen alternativen Pfad umgeschaltet wird. Dadurch kann die Erreichbarkeit eines Kommunikationsendpunktes deutlich erhöht werden. Zusätzlich werden die sogenannten Sekundärpfade zur effizienten Übertragung von nicht rechtzeitig bestätigten Pakten verwendet.

Das Problem des Datentransfers über mehrere Kanäle ist mehrfach bereits Thema von Veröffentlichungen gewesen. Im Folgenden wird speziell auf die Verwendung von SCTP abgestellt. Grundsätzlich ist SCTP-Multihoming nicht für die parallele Übertragung von Daten ausgelegt. Hierbei treten unterschiedlichste Probleme auf, die u.a. in [Jungmaier und Rathgeb 2006] ausführlich behandelt werden, aber nicht abschließend gelöst werden konnten. Einen Sonderfall der parallelen Übertragung nimmt die sogenannte *konkurrierende Multipfad-Übertragung* ein. In [Iyengar et al. 2006] ist die Übertragung

[1] A host is said to be multihomed if it has multiple IP addresses.

über mehrere Kanäle folgendermaßen definiert:

Definition 5.1.2 (konkurrierende Multipfad-Übertragung (CMT)) *Unter* Concurrent Multipath Transfer (CMT)[2] *wird die Verteilung des Datentransfers auf mehrere Kanäle und Zieladressen bezeichnet, mit dem Ziel, einer Anwendung einen größeren Datendurchsatz anbieten zu können.*

In diesem Zusammenhang wurden in dem o.a. Artikel folgende Probleme herausgearbeitet und Lösungsvorschläge unterbreitet:

1. **Problem** Unnötige Fast-Retransmission[3]
2. **Problem** Verringerung des Berechnungszyklus für das Sendefenster (cwnd)[4]
3. **Problem** Anstieg des Traffics durch vermehrtes Senden von Bestätigungen (Acknowledgements)[5]

Ein bisher noch nicht angesprochenes Problem stellt die Pfadwahl dar. Geht man davon aus, dass nicht jeder Kanal dieselben physikalischen Eigenschaften besitzt bzw. durch unterschiedliche Auslastung auch unterschiedliche Leistungsmerkmale besitzt, so ist sicherlich die gleichmäßige Verteilung der Daten auf alle zur Verfügung stehenden Kanäle nicht sinnvoll. Allerdings stehen im Normalfall dem Netz keine bzw. nicht ausreichende Informationen über die aktuelle Netzsituation zur Verfügung, um hier nach einfachen Kriterien immer den besten Pfad zu wählen. SCTP stellt diese Informationen zwar bereit, sie werden derzeit aber nicht weiter verwendet, da grundsätzlich die Ausfallsicherheit im Vordergrund steht. Zudem ist derzeit kein allgemeingültiger Algorithmus bekannt, der in der Lage wäre, diese Entscheidung zu treffen.

Um den komplexen Zusammenhang der einzelnen Parameter auf einfache Merkmale, Regeln und Funktionen abzubilden, bieten sich Algorithmen, Verfahren und Modelle aus der statistischen Mustererkennung und des Data Mining an, und somit besteht die Möglichkeit, ein solches System in das IN einzubinden. Das IN ist somit in der Lage, für jeden zu übertragenden Chunk bzw. für jedes zu übertragende Datenpaket zu entscheiden, welcher Pfad die günstigsten Übertragungseigenschaften aufweist. Im weiteren Verlauf der vorliegenden Arbeit wird auf diesen Punkt fokussiert und nach einer allgemeingültigen Lösung des Pfadwahlproblems bei Parallelübertragung von Daten in einem Multihomed-SCTP-Netzwerk gesucht (vgl. auch die Problembeschreibung in Abschnitt 7.1).

[2] We propose using Concurrent Multipath Transfer (CMT) between multihomed source and destination hosts to increase an application's throughput.
[3] Preventing Unnecessary Fast Retransmissions
[4] Avoiding Reduction in cwnd Updates
[5] Curbing Increase in Ack Traffic

5.2 Die Testumgebung

Für die programmtechnische Umsetzung wurde die SCTPLib-Implementierung [Tüxen] verwendet. Diese im universitären Umfeld beliebte Implementierung bietet für die Verifikation von Netzapplikationen eine Vielzahl von Möglichkeiten. So können einfach Erweiterungen vorgesehen und eingebunden werden, die dann durch die Verwendung von speziell angepassten Testtools hinsichtlich ihrer Leistungsfähigkeit überprüft werden können.

Weiterhin stand eine prototypische Implementierung von Secure-SCTP [Unurkhaan et al. 2004] auf Basis der SCTPLibs zur Verfügung. Die notwendigen kryptographischen Verfahren werden von der Cryptolib des OpenSSL-Projekts bereitgestellt. Demnach liegen die verwendeten Algorithmen bereits optimiert vor.

Um unterschiedliche Pfade bereitstellen zu können, wurde für die Testumgebung zusätzlich ein Netzwerkemulator in das Netzwerk integriert. Aufgrund der vielfältigen Variationsmöglichkeiten kommt NISTNET [NIST 2003] als Emulator zum Einsatz. Neben Verzögerungen und angepassten Bandbreiten kann ein beliebiger Paketverlust emuliert werden. Mit Hilfe von NistNet können somit beliebige reale Szenarien und Pfadeigenschaften abgebildet werden.

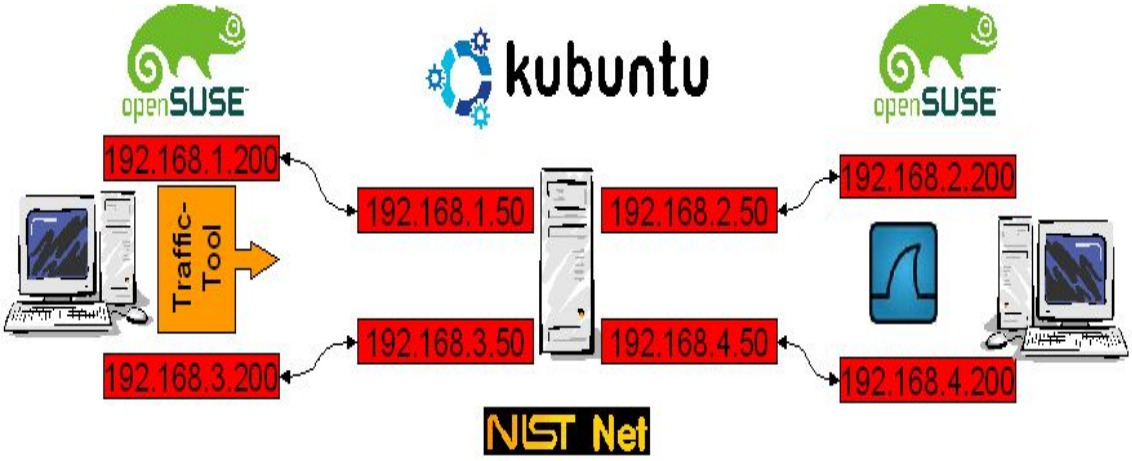


Abbildung 5.1: Testumgebung

Die vollständige Testumgebung ist in Abbildung 5.1 dargestellt. Neben NistNet für die Emulation der Netzwerkcharakteristik werden noch ein Trafficgenerator und ein Netzwerküberwachungstool benötigt. Als Traffic-Generator kommt ein kleines Test-Tool auf Basis der SCTPLib, welches ähnlich wie NetPerf (s.a. [Sloan 2001]) funktioniert, zum Einsatz, das nicht nur Daten in unterschiedlicher Größe generieren kann, sondern auch die Ergebnisse protokolliert. Die genaue Funktionsweise wird im Abschnitt 8.3 genauer

anhand von Beispielen erläutert, da hierfür spezielle Grundlagen von SCTP bekannt sein müssen. Zusätzlich wird der Netzwerksniffer Wireshark ([Orebaugh, A. and Syngress Autorenteam 2004]) eingesetzt, um auf der Empfängerseite den realen Datentransfer zu analysieren. Die dort gemessenen Werte können in beliebige Datenformate exportiert werden, sodass die Möglichkeit besteht, diese mit einem Statistikprogramm wie beispielsweise *Gnu R*, welches für sämtliche Berechnungen der vorliegenden Arbeit zum Einsatz kam, mathematisch zu beurteilen.

5.3 Gliederung

Nachdem in der Einleitung (Abschnitt 5) die existierenden Lösungsansätze sowie die verwendete Teststellung beschrieben wurden, folgt ein kurzer Abriss über die Funktionalität von SCTP (Abschnitt 6).

In Abschnitt 7.1 wird das zu lösende Problem sprachlich abgefasst und die notwendigen Definitionen bereitgestellt.

Der Abschnitt 7 legt die theoretischen Grundlagen auf dem Gebiet des maschinellen Lernens und der Statistik dar, wobei speziell auf die Problematik der Pfadwahl und der Anwendung als IN im SCTP-Kern eingegangen wird. Neben einer allgemeinen Beschreibung der Begriffe Data Mining, maschinelles Lernen und Wissensentdeckung (Abschnit 7.2) wird konkret die Lineare Diskriminanzanalyse (Abscchnitt 7.3) als zentrales Verfahren besprochen. Darüber hinaus wird auf die Spezialisierung als Regressionsbäume eingegangen (Abschnitt 7.5). Dies rundet den theoretischen Teil ab.

Im praktischen Teil folgt die Datenanalyse (Abschnitt 8) und Datenauswertung (Abschnitt 9).

Zuerst wird geprüft, welche Parameter von SCTP bereitgestellt werden (Abschnitt 8.1) bzw. für die Anwendung in Frage kommen (Abschnitt 8.2).

Danach wird erläutert, wie die Test- und Trainingsdaten u.V. der SCTPlib generiert wurden (Abschnitt 8.3), gefolgt von einem kurzen Abriss über den Prozess der Wissensentdeckung (Abschnitt 8.4). Für die Anwendung von Lernverfahren ist die Auswahl von geeigneten Attributen und aussagekräftigen Trainingsdatensätzen unabdingbar. Die Analyse der Testdaten sowie die folgende Selektion, Vorverarbeitung und Transformation werden in den restlichen Teilabschnitten (Abschnitte 8.5 und 8.6 sowie 8.7) erläutert.

Es folgt der entscheidende Abschnitt, der sich mit der Auswertung der Daten bzw. den konkreten Ergebnissen der durchgeführten Testläufe beschäftigt (Abschnitt 9). Als Erstes werden grundsätzliche Probleme des konkurrierenden Datentransfers beleuchtet (Abschnitt 9.1). Die Testläufe wurden in drei Grundszenarien zerlegt. In die Betrach-

tung der MP-SCTP-Übertragung auf Pfaden gleicher Kapazität (Abschnitt 9.3) sowie die Übertragung auf Pfaden unterschiedlicher Dienstgüte (Abschnitt 9.2), die in die Unterabschnitte „Betrachtung bei geringem Abstand“ (Abschnitt 9.2.1) und „Betrachtung bei größerem Abstand“ (Abschnitt 9.2.4) zerfällt.

Jedes Szenario betrachtet die Übertragung mit Pfadwahl durch das IN im Vergleich zu den Referenzübertragungen mittels Standard-SCTP und Loadsharing (Abschnitte 9.2.1, 9.2.4 und 9.3.1). Zusätzlich wurde für jedes Szenario die Übertragung unter Verwendung von a-priori Informationen in die Betrachtung mit aufgenommen. Die Analyse wurde anhand von charakteristischen Verläufen durchgeführt, die durch die Anwendung von Versuchsreihen statistisch verallgemeinert werden (Abschnitt 9.2.2 und 9.2.5 sowie 9.3.2).

Jede Analyse wird mit einem Kurz-Resümee abgeschlossen (Abschnitt 9.2.3 und 9.2.6). In einer abschließenden Betrachtung werden die Ergebnisse (Abschnitt 9.3.2) zusammengeführt.

6 SCTP - Überblick

Ursprünglich war SCTP gar nicht zur Übertragung von Nutzlast und somit als Ersatz für TCP bzw. UDP vorgesehen. Ziel der ersten Definition von SCTP war die Übertragung von Signalisierungsinformationen über IP-Netze. Bei der Realisierung wurde festgestellt, dass die umgesetzten Eigenschaften und „Neuerungen“ im Vergleich zu bestehenden Transportprotokollen den Einsatz als vollwertiges Transportprotokoll rechtfertigen. Einige fehlende Features und Einschränkungen lassen sich mit der Herkunft und dem ursprünglichen Zweck von SCTP erklären. Da sich die Spezifikation von SCTP noch im Fluss befindet, sind im Laufe der Zeit immer wieder Ergänzungen und Erweiterungen vorgestellt worden, die es ermöglichen, SCTP sehr weitreichend einzusetzen.

Einige Erweiterungen werden in dieser Arbeit explizit angesprochen, wobei die meisten aufgrund der besonderen Problematik bei der kryptographischen Absicherung mit herkömmlichen Standardverfahren, in Teil III dieser Arbeit, ausführlich beleuchtet werden. So wird die PR-SCTP-Erweiterung (vgl. Abschnitt 10.2) zur Übertragung von Multimedia-Daten herangezogen. Der Bereich Mobilkommunikation stellt neue und interessante Anforderungen an ein Transferprotokoll, die sich mit den herkömmlichen Methoden aus dem stationären Bereich nicht umsetzen lassen. Ein kritischer Punkt ist beispielsweise die Mobilität selbst. Wechselt ein Endgerät seinen Standort, so kann die ihm zugeordnete IP-Adresse ggf. nicht mehr erreicht werden. Für diese und ähnlichen Szenarien wurden verschiedene Lösungsansätze publiziert. Für SCTP gibt es beispielsweise die Erweiterungen *mobile SCTP (mSCTP)* und *cellular SCTP*(cSCTP), die auf Basis der dynamischen Rekonfiguration von IP-Adressen (ADDIP) den Übergang in einen anderen IP-Bereich ermöglichen. Zudem gibt es Erweiterungen, die mehr Flexibilität bei der Verwendung der zur Verfügung stehenden Pfaden ermöglicht. Eine Variante ist das LS-SCTP, welches ein rudimentäres Loadsharing über SCTP ermöglicht.

Es existieren auch Erweiterungen, die dem höheren Sicherheitsanspruch moderner Netzanwendungen Rechnung tragen. So wird beispielsweise mittels *Secure-SCTP*, welches in Abschnitt 11.1 betrachtet wird, eine flexible, schnelle und insbesondere individuell konfigurierbare Sicherheitsarchitektur bereitgestellt. Im Rahmen von Teil III dieser Arbeit wird die Implementierung von Secure-SCTP auf Basis der SCTPLib verwendet. Insbesondere wird der zusätzliche Aufwand durch die kryptografische Behandlung von Daten-Chunks im Mittelpunkt stehen. Aufbauend auf diesen Erkenntnissen wird ein flexibles Sichherheitskonzept für SCTP, unter besonderer Berücksichtigung der in Teil I eingeführten Medizinszenarien, erarbeitet.

Die bei der Erstellung dieser Arbeit letzte aktuelle Version der SCTP-Spezifikation

stammt aus dem Jahr 2007 und wurde als RFC 4960 in [Stewart 2007] veröffentlicht. Neben den RFCs gibt es mittlerweile Zusatzliteratur, die nicht nur auf die reine technische Umsetzung, sondern auch auf die nutzbringende Verwendung der neuen Konzepte fokussiert. So haben die „Erfinder“ von SCTP ein umfassendes Handbuch (vgl. [Stewart and Xie 2001]) vorgelegt, das auch auf Hintergründe und Erweiterungen von SCTP eingeht. Will man in den Code eingreifen und selber Anwendungen auf Basis von SCTP schreiben, so empfiehlt sich [Stevens 1998], welches in der aktuellen Auflage einen eigenen Abschnitt über SCTP aufgenommen hat. Hier wird SCTP aus Sicht der zugrunde liegenden C-Programmierung beleuchtet.

Für das Verständnis der Systematik des IN und die Einbindung in das SCTP-Protokoll werden in diesem Abschnitt die Eigenschaften und Besonderheiten von SCTP zusammengestellt. Da SCTP in direkter Konkurrenz zu TCP und UDP steht, werden diese althergebrachten Protokolle des Häufigeren mit SCTP verglichen und bewertet.

6.1 Grundlegende Konzepte von SCTP

Bevor ausführlich auf die Konzepte von SCTP eingegangen wird, wird mit wenigen Worten versucht, das Wesen des SCTP-Protokolls zu beschreiben.

Definition 6.1.1 (SCTP) SCTP – *Stream Control Transmission Protocoll – ist ein verbindungsorientiertes Transportprotokoll, das den zuverlässigen Transport von Nachrichten zwischen IP-basierten Endpunkten ermöglicht.*

Anzumerken ist, dass SCTP anders als beispielsweise TCP nachrichtenorientiert arbeitet. TCP minimiert hingegen den Protokolloverhead, indem Nachrichten zu einem Bytestrom zusammengefasst werden. Es werden somit keine Nachrichten, sonden ein Strom aus einzelnen Bytes transportiert (vgl. [Jungmaier 2005]).

SCTP wird zur Übertragung von Daten über IP-Netze eingesetzt. In IP-Netzen kann ein verbindungsloser Datentransfer, wie er beispielsweise von UDP praktiziert wird, durchgeführt werden. Dies führt nicht nur zu Problemen bei der Stau- und Flusskontrolle, sondern auch bei der Verschlüsselung und Authentifizierung der Daten. Daher wird auf die verbindungsorientierte Kommunikation fokussiert, d.h. alle Teilnehmer an der Kommunikation müssen vor der eigentlichen Übertragung eine Verbindung als gemeinsame Basis aushandeln. Um eine Abgrenzung zum verbindungsorientierten TCP zu erreichen, verwendet SCTP eine eigene Begrifflichkeit.

Definition 6.1.2 (Assoziation) *Die an einer Verbindung teilnehmenden Parteien werden als* Endpunkte (engl.: endpoints) *bezeichnet. Eine Beziehung zwischen den Endpunkten, sprich: die Verbindung heißt* Assoziation (eng. association).

Um eine eindeutige Zuordnung eines Netzwerk-Interfaces im Netz zu ermöglichen, werden IP-Adressen verwendet. Da SCTP auf einem anderen Port als TCP bzw. UDP läuft, ist neben der IP-Adresse die Portnummer zur Identifikation notwendig.

Definition 6.1.3 (SCTP-Transport-Adresse) *Die eindeutige Kombination aus IP-Adresse und einem SCTP-Port wird als* SCTP-Transport-Adresse (engl. SCTP transport address) *bezeichnet. Alle SCTP-Transport-Adressen eines Endpunktes weisen identische Portnummern auf.*

Bei der zu verwendenden IP-Adresse ist eine Einschränkung zu beachten, da lediglich Unicast-Adressen verwendet werden dürfen. IP-Adressen für Gruppenkommunikation, wie IP-Multicast und IP-Broadcast-Adressen, können demnach mit SCTP nicht eingesetzt werden.

Auch für die ausführenden Anwendungen hat SCTP einen eigenen Sprachgebrauch.

Definition 6.1.4 (ULP) *Eine Anwendung / Applikation, die von einer höheren Schicht (engl. Layer) aus die Dienste (engl. Services) von SCTP verwendet, wird auch als* Upperlayer-Protokoll (ULP) *bezeichnet.*

In der vorliegenden Arbeit wird meistens der aussagekräftigere Begriff der Anwendung bevorzugt verwendet.

Streams und Chunks

Wie bereits erwähnt, werden die Transportadressen, die im weiteren Verlauf der Übertragung genutzt werden sollen, beim Verbindungsaufbau ausgetauscht. Zusätzlich erfolgt die Festlegung der sogenannten *Streams*, die u.a. zur feineren Granulierung des Transports verwendet werden können. In Abschnitt 3.3 wurden Streams im erweiterten Notfallszenario bereits aus fachlicher Sicht eingeführt, an dieser Stelle wird jetzt die formale Definition des Begriff nachgeholt.

Definition 6.1.5 (SCTP-Stream) *Ein unidirektionaler, logischer Übertragungskanal für Nachrichtenströme wird als* SCTP-Stream *oder kurz* Stream *bezeichnet. Es können bis zu* 65.536 *Streams beim Verbindungsaufbau festgelegt werden, die den simultanen Datentransfer der Endpunkte ermöglichen.*

Die übergeordnete Anwendung legt fest, welche Daten auf welchen Stream übertragen werden sollen. Dies ist sinnvoll, da man so Daten mit unterschiedlichen Anforderungen an die Übertragung – beispielsweise Textdaten und Multimediadaten – auf getrennten Streams übertragen kann. Dateneinheiten, die von einer übergeordneten Anwendung an das SCTP-Protokoll herangetragen werden, heißen *Nachrichten*, da SCTP informationsorientiert und nicht auf Basis von Bytestreams agiert. Zur eindeutigen Festlegung des Sendestreams wird der *Stream-Identifier* verwendet.

Definition 6.1.6 (Stream-Identifier) *Der* Stream-Identifier*(SI) oder in der deutschsprachigen Literatur als* Stream-Identifikation *bezeichnet, legt fest, auf welchen Stream die Nutzdaten übertragen werden.*

Jeder Stream ermöglicht einen geordneten Versand, d.h. die Reihenfolge der Daten bleibt erhalten. Um dies zu gewährleisten, muss der Empfänger-Stream wissen, in welcher Reihenfolge die Daten vom Sender in das Netz eingespeist wurden. Deshalb wird den Daten zusätzlich eine aufsteigend sortierte Nummerierung mitgegeben, die sogenannte *Stream-Sequence-Nummer*.

Definition 6.1.7 (Stream-Sequence-Nummer) *Den Daten, die auf einen bestimmten Stream übertragen werden sollen, wird eine eindeutige* Stream-Sequence-Nummer (SSN) *zugeordnet, aus der die Reihenfolge der Einspeisung in den Stream hervorgeht.*

Alle Informationen, seien es Kontrollinformationen oder auch Nutzdaten, werden in SCTP-Pakete verpackt, die zwischen Sender und Empfänger ausgetauscht werden. Die nächst kleinere Einheit ist der sogenannte *Chunk*, wobei ein SCTP-Paket mehrere Chunks aufnehmen kann.

Definition 6.1.8 (Chunk) *Eine Protokolleinheit (eng. Building Block), die zur Übertragung von Informationen in ein SCTP-Paket eingebunden wird, heißt* Chunk. *Ein Chunk ist die Informationen tragende Basisstruktur von SCTP. Jeder Chunk wird durch einen eindeutigen ein Byte großen Wert identifiziert. Somit können maximal 255 verschiedene Chunk-Typen definiert werden.*

Chunks sind die Grundlage für den gesamten Datentransfer, unabhängig davon, ob es gilt, Nutzdaten oder Verwaltungsinformationen zu übertragen.

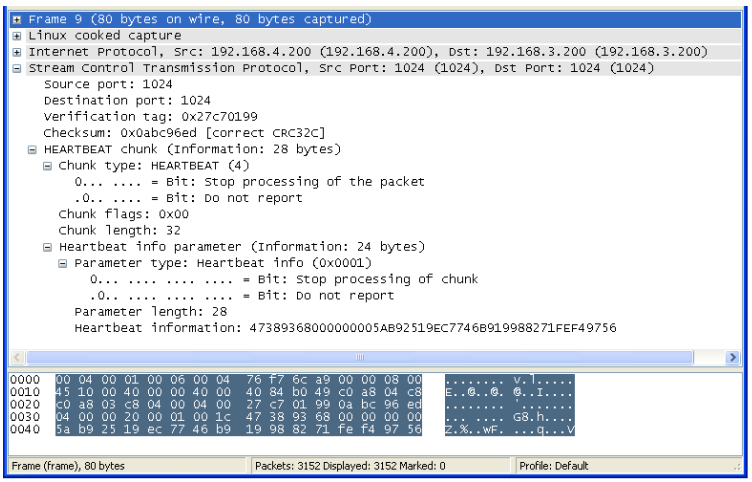


Abbildung 6.1: Der Heartbeat-Chunk in Wireshark

Definition 6.1.9 (Chunkklassen) *Es wird zwischen* Daten-Chunks *und den* Kontroll-Chunks *unterschieden. Mit Daten-Chunks werden Nachrichten, sprich Nutzdaten über die Assoziation übertragen, während Kontroll-Chunks Informationen zur Steuerung der SCTP-Assoziation enthalten.*

Vom Standard ([Stewart 2007]) wird ein Daten-Chunk *DATA* (0x00)[1] bereitgestellt, die Definition von weiteren Varianten ist denkbar und sinnvoll. So wird beispielsweise im *SecureSCTP* (vgl. Abschnitt 11.1) ein verschlüsselter Daten-Chunk *Encrypted Data EncData (0x10)* verwendet. Die konkreten Elemente eines Daten-Chunks werden im Abschnitt über den zuverlässigen Datenversand thematisiert, daher ist die grafische Darstellung in Abbildung 6.10a zu finden.

In Abbildung 6.1 ist die Sichtweise von Wireshark auf einen Kontroll-Chunk, in diesem Fall ein Heartbeat-Chunk, dargestellt. Anhand dieser Darstellung der Capture-Datei lassen sich die wesentlichen Datenfelder eines Chunks ablesen. Man erkennt den *Chunk-Typ*, in diesem Fall der Typ 4 – Heartbeat – und die Chunk-Länge von 28 Byte sowie Chunk-Flags in der Größenordnung 1 Byte. Der Chunk-Typ wird auch in einem 1 Byte großen Feld hinterlegt, während das Längenfeld zwei Byte groß ist. Es werden somit 4 Byte an Kontrollinformationen verwendet. Ein Chunk, der keine Daten enthält, hat damit eine Größe von 4 Byte, da die Kontrollinformationen mit in die Längenberechnung eingehen. Per Definition ist die Länge eines Chunks immer ein Vielfaches von 32 Bit, sodass die Nutzdaten ggf. mit Fülldaten (eng. padding) aufgefüllt werden müssen.

Ein Daten-Chunk enthält die zu übertragenden Daten als Nutzdaten, während ein Kontrollchunk zusätzliche Informationen – sogenannte Parameter – enthalten kann, die im Nutzdaten-Feld transportiert werden. Der Parameterbereich kann seinerseits wieder unterteilt werden. Jeder Parameterbereich beginnt mit einem zwei Byte großen Feld, das den Parametertyp spezifiziert und einem ebenfalls zwei Byte großen Feld, das die Parameter-Länge angibt, wobei hier die Kontrollinformationen des Parameterbereichs äquivalent zur Chunk-Länge mit einbezogen werden.

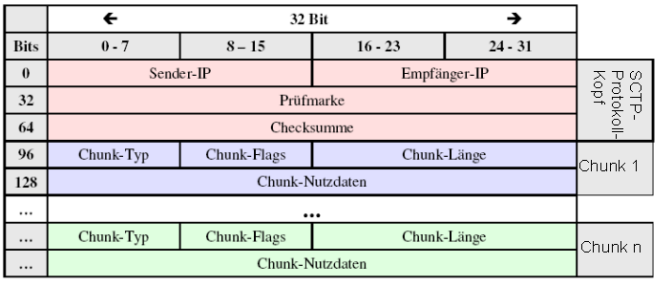


Abbildung 6.2: Das SCTP-Paket – grafische Darstellung

[1] In Klammern ist die Kodierung des Chunk-Typs angegeben.

Die Kontroll-Chunks können anhand ihrer Funktion klassifiziert werden. Derzeit sind zwölf Chunks in sieben funktionalen Klassen in [Stewart 2007] klassifiziert, für Erweiterungen können bis zu 243 weitere Chunk-Typen hinzukommen, die je nach Bedarf in das bestehende System eingefügt werden. Erweiterungen benötigen im Normallfall zusätzliche Kommunikation zwischen den Endpunkten, dies ist nur durch neue Chunk-Typen und die entsprechende Verarbeitung beim Sender und Empfänger möglich. Das PR-SCTP definiert beispielsweise einen neuen Chunk-Typ *forward-tsn*, mit dessen Hilfe ein „kontrollierter Verlust“ von Datenpaketen realisiert wird.

Folgende Chunks stehen aufgrund der Basisdefinition zur Verfügung:

Chunks für den Verbindunsaufbau

Zum Verbindungsaufbau werden die Chunks *INIT*(0x01), *COOKIE-ECHO*(0x01) und die zugehörigen Bestätigungschunks *COOKIE-ACK*(0x0b) und *INIT-ACK*(0x02) verwendet. Wie mit Hilfe dieser Chunks eine Verbindung aufgebaut wird, wird in Abschnitt 6.2.1 beschrieben.

Chunks für den Verbindungsabbau

Zum Verbindungsabbau werden die Chunks *ABORT*(0x06), *SHUTDOWN*(0x07) und *SHUTDOWN-COMPLETE*(0x0e) und *SHUTDOWN-ACK*(0x08) verwendet. Wie mit Hilfe dieser Chunks eine bestehende Verbindung wieder abgebaut wird, wird in Abschnitt 6.2.3 beschrieben.

Empfangsbestätigung

Um eine zuverlässige Übertragung der Daten zu erreichen, muss der Sender darüber informiert werden, welche Chunks beim Empfänger bereits eingegangen sind. Dies übernimmt der *SELECTIVE DATA ACKNOWLEDGEMENT – SACK*(0x03). Um den Empfang beim Sender zu quittieren, muss der Chunk eindeutig identifizierbar sein bzw. muss die Sendereihenfolge eindeutig zugeordnet werden können.

Definition 6.1.10 (TSN) *Um die Sendereihenfolge eines Chunks zu verfolgen, wird jedem Chunk eine eindeutige* 32*-Bit große* Übertragungs-Sequenz-Nummer *(engl. Transmission Sequence Numbers TSN) zugeordnet. Wird eine Nachricht in mehrere Daten-Chunks zerlegt (fragmentiert), so erhält jeder Chunk eine eindeutige TSN.*

Der SACK-Chunk ist in Abbildung 6.10b im Abschnitt über den zuverlässigen Datentransfer grafisch dargestellt, auf die einzelnen Parameter soll in dieser Einführung aber bereits eingegangen werden, da die einzelnen Begrifflichkeiten zu den Kernmechanismen von SCTP zählen. Als Parameter enthält ein SACK-Chunk die letzte in fortlaufender Reihenfolge empfangene Chunk-TSN, auch als *Cumulative TSN acknowledgment* bezeichnet. Bis zu dieser TSN sind alle gesendeten Chunks ordnungsgemäß beim Empfänger eingegangen.

Dies ist aber nicht die einzige Aufgabe des SACK-Chunks, sondern er steuert auch Parameter und Informationen für die Flusssteuerung bei, die in Abschnitt 6.4 behandelt werden. Zusätzlich werden indirekt Lücken, sogenannte Gaps, von fehlenden TSNs berichtet. Der SACK-Chunk erfüllt damit eine wichtige Aufgabe, da die wesentlichen Informationen über die aktuelle Netzsituation aus den eingegangenen SACK-Chunks abgelesen werden können. Der SACK-Chunk mit seiner Vielzahl an Parametern wird in Abschnitt 6.3.4 ausführlich besprochen.

Wird die Multi-Homing-Fähigkeit von SCTP verwendet, die in Abschnitt 6.5.2 formal eingeführt wird und bereits in Abschnitt 3.3 im Zusammenhang mit der Realisierung von Telemedizin informal eingeführt wurde, besteht das Problem der Pfadwahl für die Quittierung. Grundsätzlich gilt die Regel, dass ein SACK-Bestätigungs-Chunk an die Adresse gesendet wird, von der auch der zu bestätigende Daten-Chunk versendet wurde. Wann und wofür die einzelnen Pfade verwendet werden, wird in Abschnitt 6.5.2 ausführlich besprochen, an dieser Stelle sei angemerkt, dass im Multi-Homing-Szenario mehrere Pfade zur Verfügung stehen und es somit vorkommen kann, dass mit einem einzigen SACK-Chunk Daten-Chunks von verschiedenen Quellen quittiert werden müssen. Für dieses Problem sieht die SCTP-Spezifikation eine pragmatische Lösung vor, und zwar wird der SACK-Chunk an die Quelladresse des letzten zu quittierenden Daten-Chunks gesendet.

Erreichbarkeitsprüfung und Pfadüberwachung

Wenn über einen Pfad aktuell Daten gesendet werden, bezeichnet man ihn als *aktiven Pfad*. Bei einem aktiven Pfad kann man anhand der eingehenden SACK-Chunks einfach die aktuelle Übertragunsleistung des Pfads ermitteln. Wenn in einem gegebenen Zeitraum keine Rückmeldungen auf bereits gesendete Daten eingehen, kann man weiterhin schließen, dass der Pfad nicht mehr erreichbar ist.

Anders sieht es bei sogenannten *inaktiven* Pfaden aus, da hier keine Daten gesendet werden und daher auch keine SACK-Quittungen eingehen. Bei inaktiven Pfaden kann es sich neben einem ungenutzten Primärpfad auch um die Sekundärpfade in einem Multihoming-Szenario handeln, da diese nur als Ausfalllösung gedacht sind und bei aktivem Primärpfad nicht zum Datentransfer herangezogen werden. Um Aussagen über die Erreichbarkeit eines Endpunktes machen zu können, wird demnach ein zusätzlicher Mechanismus benötigt.

Über den *Heartbeat-Chunk*(0x04) und den *Heartbeat-ACK-Chunk*(0x05) stellt SCTP diesen Mechanismus zur Verfügung. Auf allen inaktiven Leitungen werden periodisch Heartbeat-Chunks gesendet und entsprechend quittiert, sodass hieraus die aktuelle Übertragungsrate des Pfades ermittelt werden kann. Treffen in einem bestimmten Abstand – hier greift der Parameter Path-Max-Retrans – keine Quittungen ein, wird davon ausgegangen, dass der Endpunkt nicht mehr erreichbar ist, und der Pfadzustand wird auf „unerreichbar“ gesetzt.

Die Informationen, die aus dem Heartbeat-Mechanismus gewonnen werden, können direkt zur Bestimmung der aktuellen Netzsituation verwendet werden, indem sie in die Berechnung der Rundenlaufzeit *RTT* gem. Definition 6.3.6 des betroffenen Pfades einfließen.

Fehlermeldungen

Mit dem *Operation Error-Chunk kurz ERROR*(0x9) kann ein Endpunkt Fehlermeldungen an seinen Verbindungspartner absetzen. Der Error-Chunk wird nur verwendet, wenn es sich nicht um einen schwerwiegenden Fehler handelt. Schwerwiegende Fehler sind immer mit dem Abbruch der Verbindung verbunden und werden über den ABORT-Chunk kommuniziert. Da die Fehler in Form eines Parameterbereichs – Typ, Länge, Wert – definiert werden, kann jeder Fehler über den Error-Chunk wie auch den Abort-Chunk transportiert werden.

Da Fehler im gesamten Lebenszyklus einer Assoziation auftreten können, folgt die detaillierte Beschreibung der einzelnen Fehler an den Stellen, an denen sie auch im Ablauf vorkommen.

Das SCTP-Paket

Chunk sind eingebettet in ein SCTP-Paket, dessen Aufbau in Abbildung 6.2 skizziert ist. Die entsprechenden Felder finden sich auch in der Wireshark-Capture-Ansicht in Abbildung 6.1 wieder.

So sind im *Protokollkopf* – im Englischen als *Common Header* bezeichnet – die SCTP-Transport-Adressen vom Sender und Empfänger hinterlegt. Im konkreten Beispiel sind dies die IP-Adressen 192.168.4.200 mit Port 1024 für den Sender und 192.168.3.200 mit demselben Port für den Empfänger. Ein SCTP-Paket wird eindeutig einem Pfad zugeordnet. Auch im später behandelten Multihoming-Szenario, bei dem einem Endpunkt mehrere Adressen zugewiesen sein können, ist die Ziel- und Quelladresse eindeutig. Wird ein Paket neu übertragen, so muss dafür bei Verwendung von Multihoming nicht zwingend der gleiche Pfad gewählt werden. Jede STCP-Transport-Adresse ist genau 16 Bit groß. Zu dem 96 Bit großen Protokollkopf gehören weiterhin eine 32-Bit große Prüfmarke – im englischen Sprachraum als *Verification-Tag* bezeichnet – und eine 32-Bit große Checksumme. Beide Informationen dienen der Sicherheit, um gefälschte bzw. veränderte Pakete zu erkennen.

Definition 6.1.11 (Verification-Tag) *Unter dem* Verification-Tag *versteht SCTP eine 32-Bit große Prüfmarke, die zufällig gewählt wird und es den Endpunkten ermöglicht, zu überprüfen, ob es sich bei einem SCTP-Paket um ein aktuelles Paket handelt und nicht beispielsweise von einer früheren Assoziation stammt.*

Der Verification-Tag wird bei der Initialisierung der Assoziation für die „Lebenszeit“ der Assoziation definiert, sodass die Beschreibung und Verwendung in Abschnitt 6.2.1 beim

Verbindungsaufbau behandelt wird.

6.2 Verbindungsaufbau und Verbindungsabbau

Nachdem die grundsätzlichen Begriffe von SCTP eingeführt sind, kann gezeigt werden, wie SCTP eine Verbindung, sprich: Assoziation aufbaut bzw. abbaut.

Der Verbindungsaufbau erfolgt mit einem sogenannten *4-Wege-Handshake*, d.h. es werden vier Pakete respektive Chunks für die Initialisierung übertragen und ausgewertet. Mit dem dritten bzw. dem vierten SCTP-Paket können allerdings bereits zusätzlich zu den für die Initialisierung benötigten Kontroll-Chunks die ersten Daten-Chunks übertragen werden.

Jeder Endpunkt befindet sich immer in einem wohldefinierten Zustand. Wenn noch keine Verbindung aufgebaut und kein Paket ausgetauscht wurde, befindet sich ein Endpunkt im Status *CLOSED*. Wenn die Assoziation etabliert wurde, d.h. die Initialisierung erfolgreich abgeschlossen werden konnte, befinden sich die Endpunkte im Status *ESTABLISHED*. Während des Verbindungsaufbaus können sich die Endpunkte in Zwischenzuständen, die als COOKIE-SENT und *COOKIE-ECHOED* bezeichnet werden, befinden. Bei der folgenden Beschreibung des Verbindungsaufbaus wird nach jedem „Handshake“ auf die Zustände und ihre Bedeutung für die Assoziation eingegangen.

Für die folgende Beschreibung des Verbindungsaufbaus werden die beteiligten Endknoten eindeutig benannt. Der Endknoten, der den Verbindungsaufbau initiiert, wird als *Sender* und der reagierende Endpunkt als *Empfänger* bezeichnet. Im ersten Schritt werden die ersten zwei „Handshakes“ betrachtet.

6.2.1 Die ersten zwei Pakete einer Assoziation

Der Verbindungsaufbau wird gestartet, indem der Sender einen Init-Chunk als erstes Paket der zu etablierenden Assoziation sendet, der von dem Empfänger mit einem Init-Ack-Chunk quittiert wird. Für die Initialisierung gibt es neben den vorgeschriebenen Parametern auch eine Reihe von Parametern, die optional verwendet werden können.

Pflichtparameter des INIT-Chunks

In Abbildung 6.3 ist der Aufbau eines minimalen Init-Chunks dargestellt, der sich lediglich aus den Pflichtparametern zusammensetzt.

Der Init-Chunk enthält grundsätzliche Parameter, die für den gesamten Lebenszyklus der Assoziation Bestand haben. Beginnen wir von unten, so wird die initiale TSN vorgeschrieben.

	← 32 Bit →			
Bits	**0 - 7**	**8 – 15**	**16 - 23**	**24 - 31**
0	Chunk-Typ 0x01	Flags = 0	Chunk-Länge 0x14	
32	Initiation-Tag (Initialisierung der Prüfmarke)			
64	a_rwnd (Advertised rwnd) (Initialisierung)			
96	Ausgehende Streams		Maximale Anzahl eingehender Streams	
128	Initialisierungs- TSN			

Abbildung 6.3: Die minimale Version des INIT-Chunks

Definition 6.2.1 (Initial-TSN) *Über das Feld* Initial-TSN *legt der Sender einen Anfangswert für die TSN fest, der als Basis für den Transfer verwendet wird. Normalerweise wird ein Zufallswert zwischen* $0x0$ *und* $0xFFFFFFFF$ *verwendet.*

Der erste Daten-Chunk wird demnach mit der Initial-TSN als TSN ausgeliefert.

Die Anzahl der für das Multi-Streaming (vgl. Abschnitt 6.5.1) benötigten Streams werden über die Felder *Ausgehende Streams* (OS) und *Maximal eingehende Streams* (MIS) ausgehandelt. Die hier vorgegebenen Werte sind als Verhandlungsgrundlage zu verstehen, die der Empfänger, falls er die hier vorgegebene Anzahl von Streams nicht unterstützt, korrigieren kann. Bei der Quittierung mittels Init-Ack-Chunk werden diese Werte entweder bestätigt oder korrigiert. Der Wert *Maximal eingehende Streams* enthält dabei die maximale Anzahl der Streams, die der Sender bereitstellen kann. Im Falle, dass der Empfänger eine größere Anzahl an Streams benötigt als der Sender verarbeiten kann, leitet der Empfänger den Abbruch der Verbindung ein. Im Falle, dass der Sender seinerseits nicht die Anzahl der vom Sender gewünschten ausgehenden Streams verarbeiten kann, teilt er dies dem Sender mit, indem er im Init-Ack-Chunk den Wert der aus seiner Sicht maximal möglichen eingehenden Streams im Feld MIS vermerkt.

Ein wichtiger Parameter für die Staukontrolle (vgl. Abschnitt 6.4) ist die aktuelle Größe (a_rwnd) des Empfangsfensters (rwnd), welches festlegt, wie viele Daten der Empfänger noch aufnehmen kann. Der Empfänger legt hierfür einen Bufferbereich an, der einen vorgegebenen Maximalwert nicht überschreiten kann. Die formale Definition 6.3.1 wird in Abschnitt 6.3.3 im Rahmen der konkreten Beschreibung der Empfängersicht auf die Datenübertragung nachgeholt.

Definition 6.2.2 (a_rwnd) *Mit dem Parameter* a_rwnd[2] *teilt der Sender des Chunks dem Empfänger mit, welche Buffergröße in Byte er für eingehende Daten reserviert hat.*

Die Abkürzung *a_rwnd* wird auch für den Parameter selber verwendet, was eine Verwechslungsgefahr in sich birgt, der ständig neu berechnet wird und eine Aussage zur

[2] Advertised Receiver Window Credit

aktuellen Größe des Empfangsfensters beim Datentransfer erlaubt. Der neu berechnete Wert wird unter Verwendung des Sack-Chunks ausgetauscht.

Als letzter Pflichtparameter wird der *Initial-Tag* (Die initiale Prüfmarke) besprochen. Hierfür wird die Kenntnis des *Transmission-Control-Blocks* vorausgesetzt, der vorher eingeführt wird.

Definition 6.2.3 (TCB) *Sender und Empfänger erzeugen beim Übergang zu einer gültigen Assoziation eine Datenstruktur, die alle relevanten Daten einer Assoziation enthält. Diese Struktur heißt* Transmission-Control-Block *oder kurz* TCB.

Was unter allen relevanten Daten zu verstehen ist, ist nicht eindeutig festgelegt. In Abschnitt 13 „Recommended Transmission Control Block (TCB) Parameters“ der Referenz [Stewart 2007] werden mögliche Werte vorgeschlagen. Für die folgenden praktischen Auswertungen wurde die *SCTPLib-Implementierung* verwendet, auf die in Teil III dieser Arbeit eingegangen wird. Daher kann die konkrete Umsetzung des TCB für die SCTPLib kann in der Dokumentation [Tüxen] nachgelesen werden.

Definition 6.2.4 (Initiation-Tag) *Der* Initiation-Tag *ist eine zufällige Zahl, die der Empfänger des Chunks in seinen TCB aufnimmt und im SCTP-Protokoll-Kopf als Prüfmarke – Verification-Tag – verwendet. Sämtliche Pakete, die der Empfänger des Chunks an den Sender des Chunks übermittelt, müssen diesen vom Sender des Chunks festgelegten Wert als Verification-Tag enthalten.*

Der Begriff des Initiation-Tags wird von Sender und Empfänger gleichermaßen verwendet, da der Empfänger im Gegenzug mit dem Init-Ack-Chunk seinen Initiation-Tag an den Sender übermittelt, der diesen seinerseits als Verification-Tag in die von ihm zum Versand vorgesehenen Pakete einstellt. Pakete ohne gültigen Verification-Tag werden abgelehnt. Mit dem Verification-Tag können Angriffe verhindert werden, bei denen ein Angreifer „blind“ Daten einschleust. Wie der Verification-Tag diese Angriffsform unterbindet, wird in Abschnitt 6.2.4 unter dem Stichwort Sicherheitsmaßnahmen erläutert.

Zustandsdiagramm nach Absenden des INIT-Chunks

Nachdem die Felder und Parameter des INIT-Chunks besprochen wurden, kann jetzt auf die Zustandsveränderung der Endpunkte eingegangen werden. Das zugehörige Diagramm ist in Abbildung 6.4 dargestellt.

Am Anfang befindet sich der Sender und Empfänger im Zustand *CLOSED*, sprich: es ist noch keine Assoziation und noch kein Wunsch, eine solche aufzubauen, vorhanden. Nachdem der Sender alle notwendigen Felder des INIT-Chunks belegt hat, wird der Chunk in ein SCTP-Paket verpackt und an den Empfänger gesendet. Da die Möglichkeit besteht, dass das Paket verloren geht, ist eine eventuelle Neuübertragung vorgesehen. Hierzu wird

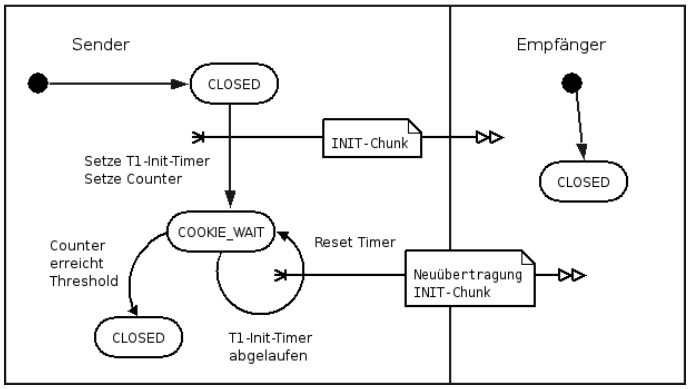


Abbildung 6.4: Zustandsdiagramm – Absenden des INIT-Chunks, um eine Assoziation zu etablieren

nach dem Absenden ein Timer gestartet, der nach einer bestimmten vorgegebenen Zeit erlischt, wenn der INIT-ACK-Chunk vom Empfänger noch nicht eingegangen sein sollte. Der Timer, der hier verwendet wird, wird als *T1-Init-Timer* bezeichnet.

Wenn der T1-Init-Timer erloschen ist, wird angenommen, dass der INIT-Chunk den Empfänger nicht erreicht hat. In diesem Fall wird eine Kopie des bereits gesendeten SCTP-Pakets angefertigt und übertragen. Der T1-Init-Timer wird zurückgesetzt, wobei die Ablaufzeit verdoppelt wird, wodurch dem Empfänger mehr Zeit für die Beantwortung des Inits gegeben und einer Überlastung des Netzes vorgebeugt wird.

Falls nach einer vorgegebenen Zeit keine Rückantwort eingetroffen ist, wird der gesamte Vorgang abgebrochen und in den Zustand *CLOSED* zurückgesprungen. Hierfür wird zusätzlich ein Zähler (Counter) kontinuierlich hochgezählt, der beim Erreichen eines Thresholds den Abbruch des Verbindungsaufbaus initiert.

Optionale Parameter des INIT-Chunks

Für den INIT-Chunk stehen einige optionale Parameter zur Verfügung, die hier nicht alle aufgeführt werden sollen. Ein Aspekt ist aber entscheidend für das Multihoming bzw. den in diesem Kapitel noch betrachteten konkurrierenden Transfer von Daten über mehrere Pfade.

Um Multihoming zu realisieren, werden einem *multihomed* Endpunkt mehrere Adressen zugewiesen. Eine Adresse kann auf verschiedene Arten dargestellt werden, wobei SCTP neben der Angabe einer *IPv4* oder *IPv6*-Adresse auch logische Adressen in Form des *Hostnamen* verarbeiten kann, sofern die Anbindung an einen DNS-Server gegeben ist. Bei einer Standard-Verbindung mit einer Adresse ist diese bereits im SCTP-Protokollkopf in der Sendeadresse kodiert. Sollen dem Endpunkt weitere Adressen zugeordnet werden,

geschieht dies über optionale Parameter.

Für jeden verwendbaren Adress-Typ steht ein Parameterbereich zur Verfügung. So kann der Parameterbereich *Hostname address*(0x000b) einen Hostnamen in Form eines Strings aufnehmen, während der Parameterbereich *IPv4 address*(0x005) und *IPv6 address*(0x006) die entsprechende Kodierung einer IP-Adresse als Nutzdaten übermitteln kann.

Als weiterer Parameter im Zusammenhang mit den zusätzlichen Adressen gibt es den *Supported address type*-Parameter (0x000c), mit dem die für die Assoziation möglichen Adress-Typen ausgehandelt werden. Dies hat u.a. auch Auswirkungen auf Erweiterungen wie ADD-IP, da hier die Adressen einer etablierten Assoziation verändert bzw. ausgetauscht werden können. Alle unterstützten Adress-Typen werden in Form ihres o.a. Typs hintereinander in den Nutzdatenbereich des Parameterblocks eingetragen, wobei auf das mögliche Padding geachtet werden muss, wenn nur ein oder alle drei Adressarten Verwendung finden sollen.

Einschub: Ein Cookie und der Transmission-Control-Block

Nachdem der Init-Chunk beim Empfänger eingegangen ist, quittiert er diesen mit dem INIT-ACK-Chunk. Die Parameter entsprechen weitgehend den Parametern des INIT-Chunks. Die optionalen Parameter des INIT-Chunks sind auch beim INIT-ACK-Chunk anwendbar, d.h. die zusätzlichen Adressen für das Multihoming auf Empfängerseite können mit dem Quittungssatz ausgehandelt werden.

Hinzu kommt ein *Cookie* (engl. State Cookie), der zum Schutz gegen Denial-of-Service-Attacken eingeführt wurde. Der als *Syn-Flood-Angriff* bekannte Angriff auf TCP kann unter Verwendung von SCTP verhindert werden. Wie dieser Angriff gegen TCP geführt und wie der hier eingeführte Cookie als Gegenmaßnahme eingesetzt wird, wird in Abschnitt 6.2.4 über Sicherheitsmaßnahmen diskutiert.

Was muss der Empfänger grundsätzlich machen, wenn er einen INIT-Chunk erhält und damit um Aufnahme einer Verbindung gebeten wird? Falls er seinerseits die Verbindung aufnehmen möchte, muss er seine Informationen über die Verbindung in Form des TCBs bereitstellen und mit dem INIT-ACK-Chunk den INIT-Chunk quittieren. Für die Speicherung des TCBs werden Ressourcen beim Empfänger belegt. Programmtechnisch werden die Informationen über die aufzubauende Assoziation über C-Strukturen im Hauptspeicher abgelegt.

Um den Verbindungsaufbau abschließen zu können, werden diese Informationen nach Erhalt des folgenden Chunks – COOKIE-ECHO-Chunk – benötigt. Es kann nicht sichergestellt werden, dass dieser Folgechunk überhaupt beim Empfänger eintrifft. Dies kann zum einen auf Netzprobleme, zum anderen aber auch auf böswilliges Unterlassen zurückzuführen sein. Ziel des Empfängers ist es demnach, keine Ressourcen für eine

halboffene Verbindung zu belegen, die möglicherweise nicht abgeschlossen werden kann. Daher sendet der Empfänger den INIT-ACK-Chunk als Quittierung an den Sender und löscht alle Informationen über die aufzubauende Verbindung, d.h. er löscht die TCB-Struktur vollständig aus dem Speicher.

Wenn keine weiteren Nachrichten vom Sender eingehen, stellt dies auf Empfängerseite kein Problem dar, da er sich nach dem Versenden des INIT-ACK-Chunks im selben Zustand wie vor der Kontaktaufnahme durch den Sender befindet. Wenn der Sender den Verbindungsaufbau korrekt abschließen möchte, geht der folgende COOKIE-ECHO-Chunk ein.

Wie kommt der Empfänger jetzt an seine TCB, den er ja gelöscht hat? Hier kommt der *Cookie* ins Spiel. Der Empfänger sichert den TCB und damit alle verbindungsrelevanten Daten in einem Cookie und sendet diesen mit dem INIT-ACK-Chunk an den Sender, der seinerseits den Cookie als Parameter im COOKIE-ECHO-Chunk zurücksendet. Der Cookie fungiert hier sozusagen als „kleine“ Datenbank, in der der TCB abgespeichert wird. Somit ist der Empfänger in der Lage, nach Eingang des COOKIE-ECHO-Chunks seinen TCB aus dem Cookie zu rekonstruieren. Erst jetzt werden die Ressourcen dauerhaft – bis die Assoziation wieder abgebaut wurde – belegt.

Jetzt stellt sich ein neues Problem! Wie kann sichergestellt werden, dass der Sender auch die Orginaldaten im Cookie überträgt? Ohne zusätzliche Maßnahme kann der Empfänger nicht feststellen, ob der Cookie auch die von ihm hinterlegten Daten enthält, da ja alle relevanten Daten gelöscht wurden. Um dieses Problem zu lösen, greift man in die Kryptographie-Trickkiste, indem der Empfänger den Cookie für eine spätere Authentifizierung signiert, bevor er ihn absendet und den TCB löscht. Somit braucht der Empfänger nur die Signatur des Cookies vor der erneuten Generierung des TCBs zu überprüfen, um festzustellen, ob es sich um die Informationen handelt, die er selber festgelegt hat.

Welche Informationen werden im Cookie zwischengespeichert? Hier ist die Spezifikation im RFC [Stewart 2007] nicht eindeutig, sodass es auf die Implementierung ankommt. Für das Zusammenspiel mit anderen SCTP-Varianten ist eine eindeutige Festlegung auch nicht notwendig, da der TCB nur lokal ohne Außenwirkung beim entsprechenden Endknoten verwaltet wird. Gleiches gilt für die erwähnte Authentifizierung des Cookies; da die Abarbeitung des Vorgangs vollständig beim Empfänger erfolgt, können einzelne Implementierungen voneinander abweichen. In diesem Zusammenhang sei darauf hingewiesen, dass das für das Signieren notwendige Passwort auch nur auf Seiten des Empfängers benötigt wird und daher als geheimes Passwort beim Empfänger hinterlegt wird.

In [Stewart and Xie 2001] werden folgende Felder als Grundbaustein für den Cookie vorgeschlagen. Es fließen Informationen über den INIT-Chunk sowie über den zu bildenden INIT-ACK-Chunk in den Cookie ein. Die Informationen vom INIT-Chunk werden verwendet, um sicherzustellen, dass die beim Aushandeln verwendeten Parameter nicht

verändert wurden, während der INIT-ACK-Chunk den TCB enthält, da dies den Werten entspricht, die der Empfänger zur Aushandlung beigesteuert hat. Der Cookie selber bleibt dabei selbstverständlich außen vor.

Zudem wird eine Zeit (engl. time-to-live) vereinbart, während der der Cookie gültig sein soll, die ebenfalls in den Cookie eingeht. Dieser Wert ermöglicht es dem Empfänger einen bereits sehr lange zurückliegenden Verbindungswunsch abzulehnen.

Als weiterer Eintrag werden die sogenannten *Tie-Tags* in den Cookie mit aufgenommen. *Tie-Tags* werden in [Vanit-Anunchai 2008] als die Kopie der Verification-Tags bezeichnet. Verwendet werden Tie-Tags, um bestimmte komplexe Situationen, die während der Lebenszeit einer Assoziation auftreten können, wie beispielsweise ein *Assoziations-Neustart*, zu erkennen und entsprechend zu behandeln. Die Verwendung und Speicherung von Tie-Tags weicht im neueren Standard in [Stewart 2007] von der vorher gültigen Version aus [Stewart et al. 2000] ab, sodass man bei Verwendung von Tie-Tags auf die Version der Implementierung achten sollte. An dieser Stelle wird auf Tie-Tags nicht weiter eingegangen, da sie für das grundsätzliche Verständnis von SCTP und auch für die folgenden Überlegungen auf dem Gebiet des IN nicht benötigt werden.

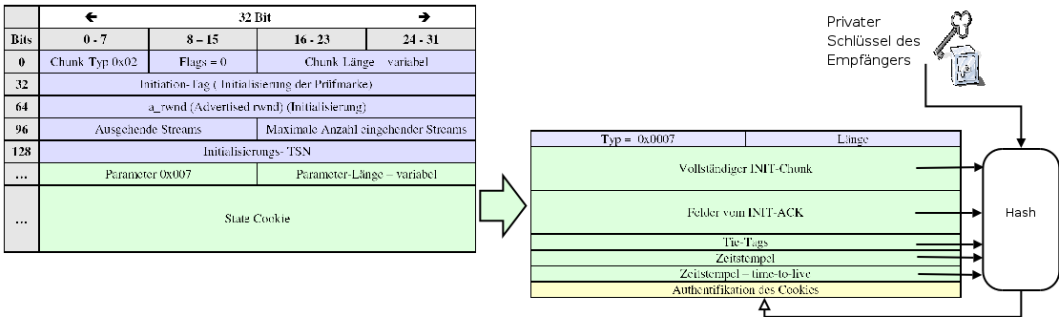


Abbildung 6.5: Der Cookie als Parameter im INIT-ACK-Chunk

Die Generierung und Einbettung des Cookies in den Parameterbereich des INIT-ACK-Chunks ist in Abbildung 6.5 zusammengefasst. Man erkennt die notwendigen Werte des Cookies, die unter Verwendung eines *Key-Hash-Verfahrens* und dem privaten Schlüssel des Empfängers auf einen 32-Bit großen Authentifikationswert abgebildet werden. Einzelheiten zum „Hashing“ werden im folgenden Einschub besprochen, hier wird das grundsätztliche Vorgehen betrachtet. Der Cookie enthält somit den „gehashten“ Bereich und den zugehörigen Authentifikationswert.

Einschub: Authentifikation mittels MAC, HMAC und Hashfunktionen

Wie und mit welchen Verfahren die Authentifikation des Cookies vorzunehmen ist, ist in der Referenz [Stewart 2007] nicht abschließend spezifiziert. Bei der Definition des

Message Authentification Code(MAC) in Abschnitt 1.3 der Referenz wird lediglich auf den RFC2104 – [Krawczyk et al. 1997] – verwiesen, der wiederum den Begriff des *HMAC* einführt. Daraus kann geschlossen werden, dass dieses Verfahren zur Authentifizierung des Cookies empfohlen wird. Die Begriffe werden im Folgenden kurz vorgestellt, wobei für die den Verfahren zugrunde liegenden mathematischen Beweisen auf die Literatur verwiesen wird. Die grundsätzlichen kryptographischen Funktionen können u.a. in [Menezes et al. 1996] oder [Wätjen 2003] nachgelesen werden.

Definition 6.2.5 (Hashfunktion) *Sei x eine Eingabe mit einer endlichen beliebigen Bitlänge. Eine Funktion h heißt* Hashfunktion*, falls*

(1) $y = h(x)$, mit fester Bitlänge von y, gilt (Kompression) *und*

(2) $h(x)$, bei gegebenen x und h, einfach zu berechnen ist. (Einfachheit)

Der Hashwert $h(x)$ wird häufig in Analogie zur „realen Welt“ als Fingerabdruck (engl. fingerprint) der Eingabe x bezeichnet. Eine „einfache“ Berechnung, wie sie in der zweiten Bedingung gefordert ist, ist genau dann erfüllt, wenn die Rechnung in polynomialer Zeit in Bezug auf die Bitlänge der Eingabe x durchführbar ist.

Eine Hashfunktion nach Definition 6.2.5 erfüllt noch keine kryptographischen Bedingungen, die verhindern, dass der Fingerabdruck gefälscht werden kann.

Definition 6.2.6 (Stark kollisionsfreie Hashfunktion) *Eine Hashfunktion h heißt* stark kollisionsfrei*, falls es berechnungsmäßig praktisch unmöglich ist, zwei Werte x und x' mit $x \neq x'$ und $h(x) = h(x')$ zu finden.*

Aus der *starken Kollisionsfreiheit* ergibt sich direkt der Begriff der fälschungssicheren oder auch *kryptographischen Hashfunktion.*

Definition 6.2.7 (kryptographische Hashfunktion) *Eine stark kollisionsfreie Hashfunktion h heißt* kryptographische Hashfunktion*.*

Häufig werden zwei zusätzliche Forderungen, nämlich die *schwache Kollisionsfreiheit* und die *Einweg-Funktions-Eigenschaft*, an eine kryptographische Hashfunktion gestellt, die hier der Vollständigkeit halber aufgeführt werden sollen. Für die Definition der kryptographischen Hashfunktion sind diese Zusatzbedingungen nicht notwendig, da aus der starken Kollisionsfreiheit auch die schwache Kollisionsfreiheit sowie die Eigenschaft der Einwegfunktion folgt. Die zugehörigen Beweise können u.a. [Wätjen 2003] entnommen werden.

Definition 6.2.8 (Schwache Kollisionsfreiheit) *Eine Hashfuntkion h heißt* schwach kollisionsfrei *für ein gegebenes x, falls es berechnungsmäßig praktisch unmöglich ist, ein x' zu finden, für das $h(x) = h(x')$ gilt.*

Mit Hilfe der Einweg-Eigenschaft lässt sich die Cookie-Authentifikation anschaulich erklären. Der Sender darf nicht in der Lage sein, einen neuen veränderten Cookie so zu konstruieren, dass der Empfänger ihn für seinen eigenen versendeten hält.

Definition 6.2.9 (Einweg-Funktion) *Eine Hashfunktion h ist eine Einweg-Funktion, falls es für einen Fingerabdruck y berechnungsmäßig praktisch nicht möglich ist, ein x mit $h(x) = y$ zu bestimmen.*

Mit Hilfe von kryptographischen Hashfunktionen kann die Integrität von Daten sichergestellt werden. Sei beispielsweise x ein Cookie, von dem der Empfänger den Hashwert $y = h(x)$ mit einer kryptographisch sicheren Hashfunktion gebildet hat. Aufgrund der geforderten starken Kollisionsfreiheit führt eine Änderung der Cookiedaten $x' \neq x$ mit sehr großer Wahrscheinlichkeit dazu, dass der Hashwert $y' = h(x')$ ungleich y ist. Somit ist der Empfänger in der Lage, die Veränderung zu erkennen und den Verbindungsaufbau abzubrechen. Was aber ist, wenn der Sender nicht nur die Daten des Cookies verändert, sondern auch einen anderen Hashwert an den Cookie anhängt? Da der Empfänger über keine Informationen mehr über den ersten Teil der Initialisierungsphase verfügt, könnte er den Betrug nicht feststellen. Daher muss zusätzlich die Authentifizierung des Cookies gegeben sein.

Die Verwendung der Authentifizierung ist ungewöhnlich, da der Empfänger sich nicht gegenüber jemand anderem authentifiziert, sondern gegen sich selber. Dies macht aber für das grundsätzliche Vorgehen keinen Unterschied. Um die Authentifizierung zu gewährleisten, wird der Hashwert zusätzlich mit einem Schlüssel parametrisiert. Dieses Vorgehen wird häufig auch als *Keyed Hashing* (siehe u.a. [Doraswamy und D. 2000]) bezeichnet. Dies führt zum Begriff des Message-Authentication-Codes (MAC).

Definition 6.2.10 (Message-Authentication-Code (MAC)) *Sei $\mathcal{S}$ eine Menge von Schlüsseln, $\{ h_s | s \in \mathcal{S}\}$ eine Familie von Hashfunktionen, x eine Eingabe beliebiger, endlicher Bitlänge und $\mathcal{M} = \{(x_i, h_s(x_i)) \mid i \in \mathbb{N}\}$ eine Menge von bekannten Wertepaaren. Ein* Message-Authentication-Code (MAC) *liegt vor, falls*

(1) $y = h_s(x)$, mit fester Bitlänge von y, gilt – Kompression – *und*

(2) $h_s(x)$ bei gegebenen x, $s \in \mathcal{S}$ und h_s einfach zu berechnen ist – Einfachheit – *und*

(3) es berechnungsmäßig praktisch unmöglich ist, bei unbekanntem Schlüssel $\mathcal{S}$ aus $\mathcal{M}$ einen gültigen Wert $h_s(x)$ für ein $x \neq x_i \forall i$ zu bestimmen. – Fälschungsresistenz –

Man erkennt, dass ein MAC sich dadurch auszeichnet, dass die Bedingungen, die an eine kryptographische Hashfunktion gestellt werden, auch für die Hashfunktionen mit zusätzlichem Schlüssel gelten. Für die Authentifikation des Cookies bedeutet dies, dass ein MAC neben der Datenintegrität unter der Voraussetzung, dass der Sender den privaten Schlüssel des Empfängers nicht kennt, auch die Datenautentifikation gegeben ist. Der Empfänger ist demnach in der Lage festzustellen, dass er die Daten auch wirklich in dieser Form gesendet hat.

Um den theoretischen Teil abzuschließen, wird der *HMAC*, wie er im RFC [Krawczyk et al. 1997] verwendet wird, eingeführt. Ein HMAC ist ein MAC, der auf einer beliebigen kryptograpisch sicheren Hashfunktion aufsetzt. Häufig eingesetzte Hashfunktionen

als Basis für den HMAC sind die Hashfunktionen SHA und $MD5$, wobei die $MD5$-Variante nicht mehr uneingeschränkt als sicher gilt (vgl. hierzu [Wang und Yu 2005]. In Abbildung 6.5 ist für eine sichere Authentifizierung des Cookies als Hash-Operator ein HMAC zu nehmen. In [Menezes et al. 1996] wird folgende Konstruktion vorgeschlagen:

Definition 6.2.11 (HMAC) *Sei x eine Eingabe beliebiger, endlicher Bitlänge, $s \in \mathcal{S}$, h eine Hashfunktion und (p_1, p_2) Fülldaten (padding), die geeignet sind, den Schlüssel s auf die Eingabeblocklänge der Hashfunktion aufzufüllen, dann kann ein* HMAC *wie folgt konstruiert werden:*

$$HMAC(x) = h_s(x) = h(\ s \ ||\ p_1 \ ||\ h(s \ ||\ p_2 \ ||\ x))$$

Zustandsdiagramm nach Absenden des INIT-ACK-Chunks

In Abbildung 6.6 ist das erweiterte Zustandsdiagramm angegeben. Zu beachten ist, dass der Empfänger auch nach dem Absenden des INIT-ACK-Chunks im CLOSED-Zustand ist. Da er die Daten, die er zur Generierung des Cookies erzeugt hat, sofort wieder entfernt, stellt sich für den Empfänger die Situation so dar, als wäre nie eine Anfrage zum Verbindungsaufbau bei ihm eingetroffen.

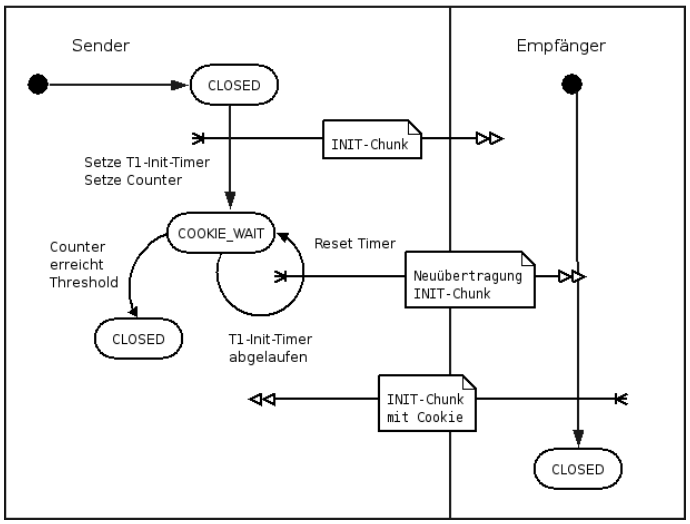


Abbildung 6.6: Zustandsdiagramm – Absenden des INIT-ACK-Chunks mit authentifiziertem Cookie

Damit ist der erste „Zwei-Wege-Handshake“ abgearbeitet. Die folgenden zwei Pakete stellen die Verbindung her.

6.2.2 Die Assoziation etablieren

Mit den nächsten zwei gesendeten Paketen wird die Initialisierung abgeschlossen und die Assoziation etabliert. In diesen Schritten können bereits Daten zusammen mit den Kontrollinformationen gesendet werden.

Der Cookie wird zurückgesendet

Der Sender, der seinen TCB allociert hat, überprüft bei eingehendem INIT-ACK-Chunk, ob ihm die zugehörige Assoziation bekannt ist und ob der Verification-Tag korrekt belegt wurde. Bei der Überprüfung der Daten ist zu beachten, dass der INIT-ACK-Chunk nicht zwingend mit der Adresse versendet wurde, an die der INIT-Chunk gesendet wurde, da aufgrund der Multihoming-Eigenschaft der Empfänger mehrere Pfade und damit Adressen zur Auswahl hat. Für den Sender heißt das, dass er die Adress-Parameter des INIT-ACK-Chunks auswerten und beachten muss. Da der T1-Init-Timer nicht mehr benötigt wird, weil die Antwort bereits eingegangen ist, wird dieser zurückgesetzt und damit der Neuübertragungs-Mechanismus für den INIT-Chunk ausgesetzt.

Als Antwort auf den INIT-ACK-Chunk sendet der Sender ein COOKIE-ECHO, das, wie der Name sagt, eine Kopie des Cookies enthält. Dazu wird der Cookie, der als Parameterbereich vorliegt, durch eine einfache Änderung des Feldes Parameter-Typ in einen gültigen COOKIE-ECHO-Chunk transformiert. Der Parameter-Typ des Cookies wird mit zwei Werten aufgefüllt, zum einen mit dem COOKIE-ECHO-Chunk-Typ, in diesem Fall $0x0a$, und zum anderen mit einer Chunk-Flag, die auf 0 gesetzt wird.

Diesen so generierten COOKIE-ECHO-Chunk können bereits Nutzdaten in Form von DATA-Chunks angehängt werden, die, falls der Empfänger unseren Cookie akzeptiert, vom Empfänger verarbeitet werden können. Der Aufbau des Daten-Chunks sowie der Ablauf der Übertragung wird in Abschnitt 6.3 behandelt, an dieser Stelle soll nur festgehalten werden, dass ein Zeitgewinn durch das Mischen von Kontroll- und Daten-Chunks bereits beim Verbindungsaufbau erreicht werden kann.

Betrachtet man die Zustände der einzelnen Endpunkte, wie sie in Abbildung 6.7 schematisch dargestellt sind, ergibt sich folgendes Bild. Nachdem der COOKIE-Echo-Chunk abgesendet wurde, wechselt der Sender in den *COOKIE-ECHOED-Zustand* und setzt mit dem *T1-Cookie-Timer* den Neuübertragungsüberwachungs-Prozess in Gang, der nach der Systematik wie beim Versand des INIT-Chunks abläuft. Der Empfänger befindet sich zu diesem Zeitpunkt immer noch im Zustand *CLOSED*.

Die Initialisierung abschließen

Als Antwort auf das COOKIE-ECHO sendet der Empfänger einen COOKIE-ACK-Chunk, der keine neuen Informationen enthält und lediglich dazu dient, den Sender über die korrekte und vollständige Initialisierung zu informieren.

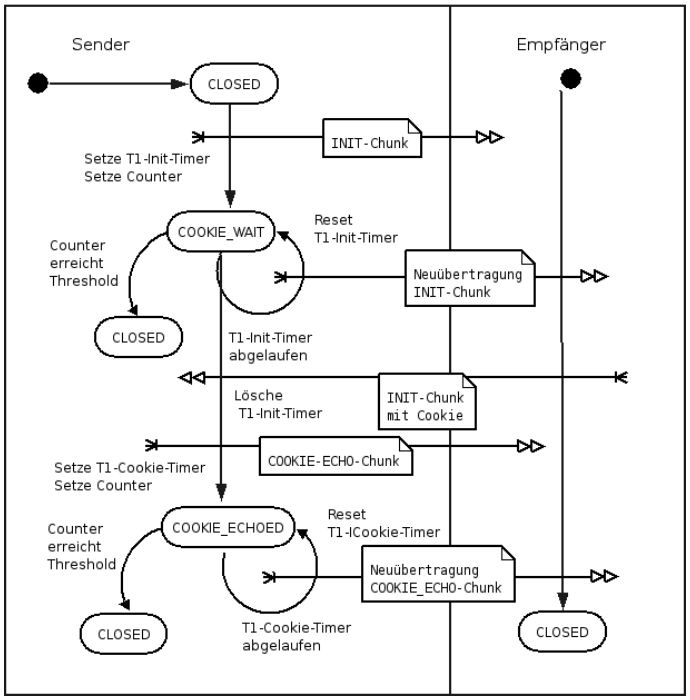


Abbildung 6.7: Zustandsdiagramm – Zurücksenden des Cookies mit dem COOKIE-ECHO-Chunk

Für den Empfänger beginnt jetzt aber erst die Arbeit, da er auf Basis des Cookies seinen TCB aufbauen muss. Hierfür muss er als Erstes die Authentifikation des Cookies überprüfen. Dafür erstellt er den Authentifikation-Wert unter Verwendung seines privaten Schlüssels nochmal neu, diesmal auf Basis des zurückgelieferten Cookies. Wenn der mitgelieferte Authentifikations-Wert mit dem neu berechneten übereinstimmt, ist davon auszugehen, dass die Cookie-Daten nicht verändert und ursprünglich von ihm festgesetzt wurden. Zusätzlich wird überprüft, ob der Cookie noch gültig ist, d.h. es wird geprüft, ob die Live-Time des Cookies nicht bereits abgelaufen ist.

Falls die Überprüfung scheitert, sendet der Empfänger einen ERROR-Chunk mit dem Fehlerparameterbereich *State Cookie Error*(0x003) und beendet die Initialisierung. Andernfalls werden die Informationen aus dem Cookie genutzt, um den TCB zu generieren und wechselt in den Zustand *ESTABLISHED*. An dieser Stelle kann mit der Verarbeitung möglicher mitgelieferter Daten-Chunks begonnen werden. Der genaue Ablauf wird in Abschnitt 6.3 besprochen, hier sei nur erwähnt, dass mit dem abschließenden ECHO-ACK-Chunk bereits Quittierungs-Chunks, die sogenannten SACKs mit dem SCTP-Paket an den Sender übermittelt werden können. Dies bedingt natürlich, dass die Verarbeitung der Daten-Chunks vor dem Zustand *ESTABLISHED* des Senders durchgeführt werden müssen. Es kann natürlich auch vorkommen, dass der Empfänger bereits Daten für

den Versand an den Sender aufgespart hat, diese können ebenfalls zusammen mit dem COOKIE-ACK-Chunk in einem SCTP-Paket übertragen werden.

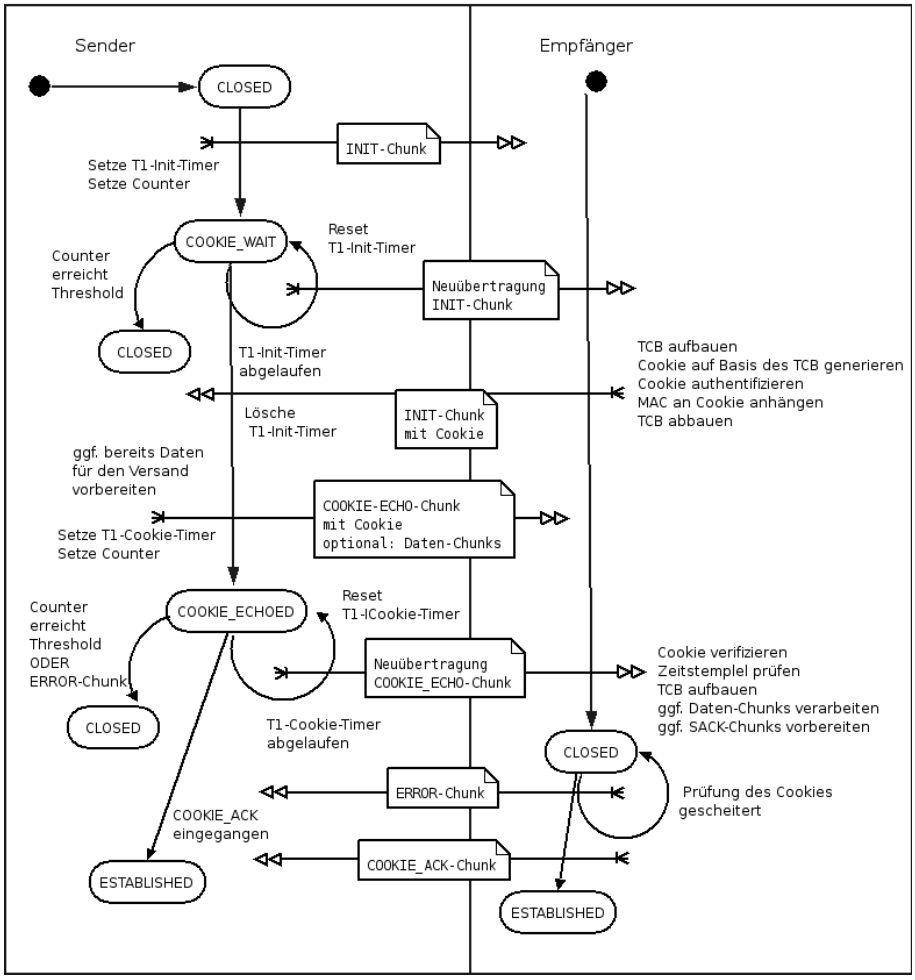


Abbildung 6.8: Zustandsdiagramm – Das vollständige Vier-Wege-Handshake

Das jetzt vollständige Zustandsdiagramm des Vier-Wege-Handshakes ist in Abbildung 6.8 dargestellt. Nach Erhalt des COOKIE-ACK-Chunks geht auch der Sender in den Zustand *ESTABLISHED* über und kann mit der Verarbeitung der möglicherweise mitgelieferten Daten- und SACK-Chunks beginnen. Die Assoziation konnte erfolgreich etabliert werden.

6.2.3 Verbindungsabbau

Der Verbindungsabbau kann grundsätzlich auf zwei verschiedene Arten erfolgen, entweder durch einen gewollten Abbruch, dem sogenannten *Shutdown*, oder aufgrund einer

Fehlersituation durch *Abort*.

Graceful Shutdown - ordentliche Beendigung der Verbindung

Anders als beim Konkurrenten TCP kann es bei SCTP nicht zu halboffenen Verbindungen kommen, d.h. nachdem die Verbingung beendet wurde, werden keine Daten mehr in das Netz eingespeist. Ausstehende Daten werden demnach vor dem endgültigen Verbindungsende ausgetauscht und quittiert. Um dies zu erreichen, verwendet SCTP einen dreistufigen Shutdown-Prozess, der auch als *Gracefull Shutdown* bezeichnet wird.

Auch beim Shutdown durchlaufen die einzelnen Endpunkte bestimmte Zustände, bis sich die Endpunke nach erfolgreicher Beendigung im Zustand *CLOSE* befinden. Bis dahin werden die Zustände *SHUTDOWN_SENT*, *SHUTDOWN_PENDING* beim Sender und *SHUTDOWN_RECEIVED* und *SHUTDOWN_ACK_SENT* beim Empfänger durchlaufen, die im Folgenden erläutert werden. Analog zum Verbindungsaufbau in Abschnitt 6.2.1 wird der Endpunkt, der den Shutdown einleitet, als *Sender* und der reagierende Endpunkt als *Empfänger* bezeichnet.

In der Einleitung wurde von „ausstehenden Daten“ gesprochen, die vor der endgültigen Beendigung der Verbindung noch vollständig übermittelt werden müssen. Da sich die Daten in verschiedenen Stadien des Transports befinden können, muss der Begriff der ausstehenden Daten weiter verfeinert werden, wobei weitgehend der englische Sprachgebrauch beibehalten wurde, da eine direkte Übersetzung der Begriffe nur unter Verlust des Verständnisses möglich erscheint.

Definition 6.2.12 (Ausstehende Daten) *Folgende* ausstehende Daten *können unterschieden werden:*

(1) Als User-Daten *werden die Daten bezeichnet, die von der Anwendung – sprich: dem ULP*[3] *– an SCTP herangetragen werden.*

(2) Als Outbound-User-Daten *werden die Daten bezeichnet, die bereits vom SCTP-Protokoll übernommen wurden, sich in einem eindeutigen Status befinden und noch nicht vom Endpunkt als eingetroffen quittiert wurden. Outbound-Daten können folgende zwei Zustände annehmen:*

(a) Unter Outstanding-User-Daten *versteht man die Daten, die bereits gesendet, aber noch nicht mittels SACK-Chunk quittiert wurden.*

(b) Unter Pending-User-Daten *versteht man die Daten, die von SCTP bereits in der Sendqueue untergebracht wurden, die aber noch nicht abgesendet wurden.*

Ausgangspunkt für den ordentlichen Shutdown ist die Mitteilung des ULP an den Sender, dass er den Abbruch der Verbindung wünscht. Ab diesem Zeitpunkt wechselt der

[3] Upper Layer Protocol

Sender vom Zustand *ESTABLISHED* in den Zustand *SHUTDOWN-PENDING* und nimmt keine weiteren Daten mehr von der Anwendung entgegen. Bevor er allerdings über einen *SHUTDOWN*-Chunk den Empfänger davon in Kenntnis setzt, arbeitet er erst seine Outbound-Daten ab, d.h. der Shutdown wird erst eingeleitet, wenn sämtliche Daten, die noch vom Sender verwaltet werden, quittiert wurden. Zu diesem Zeitpunkt hat der Sender in Bezug auf die Daten einen sauberen Zustand erreicht. Anders sieht es beim Empfänger aus, der von dem eigentlichen Shutdown noch keine Kenntnis erlangt hat. Um auch hier einen sauberen Zustand zu erreichen, werden beim Shutdown-Prozess diese Daten abgearbeitet. Der hier beschriebene Ablauf kann im Zustandsdiagramm 6.9 mitverfolgt werden.

Jetzt teilt der Sender dem Empfänger den Wunsch des Verbindungsabbaus unter Verwedung des *SHUTDOWN*-Chunks mit. Gleichzeitig geht der Sender in den Zustand *SHUTDOWN-SENT* über. Der Shutdown-Chunk enthält als einzigen Parameter die Commulative-TSN, wie sie in Definition 6.1.10 beschrieben ist. Der Empfänger verwendet die Coummulative-TSN für die Verwaltung der Neuübertragungen bei der Restabwicklung der Assoziation.

Der Empfänger seinerseits nimmt den SHUTDOWN-Chunk entgegen, wechselt den Zustand von *ESTABLISHED* auf *SHUTDOWN-RECEIVED* und blockiert die Annahme von User-Daten vom ULP. Ansonsten läuft die Verarbeitung *fast normal* weiter, d.h. es werden Bestätigungen für beim Sender eingegangene Daten ausgewertet und die noch ausstehenden Daten an den Sender übermittelt. Unterschiede gibt es bei der Bewertung der Daten sowie bei der Art und Weise, wie der Sender eingehende Daten-Chunks quittiert.

Im *SHUTDOWN-SENT*-Zustand quittiert der Sender die eingehenden Daten nicht mehr ausschließlich mit dem SACK-Chunk. Falls die Übertragung reibungslos verläuft, d.h. keine Lücken bei den eintreffenden TSNs auftreten, beantwortet der Sender die Daten-Chunks unter Verwendung des SHUTDOWN-Chunks. Somit hat der SHUTDOWN-Chunk zwei Bedeutungen, zum einen leitet er die Shutdown-Phase ein, zum anderen fungiert er als Quittierungssatz. Im Falle, dass Lücken bei den quittierten TSNs auftreten, also einige ältere Daten-Chunks noch ausstehen, werden neben dem SHUTDOWN-Chunk auch ein SACK-Chunk zur Quittierung verwendet, die im Normalfall zusammen in einem SCTP-Paket versendet werden.

Laut [Stewart and Xie 2001] wird die Verwendung des Shutdown-Chunks zur Quittierung damit begründet, dass es Konstellationen gibt, bei denen der Shutdown-Chunk den Empfänger nicht erreicht und der Empfänger immer weiter neue Daten in das Netz einspeist. Damit würde der endgültige Abbruch der Verbindung nicht zustande kommen.

Hat der Empfänger seinerseits alle ausstehenden Daten abgearbeitet, teilt er dies dem Sender mit einem SHUTDOWN-ACK-Chunk mit und wechselt in den Zustand *SHUTDOWN-ACK-SENT* und wartet auf die Bestätigung des Senders, damit er die Verbin-

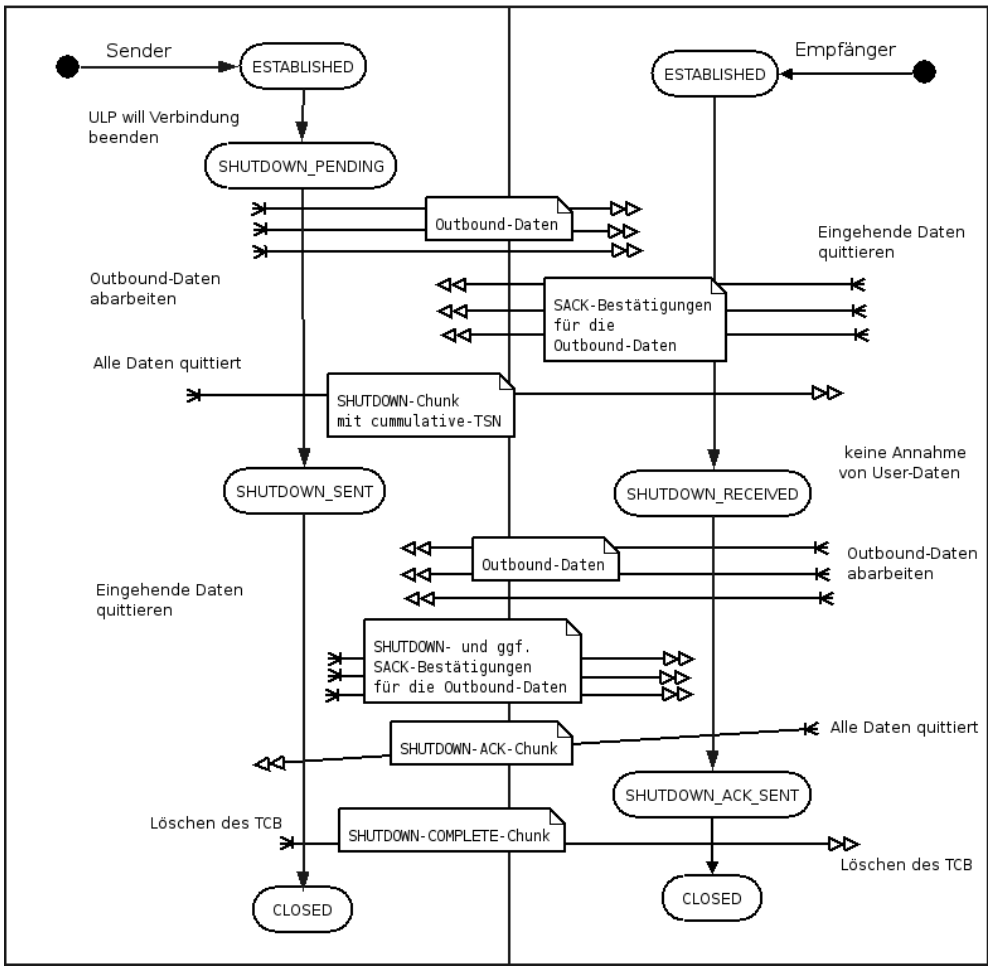


Abbildung 6.9: Der ordentliche Shutdown – Graceful-Shutdown

dung endgültig schließen kann.

Der Sender nimmt den SHUTDOWN-ACK-Chunk entgegen, sendet seinerseits den SHUTDOWN-COMPLETE-Chunk und löscht den gesamten TCB und damit alle mit der Assoziation verbundenen Daten. Jetzt kann er in den Zustand *CLOSED* wechseln. Der SHUTDOWN-ACK-Chunk verhält sich ein wenig anders als die anderen Chunks, da er der einzige Chunk ist, zu dem es, wenn er gesendet wird, keinen gültigen TCB beim Sender mehr gibt. Die klassische Variante, einen Timer zu setzen und ggf. den Chunk noch einmal neu zu übertragen, kann hier nicht zum Erfolg führen. Es können mehrere Fälle konstruiert werden, bei denen der Shutdown-Vorgang nicht ordnungsgemäß beendet werden kann, die in [Stewart and Xie 2001] ausführlich abgehandelt werden, hier sei lediglich der Standardfall thematisiert.

Es fehlen noch die Abschlussarbeiten beim Empfänger. Wenn dieser den SHUTDOWN-COMPLETE-Chunk erhält, löscht auch er seinen TCB und damit alle Informationen über die Assoziation und wechselt in den *CLOSED*-Zustand. Damit ist der Shutdown vollständig, und die Verbindung ist abgebaut.

Ähnlich wie beim Verbindungsaufbau werden auch beim Verbindungsabbau Timer und Counter zur Steuerung von möglichen Neuübertragungen bzw. zur Fehlererkennung verwendet. Insbesondere greift in der Phase der Restabwicklung von Daten ein Fehler-Counter, der bei Erreichen eines gegebenen Schwellwertes die Verbindung als *inaktiv* annimmt und die Löschung des TCB mit anschließendem Zustandswechsel in den *CLOSED*-Zustand vornimmt. Der Fehler-Counter wird bei Eintreffen von neuen Daten-Chunks beim Sender respektive SHUTDOWN- bzw. SACK-Chunks beim Empfänger immer wieder auf 0 zurückgesetzt, sodass unabhängig davon, wie viele Daten noch ausstehen, diese auch korrekt abgearbeitet werden können. Treffen allerdings keine Daten mehr ein, kann darauf geschlossen werden, dass die physische Verbindung nicht mehr besteht, so dass sich die Beendigung des Shutdown-Vorgangs als logische Folge ergibt.

Zudem wird der Verwendung von Multihomed-Adressen Rechnung getragen, indem bei einer Neuübertragung aufgrund von abgelaufenen Timern eine andere Zieladresse des Empfängers gewählt wird. So ist es möglich, auch bei Inaktivität des Primärpfades die Assoziation *ordentlich* zu beenden.

Abort – Abbruch der Verbindung

Es gibt Situationen, in denen es nicht möglich ist, den Shutdownprozess, wie er im vorherigen Teilabschnitt erläutert wurde, vollständig auszuführen. In diesen Fällen besteht die Möglichkeit eines „harten“ Abbruchs über den Abort-Shutdown. In [Stewart and Xie 2001] wird der unvollständige Shutdown als unzuverlässiger So-gut-wie-es-eben-geht-Versuch beschrieben, den Endpunkt von dem Niedergang der Assoziation in Kenntnis zu setzen.

Ein Abbruch der Verbindung kann zu jeder Zeit erfolgen, und zwar initiiert von der darüberliegenden Anwendung oder von SCTP selber. Falls eine Anwendung die Verbindung sofort ohne weiteren Datenverkehr beenden möchte, kann sie dies durch einen harten Abbruch erreichen. Als Beispiel für einen benutzerseitigen Abbbruch wird häufig das Herunterfahren des Systems genannt, das beim Shutdown nur noch in der Lage ist, eine Abort-Nachricht an alle offenen Assoziationen zu senden, aber nicht die Zeit vorhanden ist, sämtliche Ressourcen ordnungsgemäß freizugeben. Somit können die angeschlossenen Endpunkte zumindest zeitnah die entsprechende Assoziation freigeben und das ULP über die nicht mehr vorhandene Verbindung informieren.

In Fällen, in denen offensichtlich nicht korrekte Daten übergeben werden, kann SCTP über den Abort-Prozess eine Fehlermeldung transportieren, bevor die Asssoziation in die Wüste geschickt wird. Ein Beispiel, bei dem der Abort in der Spezifikation vorgesehen

ist, ist beispielsweise die Reaktion auf einen eingehenden INIT-Chunk, dessen Pflichtparameter mit ungültigen Werten versorgt sind. In einem solchen Fall wird lediglich ein ABORT-Chunk samt Fehlermeldung generiert und zum Sender zurückgeschickt. Die Spezifikation erlaubt den Abbruch auch für extreme Probleme während der Ausführung des Protokolls.

Da ein harter Abbruch keine Rücksicht auf den ausstehenden Datenverkehr oder den kontrollierten Abbau von Ressourcen nehmen muss, ist der Ablauf relativ einfach gestaltet. Sei der Sender der Endpunkt, der die Assoziation hart beenden möchte, und der Empfänger der Endpunkt, der von dem Abbruch überzeugt werden soll.

Der Sender sendet erst einen Abort-Chunk ggf. mit Fehlermeldung an den Empfänger und löscht seinen TCB, womit er alle Informationen über die Assoziation entfernt. Der Sender wird in den Zustand *CLOSED* versetzt. Der Empfänger seinerseits prüft den ankommenden Abort-Chunk anhand des Verification-Tags und beendet, falls dieser einen gültigen Wert beinhaltet, auch seinen Teil der Verbindung, d.h. er löscht ebenfalls den TCB und geht in den Status *CLOSED* über. Falls der Sender eine Fehlermeldung im Parameterbereich mitgeliefert hat, würde der Empfänger vor der Löschung der Daten diese an die Anwendung weiterreichen, damit diese entsprechend auf den Abbruch reagieren kann.

6.2.4 Sicherheitsmaßnahmen

Bei der Definition von SCTP konnte man auf jahrelange Erfahrungen mit TCP zurückgreifen und so Sicherheitsproblemen vorbeugen, die sich im Laufe der Zeit mit TCP ergeben haben. Der im Abschnitt 6.2.1 geschilderte Verbindungsaufbau sieht u.a. Maßnahmen gegen zwei typische Angriffe auf das TCP-Protokoll vor.

Blinde Angreifer

Ein Angreifer, der die Verbindung nicht abhören kann, sondern lediglich in der Lage ist fehlerhafte Pakete in das Netz oder genauer in die Assoziation einzuspeisen, wird auch als *blinder Angreifer* bezeichnet. Ein blinder Angreifer versendet ungültige Pakete an den Empfänger, die ohne zusätzliche Schutzmaßnahme als gültig erkannt werden könnten.

SCTP verwendet den Verification-Tag, wie er in Definition 6.2.4 beschrieben wurde, um diese ungültige Pakete zu erkennen und abzulehnen. Wie bereits beschrieben, wird ein Paket mit ungültigen Verification-Tag verworfen bevor mit der eigentliche Bearbeitung des Pakets beim Empfänger begonnen wird. Da es sich bei dem Verification-Tag um einen zufälligen 32-Bit-Wert handelt, ist ein blinder Angreifer im Worst-Case gezwungen 2^{31} Pakete zu senden, bevor der Empfänger ein ungültiges Paket tatsächlich akzeptiert.

SYN-Flood-Angriffe

In Definition 6.2.3 wurde der Transmission-Control-Block als Datenstruktur beschrieben, der alle relevanten Daten einer Assoziation enthält. Wird bereits bei der ersten Anfrage auf Verbindungsaufbau eines Senders beim Empfängers der TCB angelegt und die Daten beim Empfänger gespeichert, spricht man auch von einer *halboffenen Verbindung*.

Beim SYN-Flood-Angriff „flutet“ der Angreifer den Empfänger mit unzähligen Anfragen zum Verbindungsaufbau. Für jede dieser Anfragen werden bei Verwendung der halboffenen Verbindung Ressourcen für den TCB reserviert bis der Empfänger nicht mehr in der Lage ist, weitere Anfragen anzunehmen. Ein ehrlicher Sender hat in einem solchen Fall keine Möglichkeit eine Verbindung mit dem Empfänger aufzunehmen.

SCTP kennt keine halboffenen Verbindungen, da der TCB bzw. die Informationen zum Anlegen des TCB über den Cookie-Mechanismus, wie er in Abschnitt 6.2.1 beschrieben ist, erst beim endgültigen Aufbau der Verbindung beim Empfänger angelegt wird. Somit führt ein SYN-Flood-Angriff auf SCTP nicht zum gewünschten Erfolg.

6.3 Zuverlässiger Datentransfer

Im Grundsatz ist SCTP für den *zuverlässigen* Versand von Daten konzipiert. Kurz zusammengefasst wird der zuverlässige Versand dadurch erreicht, dass der Empfänger eines Daten-Chunks diesen durch einen SACK-Chunk quittiert. Somit wird der Sender in die Lage versetzt zu erkennen, ob Daten möglicherweise auf dem Weg zum Empfänger verloren gegangen sind. Wenn er einen solchen Verlust feststellt, wird der noch fehlende Chunk erneut gesendet, bis sämtliche abgesendete Daten-Chunks korrekt quittiert wurden.

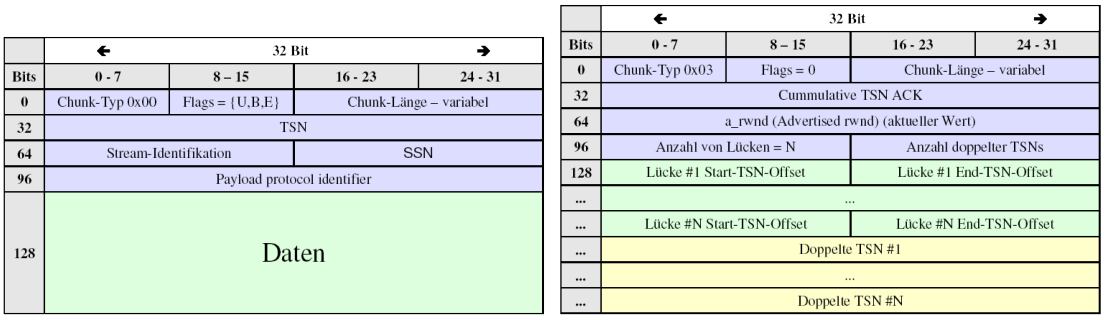
In diesem Abschnitt wird neben dem reinen Versand auch die Methodik der Neuübertragung behandelt, die auf die Performance des Datentransfers einen entscheidenden Einfluss hat. Werden die Daten sehr schnell nach dem Versand neu übertragen, kommt es zu einem erhöhten nicht zielgerichteten Traffic auf der Leitung. Zudem sind Einbußen im möglichen Sendevolumen hinzunehmen, da Neuübertragungen als Warnhinweis für die Staukontrolle (vgl. Abschnitt 6.4) gewertet werden. Werden die Daten erst nach einer zu groß gewählten Wartezeit neu übertragen, kann ggf. der Empfänger die Daten nicht zeitnah an die Anwendung weiterleiten, da die Reihenfolge der Chunks berücksichtigt werden muss.

6.3.1 Grundsätzlicher Ablauf des Datentransfers

Der grundsätzliche Ablauf des Datentransfers beschreibt den Übergang der von der Anwendung eingehenden Nutzdaten bis hin zum SCTP-Paket und der entsprechenden Rücktransformation beim Empfänger. Das Verständnis für den Ablauf ist für die praktische Realisierung von SCTP und die darauf aufbauenden Erweiterungen von unschätz-

barem Wert.

Bei der Realisierung des IN wird in diesen Prozess ergänzend eingegriffen. Im dritten Teil der Arbeit wird auf die Umsetzung von sicherheitsrelevanten Komponenten im Allgemeinen und mit Hilfe des IN im Speziellen eingegangen. Wird beispielsweise Secure-SCTP, wie es in Abschnitt 11.1 beschrieben ist, eingesetzt, muss bereits vor dem eigentlichen *Bundling* die kryptographische Behandlung der Daten erfolgen. Ähnlich verhält es sich bei der in Abschnitt 8 vorgestellten Architektur zur Multi-Pfad-Übertragung von Daten mittels SCTP, da hier direkt die Sendqueue des Senders abgegriffen wird.



(a) Der Daten-Chunk (b) Der SACK-Chunk

Abbildung 6.10: Chunks für den zuverlässigen Datentransfer

Die Struktur des Daten-Chunks sowie des SACK-Chunks sind in Abbildung 6.10 dargestellt. Die Bedeutung der einzelnen Parameter wird mit der Beschreibung des Prozesses nachgeliefert.

6.3.2 Die Datenübertragung aus Sender-Sicht

Die Sender-Sicht beschreibt den Weg von Nachrichten, die von der Anwendung zur Übertragung an SCTP weitergereicht werden, bis hin zur Einspeisung als IP-Paket in das Netz.

Von der Nachricht zum Daten-Chunk

Die Daten gehen in Form von User-Nachrichten beim SCTP-Kern ein. Im ersten Schritt müssen diese „Rohdaten“ in einzelne Datenchunks zerlegt werden, damit sie in ein SCTP-Paket, wie es in Abbildung 6.2 abgebildet ist, eingebettet werden können. Falls die angelieferte Nachricht so klein ist, dass sie in einem einzelnen Daten-Chunk Platz findet, so kann direkt ein einzelner Daten-Chunk generiert werden. Anders sieht es aus, wenn die Nachricht zu groß für einen einzelnen Chunk ist. In diesem Fall werden die Daten in einem vorverarbeitenden Schritt zunächst *fragmentiert*. Ein Daten-Chunk, der

die komplette Nachricht ohne Fragmentierung aufnehmen kann, wird im Folgenden als *ganzheitlicher* Daten-Chunk bezeichnet.

Fragmentierung von großen Nachrichten

Daten, die über ein IP-Netzwerk gesendet werden sollen, sollten eine bestimmte Größe nicht über- bzw. unterschreiten, da nur so ein effizienter Versand realisert werden kann. An diese Mechanismen ist SCTP gebunden, was wiederum die Notwendigkeit eines entsprechenden Fragmentierungs-Algorithmus notwendig macht. Die Frage nach der maximalen Größe, die ein SCTP-Paket erreichen kann, kann über die MTU, wie sie in Definition 6.4.4 festgelegt ist, beantwortet werden. Da SCTP über das Multi-Homing in der Lage ist, mehrere Pfade für den Transport zu verwenden, muss sich die maximale Paketgröße an dem Pfad orientieren, der die kleinste MTU aufweist, da im Vorfeld nicht feststeht, welcher Kanal wirklich für die Übertragung zur Verfügung steht. Insbesondere können Pakete bei einer Neuübertragung auf einem anderen Pfad übertragen werden. SCTP stellt einen Algorithmus bereit, der die maximale MTU der Pfade ermitteln kann (vgl. Abschnitt 6.4).

Ein Daten-Chunk enthält neben den Nutzdaten und den grundsätzlichen Chunk-Parametern wie Typ, Länge und Flags weitere Informationen. So wird die TSN gem. Definition 6.1.10 und ein von der Anwendung festgelegter Parameter, der sogenannte *Payload-Protokoll-Identifikator*, mitgegeben. Zusätzlich muss die Festlegung des zu verwendenden Streams nach Definition 6.1.5 erfolgen, d.h. die Parameter *Stream-Identifikator* nach Definition 6.1.6 und *Stream-Sequence-Nummer* nach Definition 6.1.7 sind zu belegen.

Bei der Fragmentierung werden einige Einstellungen im Daten-Chunk variiert, damit der Sender die Daten wieder korrekt zusammensetzen kann. Jeder fragmentierte Daten-Chunk erhält die gleiche SSN, sodass die einzelnen Chunks in ihrer Gesamtheit als eine Nachricht verstanden werden. Somit weiß der Empfänger bei Eingang eines entsprechenden Chunks, in welcher Reihenfolge die gesamte Nachricht zu behandeln ist und zu welcher Einheit der einzelne Chunk gehört. Anhand der TSN ist der Empfänger auch in der Lage, die Reihenfolge innerhalb einer Nachricht, sprich: aller Daten-Chunks einer bestimmten SSN, eindeutig zu rekonstruieren.

Allerdings kann der Empfänger allein aus diesen Informationen noch nicht herleiten, ob noch weitere Daten-Chunks für die Nachricht benötigt werden oder ob bereits alle Daten-Chunks und somit die gesamte Nachricht bei ihm eingegangen ist. Hierzu wird eine zusätzliche Information benötigt, die über die Chunk-Flags publiziert wird. Bei dem ersten Daten-Chunk einer fragmentierten Nachricht wird das *B-Flag* gesetzt, der damit als erstes Fragment ausgezeichnet ist. Äquivalent dazu wird das letzte Fragment der Nachricht mit der *E-Flag* gekennzeichnet. Die weiteren Fragmente benötigen keine weitere Kennzeichnung, sodass hier keine weiteren Flags berücksichtigt werden müssen. Jetzt ist der Empfänger in die Lage versetzt worden, die fragmentierten Daten, die auf einem

bestimmten Stream eingehen, korrekt zusammenzusetzen und geordnet auszuliefern.

Vom Chunk zum SCTP-Paket

Nachdem sämtliche Parameterfelder des Daten-Chunks gefüllt wurden, kann er in ein SCTP-Paket eingebunden – „gebundelt“ – werden. Da die Übertragung einer Kontrolle unterliegt und somit nicht alle von der Anwendung eingehenden Nachrichten zwingend sofort übertragen werden, werden die zu übertragenden Chunks in eine *Sendeschlange* (engl. sendqueue) eingereiht, die dann sukzessive abgearbeitet wird. Zusätzlich zu den Daten-Chunks generiert der SCTP-Kern möglicherweise auch Kontroll-Chunks, diese werden ebenfalls mit den notwendigen Parameterdaten versorgt, diese werden allerdings direkt an das *Bundling* weitergereicht, da das Zurückhalten von Steuerungsinformationen nicht zielführend ist.

Im ersten Schritt wird das Gerüst für das SCTP-Paket erstellt, d.h. der Kopfbereich wird mit den notwendigen Informationen – wie sie in Abschnitt 6.1 erläutert wurden – gefüllt. Danach werden so viele Daten-Chunks wie möglich in das neu generierte SCTP-Paket eingefügt. SCTP-Kontrollchunks werden bevorzugt behandelt und sind damit als Erste im Paket untergebracht, gefolgt von den Daten-Chunks, wie in Abbildung 6.2 dargestellt.

Es gibt einige wenige Ausnahmen, so gibt es Kontroll-Chunks, die nicht mit Daten-Chunks gemeinsam versendet werden dürfen. Diese werden in einem eigenen SCTP-Paket übertragen. Zudem besteht die Möglichkeit, SCTP so zu konfigurieren, dass auf das Bundling verzichtet wird. Damit kann unter Umständen ein schnellerer Versand der einzelnen Chunks erreicht werden. Auch wenn diese Einstellung aktiviert wurde, ist das Bundling nicht vollständig deaktiviert. Im Rahmen der Staukontrolle 6.4, insbesondere bei Neuübertragungen, werden mehrere Daten-Chunks aufgrund der ungünstigen Netzauslastung doch zusammen übertragen.

Vom SCTP-Paket zum IP-Paket

Als Nächstes wird das SCTP-Paket in die IP-Schicht entlassen. Hierfür muss das SCTP-Paket in ein IP-Paket eingebunden und die Kopfparameter des IP-Pakets mit plausiblen Werten versorgt werden.

6.3.3 Die Datenübertragung aus Empfänger-Sicht

Die Empfänger-Sicht beschreibt den Weg von eingehenden IP-Paketen über die Benachrichtigung des Senders, sprich: Quittierung der eingegangenen Daten bis zur Auslieferung als Nachricht an die Endanwendung.

Der Empfänger muss nicht nur das SCTP-Paket aus den Daten extrahieren und das Unbundling der Chunks durchführen, sondern auch auf die korrekte Reihenfolge der Nachrichten achten. Er darf nur vollständig übertragene Pakete in der richtigen Reihenfolge, heißt in der Reihenfolge, in der sie auch versendet wurden, an die Anwendung ausliefern.

Hier können mehrere Probleme aufkommen. Zum einen können einzelne Pakete doppelt eintreffen, zum anderen können Pakete fehlen, die die vollständige Auslieferung der bereits angekommenen Pakete verhindert.

In den Fällen, dass die eingegangenen Daten entsprechende Probleme aufweisen, muss der Empfänger nicht nur die bereits eingegangenen Chunks quittieren, sondern auch den Sender über die aufgetretenen Probleme informieren. Diese Aufgabe übernimmt im Normalfall der SACK-Chunk, wie er in Abbildung 6.10b grafisch dargestellt ist. Eine Ausnahme hiervon ist lediglich beim geordneten Shutdown möglich, wie er in Abschnitt 6.2.3 beschrieben ist, da hier u.a. der SHUTDOWN-Chunk zur Quittierung herangezogen wird. Wie und in welcher Form die Quittierung durchgeführt wird, ist dem Abschnitt 6.3.4 zu entnehmen.

Und zurück zum Daten-Chunk

Ein eingehendes SCTP-Paket wird umgehend dem *Unbundling* zugeführt und somit in die einzelnen Chunks, die in dem Paket transportiert wurden, zerlegt.

Kontroll-Chunks werden vorrangig bearbeitet. Sie werden direkt an den SCTP-Kern weitergereicht. Aus Sicht der Implementierung kann dies präziser ausgedrückt werden. Konkret werden die Kontroll-Chunks dem *SCTP-Controller* zur Verfügung gestellt, der die weiteren, notwendigen Schritte zur Bearbeitung der Kontroll-Chunks einleitet.

Die Daten-Chunks werden von ihren Kopfdaten befreit und wieder zu Nachrichten zusammengesetzt (engl. recovered). Die so reaktivierten Nachrichten werden äquivalent zum Senderverhalten in eine Schlange, der *Stream-recording-queue*, eingereiht, bevor sie an die Zielanwendung ausgeliefert werden. Zu beachten ist, dass die eingehenden Nachrichten in der richtigen Reihenfolge in die Schlange eingereiht werden müssen. Stellt man fest, dass Nachrichten fehlen, sodass bereits eingegangene Nachrichten nicht korrekt in die Schlange eingereiht werden können, so werden diese „verfrüht“ eingegangenen Nachrichten noch nicht an die Schlange abgegeben. Sie verbleiben im sogenannten Empfangsfenster, bis die fehlenden Daten eingetroffen sind und in die Schlange eingereiht wurden.

Definition 6.3.1 (rwnd) *Alle Daten, die noch nicht in die Stream-recording-queue ausgeliefert werden können, da noch nicht alle benötigten Daten-Chunks beim Empfänger eingegangen sind, werden in einem Speicherbereich, dem* Empfangsfenster*, abgelegt. Der Parameter* Receiver Window (rwnd) *gibt an, wie viele Daten in Bytes der Empfänger noch im Empfangsfenster zwischenspeichern kann. Der Sender seinerseits kann die Größe des Empfangsfensters aufgrund der Informationen, die über die Quittierungsmeldungen – SACK-Chunks – eingehen, berechnen.*

Analog zum *Empfangsfenster* werden in Definition 6.4.3 die *Sendefenster* eingeführt. Dabei ist zu beachten, dass das Empfangsfenster für die gesamte Assoziation gilt, da hier die Betrachtung der noch zur Verfügung stehenden Aufnahmekapazitäten beim

Empfänger im Mittelpunkt stehen. Die Sendefenster fokussieren auf die Aufnahmekapazitäten einzelner Pfade, sodass hier pro Pfad einer multihomed Verbindung ein Sendefenster-Parameter verwendet wird.

Die Behandlung von fragmentierten Nachrichten beim Empfänger läuft ähnlich ab wie die Bearbeitung von ganzheitlichen Daten-Chunks. Gehen Daten-Chunks ein, die als Fragment nur einen Teil einer Nachricht darstellen, werden diese an einen *Reassembler* weitergereicht, der die Nachricht erst dann in die Schlange einreiht, wenn die Nachricht vollständig aus allen Teilchunks zusammengesetzt wurde.

An der Definition 6.3.1 kann man bereits Schlüsse auf Stau- und Flußkontrolle, wie sie im Abschnitt 6.4 ausführlich beschrieben ist, ziehen. Da das Empfangsfenster nur eine begrenzte Größe aufweist, die über den SCTP-Parameter *rwnd* Sender- und Empfänger gleichermaßen zur Verfügung steht, kann der Zwischenspeicher nicht unbegrenzt mit Daten gefüllt werden. Wenn die Kapazität des Empfangsfensters erreicht ist, müssen demnach erst die fehlenden Nachrichten eingegangen sein, damit wieder Speicherbereiche freigegeben werden können. Dieses Vorgehen setzt ein kontrolliertes Senden der Nachrichten vom Sender voraus, der keine neuen Nachrichten ins Netz einspeisen darf, wenn der Empfänger diese gar nicht mehr aufnehmen kann.

Neue Nachrichten werden vom Sender somit in der Sendeschlange gehalten bzw. von der Quellanwendung nicht mehr entgegengenommen, bis ausreichend Kapazitäten im Netz zur Übertragung der Daten zur Verfügung stehen. Die Flusskontrolle ist somit sendergesteuert. Damit der Sender diese Steuerungsaufgaben wahrnehmen kann, muss er über die notwendigen Informationen verfügen, die er über die eingehenden SACK-Chunks erlangt.

6.3.4 Quittierungen unter Verwendung des SACK-Chunks

Die grundsätzliche Bedeutung des SACK-Chunks und des damit verbundenen Quittierungs-Verfahrens wurde bereits in Abschnitt 6.1 bei der einführenden Beschreibung von Chunks insbesondere der Bedeutung der Chunks zur Empfangsbestätigung, angesprochen. Die Funktion des SACK-Chunks, wie er in Abbildung 6.10b grafisch dargestellt ist, ist weit vielfältiger als das reine Bestätigen von eingegangenen Daten-Chunks.

Der SACK-Chunk unterrichtet den Sender über doppelte Daten-Chunks und indirekt über aufgetretene Lücken (engl. gaps) in der TSN-Reihenfolge, über fehlende bzw. noch nicht eingetroffene Chunks. Zudem kann der Sender über die Informationen des Parameter *a_rwnd* (vgl. Definition 6.2.2) im SACK-Chunk das Sendefenster des Empfängers, wie in Definition 6.3.1 beschrieben, berechnen. Somit erlaubt dieser Benachrichtigungsmechanismus dem Sender eine genaue Sicht auf die Situation beim Empfänger, sodass er seinerseits die Möglichkeit zur Kontrolle des Datentransfers erhält.

Das Prinzip der verspäteten Bestätigung

Die Verwendung von Kontrollchunks zur Bestätigung von eingegangenen Daten-Chunks führt zwangsläufig zu einem erhöhten Traffic, der zusätzlich zum Nutzlast-Traffic bewältigt werden muss. Bei der Spezifikation von SCTP wurde deshalb ein Mechanismus, der unter der Bezeichnung *verspätete Bestätigung* geführt wird, integriert, mit dem Ziel, den zusätzlichen administrativen Traffic zu minimieren.

Definition 6.3.2 (Verspätete Bestätigung) *Unter dem Prinzip der* Verspäteten Bestätigung *oder* Delayed Acknowledgement *versteht man die periodische Bestätigung von allen eingegangenen Daten-Chunks mit einer einzelnen SACK-Quittung.*

Müsste jeder Daten-Chunk einzeln bestätigt werden, würde dies zu einem erheblich höheren Datenvolumen für die Quittierung generiert werden, insbesondere, da die Möglichkeit besteht, viele kleine Daten-Chunks in einem Paket zu versenden. Über den *CumAck*, der in Definition 6.3.4 formal eingeführt wird, ist man in der Lage, mit einem einzelnen Parameter sämtliche bereits in der richtigen Reihenfolge eingegangenen Daten-Chunks zu bestätigen. Daten-Chunks, die außerhalb der Sortierreihenfolge eintreffen, werden ebenfalls durch die Angabe von TSN-Bereichsangaben, die in Beispiel 6.3.1 anschaulich erläutert sind, sehr effektiv zusammengefasst.

Diese Vorgehensweise wirft allerdings auch neue Probleme auf, da die entstehende Zeitverzögerung bei der Bestätigung vom Sender, insbesondere bei ausstehenden Neuübertragungen, berücksichtigt werden muss.

Definition 6.3.3 (Überflüssige Neuübertragung) *Unter einer* Überflüssigen Neuübertragung *bzw.* Unnecessary Retransmission *versteht man die Neuübertragung von Daten-Chunks, die aufgrund einer nicht korrekt berücksichtigten Verspätung der Quittung entsteht.*

Die überflüssige Neuübertragung hat zur Folge, dass aufgrund der nicht zeitnah eingegangenen Quittung fälschlicherweise angenommen wird, dass Daten-Chunks verloren gegangen sind und deshalb neu übertragen werden müssen. Ein solches Fehlverhalten führt zu einer erheblichen Störung der Datenübertragung, sodass der Sender bei der Bewertung der zur Verfügung stehenden SCTP-Parameter sehr sorgfältig vorgehen muss.

Neben der direkten Auswirkung auf den Netztraffic wirkt sich das Prinzip der *Verspäteten Bestätigung* auch auf die Ressourcen beim Sender und Empfänger aus, die für die Zwischenspeicherung von Informationen bereitgestellt werden. So gibt der Sender die Ressourcen, sprich: den reservierten Speicher für die Chunks frei, die bereits bestätigt wurden. Je größer die Verzögerung für die Bestätigung gewählt wird, desto mehr Daten-Chunks müssen beim Sender für eine eventuelle Neuübertragung vorgehalten werden.

Ähnlich verhält es sich beim Empfänger, dem nur ein begrenztes Empfangsfenster zur Verfügung steht, der Sender die Information über die Größe des Empfangsfensters aber

erst nach Eingang des SACK-Chunks berechnen kann, da die notwendigen Parameter für die Neuberechnung mit dem SACK-Chunk dem Sender übermittelt werden. Da das Empfangsfenster einen entscheidenden Einfluss auf die Leistungsfähigkeit von SCTP hat, wird die Berechnung im folgenden Teilabschnitt genauer beleuchtet.

Berechnung des Empfangsfensters des Empfängers beim Sender

Das Empfangsfenster *rwnd* des Empfängers kann vom Sender mit folgendem einfachem Algorithmus bestimmt werden. Bei der Beschreibung des Verbindungsaufbaus in Abschnitt 6.2.1 wurde bereits auf die Initialisierung eingegangen, da der initiale Wert des *rwnd*-Parmameters über den INIT-ACK-Chunk vom Empfänger vorgegeben wird. Somit wird als Startwert

$$rwnd := \mathit{INIT\text{-}ACK\text{-}Chunk} \longrightarrow a\_rwnd$$

gesetzt.

Immer wenn ein Daten-Chunk übertragen bzw. neu übertragen wird, korrigiert der Sender den *cwnd*-Parameter, indem er die Länge des Daten-Chunks vom ursprünglichen Wert abzieht. Somit ergibt sich

$$rwnd := rwnd - size(chunk\_data)$$

nach Übertragung des Daten-Chunks mit den Nutzdaten „chunk-data“.

Ein Daten-Chunk, der zur Neuübertagung ansteht, wird als solcher markiert und möglicherweise erst zu einem späteren Zeitpunkt tatsächlich gesendet. Da dieser Chunk bereits zum Versand bereitsteht, wird der *rwnd*-Parameter in diesem Fall angepasst. Eine Markierung eines bereits gesendeten Chunks zur Neuübertragung führt zur Vergrößerung des Empfangsfensters um die Größe der zu übertragenden Daten. Wenn die Daten dann tatsächlich in das Netz eingespeist werden, wird dieser Wert wieder vom *rwnd* abgezogen. Formal ergibt sich also der *cwnd*-Parameter zu

$$\begin{aligned} rwnd &:= rwnd + size(chunk\_data), && \text{falls Chunk markiert wird} \\ rwnd &:= rwnd - size(chunk\_data), && \text{falls Chunk gesendet wird.} \end{aligned}$$

Dieses Vorgehen resultiert aus der Tatsache, dass eine Übertragung von Daten nur möglich ist, wenn das Empfangsfenster zum einen eine minimale Größe besitzt, zum anderen wird das Empfangsfenster nur größer, wenn auch tatsächlich fehlende Daten eingehen und somit Daten an die Zielanwendung ausgeliefert werden können. Wenn nun auf der einen Seite das Empfangsfenster so klein wäre, dass keine Daten gesendet werden dürften und zum anderen die Daten tatsächlich verloren gegangen sind und somit neu übertragen werden müssten, würde sich ohne diesen „Trick“ eine Pattsituation ergeben, die nicht ohne Weiteres aufgelöst werden könnte.

Im Folgenden wird die Reaktion auf eingehende SACK-Chunks noch genauer erläutert, an dieser Stelle kann angenommen werden, dass der Sender in der Lage ist, aufgrund der

vorliegenden Informationen nach Eingang eines SACK-Chunks die Größe der *Outstanding-User-Daten* gem. Definition 6.2.12 bestimmen zu können. Nach Eingang eines SACK-Chunks wird der *rwnd*-Parameter aufgrund der Informationen aus dem SACK-Chunk neu gesetzt, und zwar als Differenz zwischen dem mitgelieferten *a_rwnd*-Prameter im SACK-Chunk und den bereits gesendeten und nicht quittierten Daten. Dies führt zu folgendem Schritt im Algorithmus:

$$rwnd := INIT\text{-}Chunk \longrightarrow a\_rwnd - outstandingBytes$$

Dadurch, dass der Wert des *rwnd*-Parameters durch die konkreten Werte im SACK-Chunk beim Sender abgeglichen werden können, hat er eine relativ realistische Sicht auf die Situation auf Empfängerseite.

Welche Daten sind korrekt beim Empfänger eingetroffen?

Im Abschnitt 6.1 wurde bei der Vorstellung der einzelnen Chunkklassen bereits auf einige Parameter des SACK-Chunks eingegangen, die an dieser Stelle konkretisiert werden. Aufbauend auf der Definition 6.1.10 kann der Parameter *Cumulative TSN acknowledgment* definiert werden.

Definition 6.3.4 (Cumulative TSN-Ack) *Der Parameter* Cumulative-TSN-Ack *enthält die TSN, bis zu der alle gesendeten Chunks vollständig beim Empfänger eingegangen sind. Alle Daten-Chunks, für die*

$$TSN <= Cumulative\text{-}TSN\text{-}ACK$$

gilt, sind vom Empfänger in die Stream-recording-queue *ausgeliefert worden. Im weiteren Verlauf wird die in der Literatur auch verwendete Abkürzung* Cum-Ack *verwendet.*

Anhand des *Cum-Ack* kann der Sender erkennen, welche Chunks er für eine mögliche Neuübertragung vorhalten muss, nämlich alle Daten-Chunks, deren TSN größer ist als der *Cum-Ack*. Für alle anderen Daten-Chunks kann der Sender die zugehörigen Ressourcen freigeben. Der Algorithmus für die Neuübertragung verloren gegangener Chunks wird in Abschnitt 6.3.5 dargestellt.

Der SACK-Chunk enthält weiterhin eine Liste der doppelt eingegangenen Chunks in Form der zugehörigen TSNs. Da anzunehmen ist, dass doppelte Cunks vereinzelt auftreten, werden die TSN-Werte zur Kennzeichnung herangezogen. Sind beim Empfänger Daten-Chunks eingegangen, die nicht an die Zielanwendung weitergegeben werden können, da noch Chunks mit kleinerer TSN fehlen, so bezeichnet man zusammenhängende, fehlende TSN-Blöcke als *Lücken* (engl. Gaps). Welche Chunks noch nicht beim Empfänger eingegangen sind, wird dem Sender indirekt über den SACK-Chunk mitgeteilt. Indirekt, da der SACK-Chunk als Quittierung fungiert und in dieser Funktion eingesetzt wird. Es teilt demnach zusätzlich zum Cum-Ack-Parameter dem Sender die Chunks mit, die

zwar korrekt, aber nicht in der richtigen Reihenfolge bei ihm eingegangen sind.

Da aufgrund der Algorithmen zur Staukontrolle versucht wird, das Limit des verwendeten Pfades auszunutzen, ist die Wahrscheinlichkeit hoch, dass es bei großen Datenaufkommen zu einer Vielzahl von Lücken kommt. Daher werden Lücken nicht in Form der TSN im SACK-Chunk angegeben, sondern nur als Versatz (engl. offset) zur Cum-Ack.

Ein 32-Bit-Feld reicht demnach aus, um einen Bereich durch ein 16-Bit Start- und Endeoffset zu beschreiben. In Abbildung 6.10b ist der Unterschied zwischen der 32-Bit-TSN-Angabe für doppelte Chunks und den 16-Bit-Offsets bei der Bereichsangabe deutlich zu erkennen. Ein Beispiel soll dies verdeutlichen:

Beispiel 6.3.1 (Gap-Ack-Block-Bereich) *Sei die Cum-Ack mit der* $TSN = 100$ *belegt, so sind die Chunks mit der* $TSN = 1 \ldots 100$ *korrekt beim Empfänger eingetroffen. Wenn ein Gap-Block-Ack-Bereich mit einem Startwert von* 30 *und einem Endewert von* 40 *angegeben wird, so ist der TSN-Bereich*

$$\begin{aligned} TSN_1 &= CumAck + StartOffset = 100 + 30 = 130 \qquad bis \\ TSN_2 &= CumAck + EndeOffset = 100 + 40 = 140 \end{aligned}$$

beim Empfänger korrekt eingegangen. Die fehlenden und ggf. neu zu übertragenen Chunks sind demnach die im TSN-Bereich 101 *bis* 129*. Auf diese Art und Weise kann der Sender den Zustand des Empfängers zum Zeitpunkt des SACK-Versands eindeutig nachbilden.*

6.3.5 Neuübertragung von verloren gegangenen Daten-Chunks

Da dem Sender jetzt alle Informationen über korrekt übertragene Daten vorliegen, kann er auf Fehler reagieren und fehlende Chunks neu übertragen. Das Ausbleiben von Chunks hat auch Auswirkungen auf die Staukontrolle und das Datenvolumen, welches vom Sender in das Netz einspeisen darf. Die Staukontrolle wird in Abschnitt 6.4 erörtert, in diesem Abschnitt soll die Frage geklärt werden, wann es konkret zu Neuübertragungen von Daten-Chunks kommt.

SCTP stellt zwei parallel zu verwendende Verfahren zur Neuübertragung zur Verfügung. Zum einen die timergesteuerte Neuübertragung, zum anderen einen Algorithmus, der sich *Fast-Retransmit* nennt und im deutschen Sprachraum als *schnelle Neuübertragung* bekannt ist. Im Falle von SCTP kann die schnelle Neuübertragung zum sogenannten Fast-Recovery eingesetzt werden, was wieder zum Abschnitt über Staukontrolle führt.

Timergesteuerte Neuübertragung

Auch bei timergesteuerter Neuübertragung kann zwischen zwei grundsätzlichen Strategien unterschieden werden. Es kann für jeden Daten-Chunk ein eigener Timer aufgebaut werden, der bei Ablauf ohne gültige Quittierung eine Neuübertragung des Chunks in

Auftrag gibt. Diese Variante wurde bei SCTP nicht gewählt.

SCTP sieht für jeden Pfad einen einzelnen Timer, den sogenannten *T3-rtx*-Timer vor, der bei Ablauf versucht, alle ausstehenden Daten-Chunks neu zu übertragen. Versuch, da es aufgrund der Staukontrolle, genauer gesagt aufgrund der Größe des Empfangsfensters, welches direkt das mögliche Sendevolumen steuert, ggf. nicht möglich ist, alle Daten auf einmal in das Netz einzuspeisen. Damit der Empfänger seinerseits in der Lage ist, Nutzdaten an die Zielapplikation auszuliefern, werden Neuübertragungen vorrangig behandelt. Wenn es zu sehr vielen Anforderungen zur Neuübertragung kommt, werden möglicherweise gar keine neuen Daten übertragen, sondern nur die ausstehenden Pakete abgearbeitet. Eine solche Situation muss durch die Flusskontrolle verhindert werden, da in einem solchen Fall die gesamte Übertragung zum Erliegen kommt.

Wann der T3-rtx-Timer abgelaufen ist, ist nicht fest vorgegeben, sondern wird über SCTP-Parameter, die aus der aktuellen Situation der Übertragung gewonnen werden, festgelegt. Da der Timer einen entscheidenden Einfluss auf die Leistungsfähigkeit der Übertagung hat, muss er sehr genau an die vorliegenden Gegebenheiten angepasst werden.

Die für das Fehlermanagement notwendigen Parameter sind in Abbildung 8.1 im grünen Aufgabenkreis unter der Überschrift *Zuverlässigkeit* zusammengestellt. Die Parameter werden im Abschnitt 8.1 im Zusammenhang mit der Pfadwahl durch das IN im Multi-Path-Szenario erneut eine wichtige Position zur Bewertung der gegebenen Auslastung der einzelnen Pfade geben.

Der Parameter, der anzeigt, wann der T3-rtx-Timer abgelaufen ist und damit die Neuübertragung anstößt, wird *Retransmission-Timeout* genannt.

Definition 6.3.5 (Retransmission-Timeout (RTO)) *Bei Eintreffen eines SACK- bzw. Heartbeat-Chunks wird der Parameter* Retransmission Timeout *oder kurz* RTO *neu berechnet, der festlegt, wann eine Neuübertragung von noch ausstehenden Daten initiiert wird.*

Um den Wert der RTO auf einen realistischen Wert zu setzen, wird eine Information darüber benötigt, wie lange ein Chunk bei der aktuellen Auslastung des Netzes unterwegs ist. Nur so kann der Sender entscheiden, ob der Chunk verloren gegangen ist oder sich nur aufgrund der geringen Kapazität des Übertragungspfades noch auf dem Weg zum Ziel befindet. Dabei ist auch die Verzögerung durch die in Definition 6.3.2 beschriebene Methode der verzögerten Bestätigung zu berücksichtigen. Dies führt zur Definition eines weiteren Parameters, nämlich der *Rundenlaufzeit*.

Definition 6.3.6 (Round-Trip-Time) *Die* Rundenlaufzeit *ist definiert als die Differenz zwischen der Zeit, zu der ein Daten-Chunk ins Netz eingespeist wurde, und der Zeit, wenn die Quittung über einen SACK-Chunk zu diesem Daten-Chunk eingegangen ist. Sie wird als* RTT *vom Englischen* Round-Trip-Time *abgekürzt.*

Die Rundenlaufzeit ist sehr variabel und zeitabhängig. Aufgrund der dynamischen Struktur des Netzes unterliegt der RTT-Wert einer ständigen Korrektur. Die Messung der RTT und die jeweilige Neuberechnung der RTO sind vollständig aus dem TCP-Protokoll übernommen. Auf die Berechnung wird im Zusammenhang mit der Parameterübersicht für das Intelligente-Netz kurz eingegangen. Das konkrete Vorgehen kann u.a. aus den beiden Standardtexten für TCP zu diesem Thema, nämlich [Allman et al. 1999] sowie [Jacobson 1988] entnommen werden, da die Berechnung identisch ist.

Die aktuelle RTO und damit der Zeitpunkt, der festlegt, wann ausstehende Pakete neu übertragen werden sollen, wird aus der Rundenlaufzeit (RTT) abgeleitet. Für die konkrete Bestimmung wird aber nicht direkt auf die ermittelten Werte zurückgegriffen, sondern auf Durchschnittswerte, um Messungenauigkeiten und Ausreißer besser kontrollieren zu können.

Definition 6.3.7 (SRTT) *Die* Smoothed-Round-Trip-Time (SRTT) *oder auch* geglättete Rundenlaufzeit *wird als Schätzwert für die durchschnittliche Rundenlaufzeit verwendet und mit Hilfe eines sogenannten Low-Pass-Filters aktualisiert. Die SRTT ergibt sich aus der gemessenen RTT R und einem Glättungsfaktor α zu*

$$SRTT_{neu} = (1 - \alpha) \cdot SRTT_{alt} + \alpha \cdot R,$$

wobei als Startwert die erste gemessene RTT R_1 zu

$$SRTT_{neu} = R_1$$

Verwendung findet.

Die Schätzwerte können als Zufallsvariable aufgefasst werden, die somit nicht als feststehende Größen aufgefasst werden können. Demzufolge stellt sich die Frage nach der Standardabweichung des Schätzwertes. Es wird ein Maß für die Streuung der SRTT-Werte um ihren Mittelwert gesucht. Eine zusätzliche Betrachtung der Abweichungen ermöglicht eine effektivere Reaktion auf Ausreißer und starke Schwankungen bei den gemessenen RTT-Werten. Die Berechnung der Standardabweichung ist nach [Stevens 2007] problematisch, da Wurzelberechnungen benötigt werden. Als Alternative kann nach [Stewart 2007] die geglättete Standardabweichung $RTTVAR$ in der folgenden Form angesetzt werden:

Definition 6.3.8 (RTTVAR) *Die geglättete Standardabweichung $RTTVAR$ ergibt sich aus der Messung der RTT R mit einem Glättungsfaktor β zu*

$$RTTVAR_{neu} = (1 - \beta) \cdot RTTVAR_{alt} + \beta \cdot |SRTT_{alt} - R_{neu}|,$$

wobei als Startwert

$$RTTVAR_1 = \frac{R_1}{2}$$

angenommen wird.

Basierend auf den Parametern $SRTT$ und $RTTVAR$ aktualisiert SCTP die RTO und legt damit den Zeitpunkt für mögliche Neuübertragungen aufgrund der Timersteuerung fest:

$$RTO = SRTT_{neu} + 4 \cdot RTTVAR_{neu}$$

Falls SCTP im Multi-Homing-Szenario eingesetzt wird, ist zu beachten, dass das Tripel

$$(SRTT, RTTVAR, RTO)$$

pro Pfad bestimmt wird.

Neuübertragung

Aufgrund der Vorüberlegungen kann der Algorithmus zur timergesteuerten Neuübertragung angegeben werden.

Mit den Schätzwerten SRTT und RTTVAR kann nach den Regeln des vorherigen Teilabschnitts die RTO eines Pfades bestimmt werden. Mit der RTO ist festgelegt, wann ein T3-rtx-Timer abgelaufen ist. Als Erstes wird der RTO-Wert des betroffenen Pfades bis zu einem vordefinierten RTO_{max} verdoppelt. Anschließend wird die Neuübertragung vorbereitet.

Wie bereits gesagt, kann es vorkommen, dass nicht alle fehlenden Daten-Chunks übertragen werden können, daher werden die entsprechenden Daten-Chunks erst als zur Neuübertragung vorgemerkt, markiert und in einem weiteren Schritt übertragen. Hierbei ist die Vorgehensweise zur Markierung bzw. zur Neuberechnung des Emfpangsfensters, wie sie in Abschnitt 6.3.4 beschrieben ist, zu beachten.

Ist die Multi-Homing-Option gesetzt, sodass mehrere Pfade zur Verfügung stehen, ist die Neuübertragung auf einem Sekundärpfad vorzunehmen. Die Multi-Homing-Problematik wird in Abschnitt 6.5.2 nachgereicht. Es werden immer die ältesten Daten zuerst nachübertragen, d.h. die Daten-Chunks mit den niedrigsten TSNs. Diese werden in ein SCTP-Paket „verpackt“ und gesendet. Alle markierten, aber noch nicht gesendeten Daten-Chunks bleiben markiert und werden bei den nächsten Durchläufen vorrangig abgearbeitet. Anschließend ist der T3-rx-Timer neu zu setzen, und der Vorgang ist abgeschlossen.

Schnelle Neuübertragung

Die timergesteuerte Neuübertragung ist dann von Nachteil, falls lediglich vereinzelt Daten-Chunks verloren gehen und ansonsten die Übertragung ohne Probleme und mit hohem Datendurchsatz durchgeführt werden kann. Dies hat mehrere Auswirkungen auf das Gesamtsystem.

Beim Empfänger können bereits eine große Menge an Daten-Chunks in korrekter Reihung eingegangen sein, die aber aufgrund eines einzelnen fehlenden Chunks nicht an die Zielanwendung ausgeliefert werden können. Somit blockiert ein einzelner Irrläufer die gesamte Auslieferung, da auf den Nachzügler gewartet werden muss.

Eine Neuübertragung hat immer auch Auswirkungen auf die Staukontrolle, die ihrerseits direkt die Datenmenge beeinflusst, die in das Netz eingespeist werden darf. Falls nur ein vereinzelter Daten-Chunk verloren gegangen ist, sollte dies bei der Staukontrolle berücksichtigt werden, d.h. eine weniger drastische Korrekturmaßnahme der Sendefenster bewirken. Um diesen Problemen zu begegnen, stellt SCTP neben der timergesteuerten Neuübertragung ein weiteres Konzept in Form der *schnellen Neuübertragung* zur Verfügung. Der Zusammenhang von *schneller Neuübertragung* und Staukontrolle wird in Abschnitt 6.4.4 hergestellt.

Der Schlüssel für die schnelle Neuübertragung (engl. Fast-Retransmission) ist der Gap-Ack-Block-Bereich des SACK-Chunks (vgl. Beispiel 6.3.1). Stellt der Sender Lücken fest, geht der Sender davon aus, dass die noch ausstehenden Daten-Chunks möglicherweise verloren gegangen sind. Jedem Daten-Chunk wird zusätzlich ein Zähler zugewiesen, der bei der Generierung des Daten-Chunks mit Null initialisiert wird. Stellt der Sender Lücken in der Übertragung fest, markiert er die *potenziell* verloren gegangenen Chunks, indem er diesen Zähler um eins erhöht.

Tritt ein fehlender Daten-Chunk innerhalb einer Lücke viermal hintereinander in einem SACK-Chunk auf, was über den zusätzlichen Zähler festgestellt werden kann, wird er unabhängig vom T3-rx-Timer für die Neuübertragung vorgesehen. In einem solchen Fall geht man davon aus, dass nicht die gesamte Übertragung, sondern nur ein einzelner Chunk betroffen ist. Die Markierung erfolgt analog zur timergesteuerten Neuübertragung, sodass auch bei der schnellen Neuübertragung der Parameter *flightsize*, der eine Aussage über die gesendeten, aber noch nicht quittierten Daten erlaubt, entsprechend angepasst werden muss.

In Abbildung 6.11 wurde das Vorgehen bei der schnellen Neuübertragung vereinfacht dargestellt. Da SCTP auf dem Prinzip der *verspäteten Bestätigung*, wie sie in Definition 6.3.2 dargestellt ist, beruht, wird nicht jeder eingehende Daten-Chunk direkt bestätigt, sondern es werden mit einem SACK-Chunk periodisch Bereiche von eingegangenen Daten-Chunks als korrekt eingegangen quittiert. Sieht man von diesen technischen Feinheiten ab, lässt sich anhand der Abbildung der Vorgang einfach begreifen. Beim Sender wurde der Zähler, in der Abbildung als FR-Zähler bezeichnet, für die Daten-Chunks aufgeführt, bei denen dieser Wert von der Initialisierung abweicht. Auf der Empfängerseite kann man die Informationen des SACK-Chunks ablesen, und zwar die Cum-Ack und die bereits eingegangenen weiteren Daten-Chunks in Form von Gap-Blöcken. Nachdem der FR-Zähler für den Daten-Chunk drei den Wert 4 erreicht hat, wird dieser für die Neuübertragung vorgesehen. Auf diese Art und Weise können mehrere Daten-Chunks gleichzeitig für die Neuübertragung vermerkt werden, da nicht nur ein einzelner Chunk, sondern Bereiche

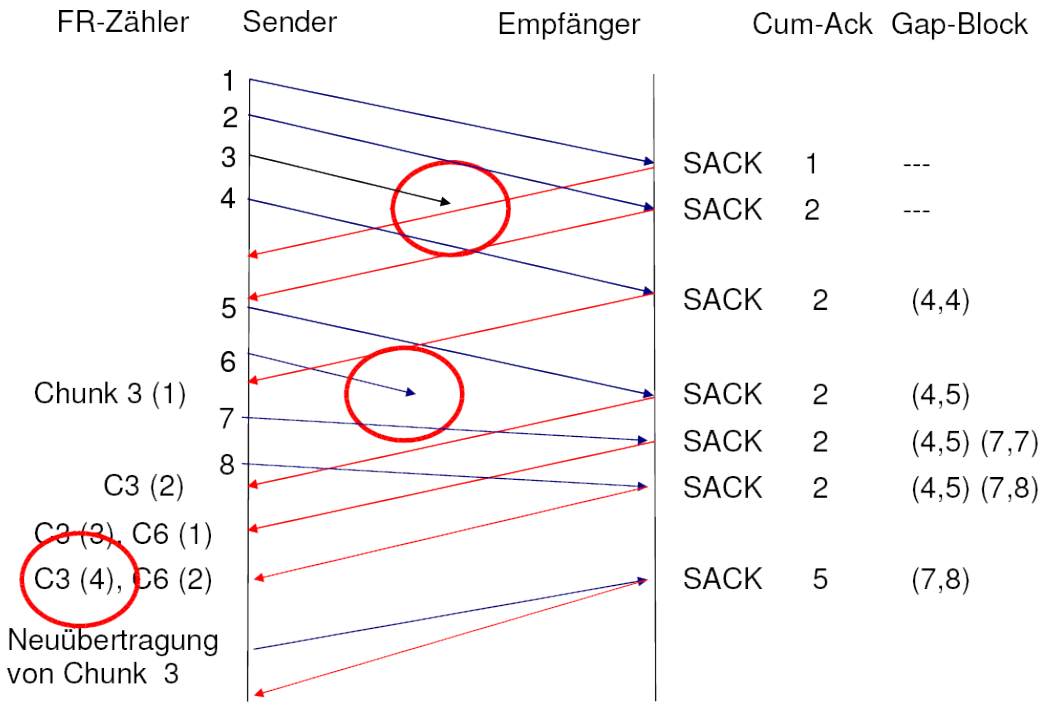


Abbildung 6.11: Stark vereinfachte beispielhafte Darstellung der schnellen Neuübertragung von SCTP

von Chunks bestätigt werden.

Wird SCTP unter Verwendung von Multi-Homing (vgl. auch Abschnitt 6.5.2) verwendet, sollte die Neuübertragung wenn möglich auf einem Sekundärpfad durchgeführt werden. Da nur der Primärpfad zur Übertragung der Nutzdaten verwendet wird und die Sekundärpfade als Ausweich-Pfade bereitgehalten werden, sollte so die Lücke „schnellstmöglich“ zu schließen sein.

Vergleich zu TCP

Das Grundprinzip der schnellen Neuübertragung wird bereits von TCP verwendet. Da TCP nur das zuletzt eingegangene Datensegment quittiert, weicht die Vorgehensweise von SCTP deutlich ab. Die Übertragung der Gap-Blöcke ermöglicht SCTP eine vorausschauende Sicht auf die Situation beim Empfänger.

Der Algorithmus zur schnellen Neuübertragung von TCP ist in Form eines einfachen Beispiels in Abbildung 6.12 dargestellt. Über den ACK wird dem Sender das höchste Da-

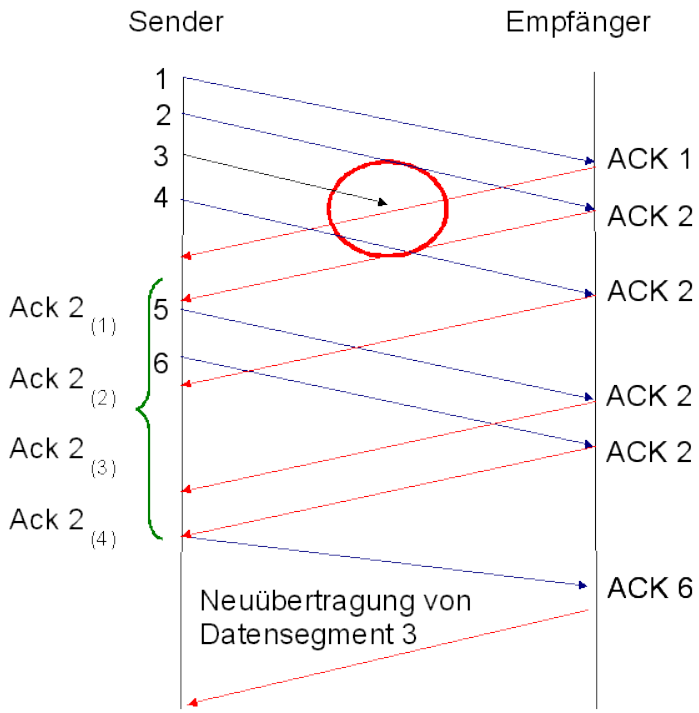


Abbildung 6.12: Schnelle Neuübertragung bei TCP – Es wird immer auf den Cum-Ack fokussiert.

tensegment angezeigt, welches in der richtigen Reihenfolge vollständig beim Empfänger eingegangen ist. Ob und inwieweit später versendete Segmente bereits beim Empfänger eingegangen sind und auf ihre Auslieferung warten, kann anhand dieser einfachen Form der Bestätigung vom Sender nicht in Erfahrung gebracht werden.

Wenn der Sender viermal in Folge dieselbe Bestätigung erhält, geht er davon aus, dass das Folgesegment nicht beim Empfänger eingegangen ist, erst jetzt erfolgt die Neuübertragung des fehlenden Segments. Greift man auf das Beispiel aus Abbildung 6.12 zurück, wird das Segment mit der Nummer 2 viermal bestätigt, sodass danach die Neuübertragung von Segment 3 angewiesen werden kann. Erst mit der folgenden Bestätigung erfährt der Sender, dass bis zum Segment mit der Nummer 6 keine weiteren Daten verloren gegangen sind.

Als Folge der schnellen Neuübertragung wird in TCP das schnelle Recovery als zusätzlicher Algorithmus angeboten. Da SCTP über die Gap-Blöcke dem Sender sämtliche relevanten Informationen zur Verfügung stellt, kann der Sender direkt auf den Algorithmus für die schnelle Neuübertragung zurückgreifen. Dies führt zum Abschnitt 6.4 über Staukontrolle.

6.4 Staukontrolle – Congestion Control

Ein modernes Transportprotokoll muss auch Mechanismen mitbringen, die eine faire Koexistenz der einzelnen Anwendungen im Netz ermöglichen. Wenn mehrere Anwendungen eine große Anzahl von Daten gleichzeitig unkontrolliert in das Netz einspeisen, kann man sich sehr gut vorstellen, das es passieren kann, dass das Netz aufgrund von Paket-Stauungen kollabiert und sämtliche Daten nicht mehr übertragen werden können. Diese Annahme hat sich in den späten Achtzigern nur zu unangenehm bestätigt, da es hier zu einigen Netzzusammenbrüchen aufgrund von unkontrollierter Dateneinspeisung gekommen ist (vgl. [Stewart and Xie 2001]).

Das Kollabieren des Netzes wird auch als *Congestion-Kollaps* bezeichnet, wobei dieser Begriff bereits 1984 im RFC 896 – [Nagle 1984] – geprägt wurde. Ein Kollaps liegt vor, wenn sehr viel Traffic im Netz festzustellen ist, wobei der sinnvolle Durchsatz extrem gering ist. Daraus ergibt sich die folgende Definition:

Definition 6.4.1 (Congestion-Kollaps) *Ein* Congestion-Kollaps *liegt vor, wenn viele Pakete im Netz eingespeist wurden, die aber fast alle nur aus neu übertragenen Paketen bestehen, die bereits beim Empfänger eingegangen sind, aber nicht korrekt quittiert wurden.*

6.4.1 Was versteht man unter Stauvermeidung?

Um einen solchen Stau zu vermeiden, sind Maßnahmen in der Transportschicht wie auch in der Netzwerkschicht denkbar. Da das IP-Protokoll keine Schutzmechanismen gegen die Bildung von *Staus* (engl. congestions) bereitstellt, ist es angebracht, dass die darüberliegende Transportschicht sich des Problems annimmt. So wurden bei der Definition von SCTP entsprechende Algorithmen zur *Staukontrolle* und *Stauvermeidung*(engl. congestion control and avoidance) vorgesehen, die sich sehr stark an den bereits für TCP entwickelten Mechanismen orientiert.

In den Anfängen von SCTP war eine solche Kontrolle nicht vorgesehen, aber nachdem die Gefahr erkannt wurde, wurde ein entsprechendes Vorgehen zur Staukontrolle erarbeitet. Die Grundideeen sind im RFC 2581 – [Allman et al. 1999] – zusammnengefasst, die sich auch in SCTP wiederfinden. Es ist aber auch heute noch nicht selbstverständlich, dass alle Transportprotokolle einen solchen Mechanismus anbieten, so wird der verbindungslose unzuverlässige Datentransfer über UDP auch heute noch ohne Algorithmen zur Staukontrolle bereitgestellt. Es bietet sich daher an, auch für den unzuverlässigen Datenversand auf SCTP, und zwar in der Form von PR-SCTP (vgl. Abschnitt 10.2), zurückzugreifen, da hier die Staukontrolle von SCTP im vollen Umfang zur Verfügung steht.

Um einen Stau zu vermeiden, werden wenn nötig weniger bzw. gar keine Datenpakete mehr in das Netz eingespeist. Einen Stau vermeiden heißt aber nicht gleich, dass der Datentransfer mit niedriger Datenrate durchgeführt werden muss, sondern dass die

Übertragung in Abhängigkeit der aktuellen Situation auf den zugeordneten Datenpfaden durchgeführt wird. Wenn eine Anwendung die Daten mit maximaler Datenrate übertragen kann, ohne Gefahr zu laufen, einen Stau zu produzieren, wird sie dies sicher auch machen. Es gilt also, Vorzeichen von Problemen auf den Datenpfaden zu erkennen und in Abhängigkeit der zu erwartenden Probleme zu reagieren.

Dies setzt allerdings voraus, dass das Protokoll während des Betriebs sich einen Eindruck von der aktuellen Leistungsfähigkeit der Datenpfade verschafft und auswertet. Bei SCTP werden hierfür sogenannte *Netzparameter* im laufenden Betrieb ermittelt, die zur Kontrolle des Datentransfers herangezogen werden können. Ein wichtiger Faktor dabei ist, dass die Informationen aus den bereits durchgeführten Übertragungen gezogen werden können. Bei SCTP spielt der SACK-Chunk eine entscheidende Rolle, da nach Eintreffen eines solchen Quittierungssatzes beim Sender eine Aussage zur Laufzeit des zugehörigen Datenchunks getroffen werden kann. Da im Normalfall eine größere Anzahl von SACKs in geringem Abstand zueinander eingehen, ist die Aussage auch repräsentativ für die aktuelle Leistungsfähigkeit des zur Übertragung verwendeten Kanals.

Da SCTP mehrere Pfade zur Verfügung stellt, wobei in der Standard-Variante nur ein Pfad zur Übertragung von Datenpaketen genutzt wird, ist es durch diese Methode nicht direkt möglich, Aussagen zu der aktuellen Situation der Sekundärpfade zu treffen. Hier kommen die bereits bei den Grundkonzepten von SCTP besprochenen HEARTBEAT-Chunks zum Einsatz. Um auch hier die Laufzeit eines Paketes zur Verfügung zu haben, wird ein Paket mit einem entsprechenden Heartbeat-Chunk anstelle eines Datenpakets gesendet. Bei Eintreffen einer zugehörigen Quittierung in Form eines HEARTBEAT-ACK-Chunks können äquivalente Werte zu Werten ermittelt werden, die aus den SACKs gewonnen wurden. Welche Werte ermittelt werden, wie dies konkret umgesetzt ist und wie sich daraus ein effizienter Algorithmus zur Staukontrolle ergibt, ist Thema dieses Abschnitts.

Einige Parameter begegnen uns im Abschnitt 8.1 wieder, da auch das IN Informationen zur aktuellen Situation im Netz, genauer die aktuelle Leistungsfähigkeit der einzelnen Pfade, benötigt. Es bietet sich demnach an, die bereits für die Staukontrolle verwendeten Parameter in das IN einfließen zu lassen. Im Abschnitt 8.2 wird bei der Auswahl und Bewertung der Parameter ein anderer Fokus gesetzt, da in diesem Fall entschieden werden soll, welcher Pfad am besten für den Versand geeignet ist, genauer ausgedrückt, welcher Pfad zum Zeitpunkt des Versands die besseren Pfadeigenschaften aufweist. Das IN verwendet kein eigenes Verfahren zur Staukontrolle, hier werden die erprobten und aus der Erfahrung heraus geeigneten, sehr guten Methoden des SCTP mit genutzt. Die Auswertung der Parameter zur Pfadwahl stellt somit einen Mehrwert und keinen Ersatz zum bestehenden Congestion Control dar.

6.4.2 Wie kann technisch eine Staukontrolle realisiert werden?

Aus [Allman et al. 1999] lassen sich die Grundkonzepte der Staukontrolle ableiten, die, wie bereits gesagt, in SCTP zur Anwendung kommen.

Liegen eine Reihe von Datenpaketen vor, die gesendet werden sollen, muss entschieden werden, wie viele Pakete tatsächlich ins Netz eingespeist werden. Dabei können grundsätzlich zwei verschiedene Ausgangssituationen unterschieden werden. Wenn SCTP sich mitten in einer Datenübertragung befindet und ständig Datenchunks gesendet und quittiert werden, kann SCTP die Netzsituation anhand der Netzparameter sehr genau abschätzen. Dieser Zustand wird im Folgenden als *Continue-Transfer* bezeichnet. Anders verhält es sich, wenn die Verbindung erst etabliert wurde oder längere Zeit inaktiv gewesen ist und keine bzw. noch keine Erfahrungswerte über die Pfadeigenschaften vorliegen. Diesen Zustand bezeichnen wir als *Start-Transfer*. Beide Situationen verlangen eine andere Strategie der Dateneinspeisung.

Im Falle des *Start-Transfers* besteht das Problem, dass noch keine Informationen über die Pfade vorliegen und somit auch keine Aussage zum vertretbaren Datentransfer gemacht werden kann. Hier bietet sich ein konservatives Vorgehen an, d.h. es wird mit der Übertragung einer kleinen Datenmenge begonnen, die sukzessive gesteigert wird. Daraus ergibt sich auch die Strategie für den Fall des *Continue-Transfers*. Die Datenrate wird erhöht, solange die Daten korrekt und vollständig beim Empfänger eingehen. Auf diese Weise kann der Sender sich langsam bis an die Grenze der Leistungsfähigkeit des Kanals heranarbeiten.

Die Erhöhung der Datenrate hat irgendwann ihr Maximum erreicht, sodass es zu Verlusten auf der Leitung kommt. Verlust heißt in diesem Fall die Zeit, die ein Paket benötigt, überschreitet eine Schwelle, die den Sender zu einer Neuübertragung von überfälligen Paketen zwingt. Jetzt besteht die Gefahr eines Congestion-Kollaps, daher muss die Übertragung sehr schnell reduziert werden. Die Neuübertragug von Paketen dient somit als *Warnsignal* für die Staukontrolle.

Wenn sich die Übertragung normalisiert hat, sprich: es zu keinen weiteren Neuübertragungen kommt, schließt sich der Kreis. Ab diesem Zeitpunkt kann wieder damit begonnen werden, die Datenrate sukzessive zu erhöhen. Mit diesem Algorithmus ist man in der Lage, eine hohe durchschnittliche Datenrate zu erzielen und gleichzeitig das Netz nicht zu überlasten. Für die Teilalgorithmen jeder Phase haben sich feste Begriffe etabliert:

Definition 6.4.2 (Algorithmen zur Staukontrolle) *Der Algorithmus, der den* Start-Transfer *regelt, wird als* Slow-Start-Algorithmus *bezeichnet. Der entsprechende Algorithmus für den* Continue-Transer *heißt* Congestion-Avoidance-Algorithmus.

Neben diesen Algorithmen mit direktem Bezug zur Staukontrolle wird ein weiterer Algorithmus benötigt, der unter besonderen Bedingungen in die Neuübertragung von Datenpaketen eingreift. Bei der *schnellen Neuübertragung* (engl. fast retransmit) wird da-

von ausgegangen, dass ein vereinzelt verloren gegangenes Paket nicht zwingend auf eine Stausituation hinweisen muss, daher wird hier als Reaktion die Reduktion des Traffics moderater gestaltet.

Die Algorithmen werden im folgenden Abschnitt informal beschrieben.

6.4.3 Steuerung der Staukontrolle über SCTP-Parmameter

Die von SCTP bereitgestellten Parameter werden im laufenden Betrieb fortwährend aktualisiert, sodass eine realistische Abschätzung der Netzauslaustung gegeben werden kann. Der Datenübertragung und die Parameter stehen in Wechselwirkung zueinander. Insbesondere wenn Parameterwerte für die Staukontrolle verwendet werden, reagiert die Datenübertragung von SCTP mit einem veränderten Sendeverhalten, worauf sich ihrerseits die Parameterwerte anpassen.

Das Zusammenspiel der Parameter und der Staukontrolle liegt im wissenschaftlichen Fokus, mit der Konsequenz, dass die Algorithmen zur Staukontrolle des öfteren dem aktuellen Kenntnisstand der Wissenschaft angepasst werden. Dabei werden weniger die im vorherigen Teilabschnitt besprochenen grundsätzlichen Methoden angepasst, sondern die Bewertung und Anpassung der Parameter wird variiert. In diesem Abschnitt wird für die Parametereinstellungen auf zwei Spezifikationen von SCTP Bezug genommen, und zwar auf den (zum Zeitpunkt der Erstellung dieses Dokuments) aktuellen RFC [Stewart 2007] und das Standardwerk über SCTP [Stewart and Xie 2001].

Für die parallele Übertragung von Daten im SCTP-Multihoming-Szenario werden die SCTP-Parameter zur Beurteilung der Kapazität der einzelnen Teilpfade herangezogen und sind daher weitgehend in Abschnitt 8.1 zusammengestellt. In Abbildung 8.1 sind die SCTP-Parameter grafisch gruppiert nach ihrer Verwendung dargestellt. An dieser Stelle sind die Parameter, die in der Grafik in dem gelben Aufgabenkreis *Flusskontrolle* abgebildeten Parameter von Interesse.

Zur Staukontrolle werden drei SCTP-Parameter, nämlich, das *cwnd*, *sshtesh* und die *flightsize* herangezogen. Da SCTP mehrere Pfade zulässt, die hinsichtlich ihrer Kapazität und Bandbreite variieren können, stehen die SCTP-Parameter zur Staukontrolle für jeden Pfad gesondert zur Verfügung. Das Sendefenster ist der wesentliche Begriff für die kontrollierte Übertragung und steht in Form der *cwnd*-Parameters zur Verfügung.

Definition 6.4.3 (cwnd) *Der Parameter* cwnd *beschreibt das* Congestion-Window *und entspricht einem Schätzwert, der angibt, wie viele Daten problemlos (unter Vermeidung einer Stausituation) in das Netz eingespeist werden können.*

Das *cwnd* legt demnach das Datenvolumen fest, das pro Zeiteinheit versendet wird. Wird dieser Wert erhöht, werden mehr Daten in das Netz eingespeist, wird dieser Wert sehr niedrig festgesetzt, werden kaum noch Daten neu versendet. Das *cwnd* steuert direkt

die physische Sendeleistung und wird von den Algorithmen zur Stauvermeidung an die tatsächlichen Kapazitäten im Netz angepasst.

Als Startwert wird das Sendefenster mit der zweifachen *MTU* vorbelegt, wobei normalerweise von einer MTU von 1500 Bytes für einen Standard-Ethernet-Knoten ausgegangen wird. Dies entspricht der älteren Spezifikation nach [Stewart and Xie 2001]. In [Stewart 2007] wird die folgende Belegung vorgeschlagen:

$$cwnd = \min(4 \cdot MTU,\ \max(2 \cdot MTU,\ 4380\ Bytes))$$

Der Wert *MTU*, der in die o.a. Berechnungen einfließt, ist folgendermaßen definiert:

Definition 6.4.4 (MTU) *Die* MTU *(engl. maximum transfer unit), auch maximale Übertragungseinheit genannt, ist ein von den Übertragungsmedien abhängiger Parameter, der die maximale Länge des zu versendenden IP-Pakets – d.h. ohne Fragmentierung – für einen speziellen Pfad angibt.*

Eine Verbindung zu einer IP-Adresse kann über unterschiedliche physikalische Leitungen erfolgen. Für die Berechnung der aktuellen MTU ist die kleinste dieser Leitungen ausschlaggebend. Bei Ethernet liegt die maximale Übertragungseinheit als Obergrenze bei maximal 15.000 Bytes. In SCTP ist ein Algorithmus implementiert, der die MTU für jeden Pfad automatisch bestimmt. Dieser entspricht weitgehend den äquivalenten Algorithmen von TCP. Zur Bestimmung werden nicht fragmentierbare Pakete in unterschiedlicher Länge versendet. Durch Auswertung der Rückantwort in Form von ICMP-Nachrichten des ersten Routers, der das Paket aufgrund seiner Größe nicht weitersenden kann, wird die MTU festgesetzt.

Damit ist der Startwert des Sendefensters relativ klein bemessen. Ob die Algorithmen das *cwnd* nach oben oder nach unten korrigieren, hängt von einem weiteren Parameter ab, und zwar vom *Slow-Start-Threshold*.

Definition 6.4.5 (sstresh) *Der Parameter* sstresh *gibt an, wann das Sendefenster vergrößert werden kann bzw. wann die Reduzierung des Traffics notwendig ist. Während das Sendefenster vergrößert wird, befindet sich die Anwendung im* Slow-Start-*Zustand (SSt), ansonsten im* Congestion-Avoidance-*Zustand (CA) . Da es sich um einen Schwellenwert zwischen den Zuständen handelt, wurde die Bezeichnung* Slow-Start-Threshold *gewählt.*

Der Schwellenwert kann zu Beginn der Übertragung mit einem sehr großen Wert initialisiert werden, da beide Algorithmen diesen Wert setzen und kontrollieren. Neben dem Sendefenster und dem Schwellenwert fließt die Information über bereits gesendete, aber noch nicht quittierte Daten in die Algorithmen zur Staukontrolle ein.

Definition 6.4.6 (flightsize) *Der Parameter* flightsize *gibt an, wie viele Daten an eine Adresse gesendet wurden, aber noch nicht mit einem SACK-Chunk bestätigt wurden.*

Die *flightsize* wird bei der Initialisierung auf Null gesetzt, da zu diesem Zeitpunkt noch keine Chunks ausgetauscht wurden.

Zusätzlich wird bei der Ausführung des Congestion-Avoidance-Algorithmus eine zusätzliche Variable verwendet, die zur Abschätzung des Sendefensters, genauer zur Berechnung seines Wachstums, herangezogen wird. Die als *Partial-bytes-acknowledged*(pba) benannte Variable wird häufig als vierter Parameter der Staukontrolle betrachtet.

6.4.4 Wie wird mit den SCTP-Parametern die Datenübertragung gesteuert?

Nachdem die notwendigen Parameter und deren initiale Werte eingeführt sind, kann der Prozess der Staukontrolle auf Basis der Parameter beschrieben werden.

Wenn noch keine Daten übertragen wurden bzw. die Pfade längere Zeit inaktiv waren, beginnt die Übertragung von Daten im *Slow-Start*-Zustand. Unter Verwendung der Parameter kann dies folgendermaßen ausgedrückt werden:

Satz 6.4.1 (Abgrenzung von Slow-Start und Congestion-Avoidance)

$$Zustand := \begin{cases} \text{„}Slow\text{-}Start\text{“} & , falls\ cwnd <= ssthresh \\ \text{„}Congestion\text{-}Avoidance\text{“} & , sonst \end{cases}$$

Geht man von einem Neustart aus, befindet man sich im Zustand *Slow-Start*, ansonsten wird der Status nach obiger Regel gesetzt. Im laufenden Betrieb befindet sich das System im Zustand *Congestion-Avoidance*, es sei denn, das System muss nach einem schweren Kollaps langsam wieder in den *Congestion-Avoidance*-Zustand zurückgeführt werden. Da hier dieselben Bedingungen gelten wie bei einem Neustart, wird das System in den *Slow-Start*-Zustand versetzt.

Jeder Pfad verwendet ein eigenes Sendefenster, im Folgenden wird der Begriff „das Sendefenster“ immer in Bezug auf den zugehörigen Pfad verstanden. Die Staukontrolle steuert den ausgehenden Datentransfer, indem das Sendefenster bei Bedarf vergrößert wird bzw. bei auftretenden Problemen in Form von Neuübertragungen verkleinert wird. Wird die Größe des Fensters nach unten korrigiert, erfolgt gleichzeitig eine Neuberechnung des zugehörigen *sstresh*-Parameters.

Vergrößerung des Sendefensters

Wird das Sendefenster vergrößert, können mehr Daten in das Netz eingespeist werden. Ein solches Vorgehen macht nur Sinn, wenn die aktuelle Pfadkapazität eine Mehraufnahme überhaupt verkraften kann. Die Bewertung der aktuellen Netzsituation erfolgt immer, nachdem neue Informationen zur Verfügung stehen, technisch gesehen, wenn ein

SACK-Chunk beim Sender eingeht. In diesem Zusammenhang wird auch die *flightsize* auf Basis der neu quittierten Daten-Chunks korrigiert.

Dies führt zu folgender Grundregel:

Satz 6.4.2 (Grundregel I zur Vergrößerung des Sendefensters) *Das Sendefenster kann nur bei einer fehlerfreien Übertragung vergrößert werden.*

Wobei eine fehlerfreie Übertragung auf zwei Bedingungen abstellt:

Definition 6.4.7 (fehlerfreie Übertragung) *Eine* fehlerfreie Übertragung *liegt vor,*

(1) wenn alle Daten-Chunks korrekt quittiert werden und es somit zu keinen Neuübertragungen von ausstehenden Daten-Chunks kommt,

(2) wenn sich der Cum-Ack kontinuierlich verschiebt.

Die Bedingung (2) in Definition 6.4.7 ermöglicht der Staukontrolle eine *vorausschauende* Kontrolle. Ein SACK-Chunk, bei dem sich der Cum-Ack nicht verschoben hat, quittiert keine neuen Daten-Chunks, die in korrekter Reihung beim Empfänger eingegangen sind. Dies deutet auf einen Bruch in der kontinuierlichen Übertragung der Daten hin und lässt vermuten, dass der Bedarf an Neuübertragungen kurz bevorsteht. In Hinblick auf diese Situation gilt, dass ein eingehender SACK-Chunk mit gleichbleibendem Cum-Ack eine weitere Vergrößerung des Sendefensters verhindert, auch wenn dies nach den folgenden Grundregeln theoretisch möglich wäre.

Das Sendefenster wird auch im Falle einer fehlerfreien Übertragung nur bei Bedarf erhöht. Solange noch *freier Platz im Sendefenster* vorhanden ist, besteht nicht die Notwendigkeit, das Sendefenster zu vergrößern.

Definition 6.4.8 (Freier Platz im Sendefenster) *Der Sender darf zusätzliche Daten in das Netz einspeisen, solange das* Sendefenster cwnd noch Platz bietet*. Genug Platz heißt, dass die Größe der gesendeten und nicht quittierten Daten* flightsize *kleiner ist als das Sendefenster* cwnd*:*

$$flightsize < cwnd$$

Eine Änderung beruht immer auf dem aktiven Eingreifen der Staukontrolle. Aus Definition 6.4.8 lässt sich eine Besonderheit ableiten. SCTP erlaubt, dass das Fenster „überlaufen“ kann, da bei einer $flightsize = cwnd - 1$ immer noch ein Paket gesendet werden kann. Ein solcher Überlauf ist in TCP nicht vorgesehen. Unter Verwendung von Definition 6.4.8 kann folgende Grundregel formuliert werden:

Satz 6.4.3 (Grundregel II zur Vergrößerung des Sendefensters) *Solange freier Platz im Sendefenster vorhanden ist, wird der zugehörige Parameter* cwnd *niemals erhöht.*

Mit den beiden Grundregeln zur Vergrößerung des Sendefensters stehen sämtliche Informationen für die Entscheidung zur Verfügung, ob ein Sendefenster *cwnd* vergrößert werden darf oder nicht. Der Zustand *Slow-Start* bzw. *Congestion-Avoidance* legt basierend auf der grundsätzlichen Entscheidung fest, um welchen Wert das Fenster vergrößert wird. Bevor auf die konkrete Erhöhung eingegangen wird, soll der Entscheidungsprozess kurz zusammengefasst werden.

Wann darf die Größe des Sendefensters nach oben korrigiert werden?

Nachdem ein SACK-Chunk beim Empfänger eingegangen ist, wird die aktuelle Netzsituation in Form der SCTP-Parameter neu bewertet. Um das Sendefenster vergrößern zu können, dürfen keine Neuübertragungen anstehen, und der Cum-Ack muss anzeigen, dass beim Empfänger kontinuierlich Daten an die Zielanwendung ausgeliefert werden. Wenn zusätzlich das Sendefenster vollständig ausgeschöpft wurde, kann davon ausgegangen werden, dass das Netz freie Kapazitäten aufweist und deshalb mehr Daten in das Netz eingespeist werden können. Daraus folgt, dass das Sendefenster vergrößert werden kann.

Um welchen Wert wird das Sendefenster vergrößert?

Um welchen Wert das Sendefenster vergrößert wird, ist abhängig vom Zustand, in dem sich die Staukontrolle befindet.

Im *Slow-Start*-Zustand wird die Menge an Daten (ack_data), die über den SACK-Chunk quittiert wurde, als Grundlage verwendet, wobei dieser Wert über die *MTU* gem. Definition 6.4.4 gedeckelt ist.

Satz 6.4.4 (Zunahme des Sendefensters im Slow-Start) *Befindet sich die Staukontrolle im Slow-Start-Zustand und ist eine Vergrößerung des Sendefensters nach Eingang eines SACK-Chunks möglich, so wird die Größe des Sendefenster wie folgt festgelegt:*

$$cwnd := \begin{cases} cwnd + ack\_data & , falls\ ack\_data < 1 \cdot MTU \\ cwnd + 1 \cdot MTU & , sonst \end{cases}$$

Die Zielsetzung im *Congestion-Avoidance* ist eine andere, hier wird versucht, einmal pro Rundenlaufzeit das Sendefenster um den Wert der MTU gem. Definition 6.4.4 zu erhöhen. Um dies zu erreichen, wird die Hilfsvariable *pba* (*Partial-Acknowledged-Byte*) verwendet, die in Abschnitt 6.4.3 bereits Erwähnung gefunden hat.

Bei jedem eingehenden SACK-Chunk wird die pba um die Größe der quittierten Daten (ack_data) erhöht. Der so ermittelte pba-Wert wird für die Zunahme des Sendefensters im *Congestion-Avoidance*-Zustand verwendet.

Satz 6.4.5 (Zunahme des Sendefensters im Congestion-Avoidance)
Befindet sich die Staukontrolle im Congestion-Avoidance-Zustand und ist eine Vergrößerung des Sendefensters nach Eingang eines SACK-Chunks möglich, so wird die Größe des Sendefenster wie folgt festgelegt:

$$cwnd := \begin{cases} cwnd + 1 \cdot MTU & , falls\ pba > cwnd \\ cwnd & , sonst \end{cases}$$

Wenn das Sendefenster vergrößert wird, muss auch der pba-Wert korrigiert werden, und zwar um genau den Wert, um den das Sendefenster vergrößert wurde, also $1 \cdot MTU$. Falls sämtliche vom Sender gesendeten Daten-Chunks als korrekt eingegangen quittiert sind, wird die *pba* auf Null zurückgesetzt.

Verringerung des Sendefensters

Wenn es zu Problemen bei der Übertragung kommt und ein Kollaps im Netz droht, wird das Sendefenster drastisch reduziert, um so dem Kollaps entgegenzuwirken. Gleichzeitig wird der Parameter *sstresh* angepasst, der festlegt, in welchem Zustand sich die Staukontrolle aktuell befindet.

Eine Verkleinerung des Sendefensters wird von zwei Ereignissen initiiert, die beide auf der Neuübertragung von Daten-Chunks zurückzuführen sind. Im Abschnitt 6.3.5 wurden zwei Situationen besprochen, die zu einer Neuübertragung von Daten führen, die gleichzeitig auch für die Reduzierung des Sendefensters verantwortlich sind. Dies ist der Ablauf des T3-rx-Timers, wenn größere Probleme auf dem Netz vermutet werden und die Kennzeichnung eines Daten-Chunks zur schnellen Neuübertragung, falls angenommen werden kann, dass nur wenige Daten aufgrund eines temporären Problems verloren gegangen sind.

Ein abgelaufener T3-rtx-Timer als Ursache für die Neuübertragung lässt auf größere Probleme im Netz schließen. In diesem Fall wird der Wert des *sshtresh*-Parameters neu bewertet und danach das Sendefenster verkleinert.

Satz 6.4.6 (Staukontrolle bei abgelaufenem T3-rtx-Timer) *Die Staukontrolle reagiert auf das Timer-Ereignis mit folgenden Einstellungen für den Schwellenwert*

$$sshtresh := max(\frac{cwnd}{2}, 2 \cdot MTU)$$

und setzt anschließend das Sendefenster auf

$$cwnd := 1 \cdot MTU.$$

Ruft man sich die Abgrenzung der Zustände bei der Staukontrolle aus Satz 6.4.1 in Erinnerung, so liegt der *Slow-Start*-Zustand vor, falls $cwnd <= ssthresh$ gilt. Damit wird die Staukontrolle für diesen Pfad zwangsläufig in den *Slow-Start*-Zustand zurückversetzt, da der *sshtresh*-Parameter nach der Neuberechnung mindestens doppelt so groß ist wie das Sendefenster. Somit werden sehr viel weniger Daten in das Netz eingespeist, und es besteht die Möglichkeit, dass es sich von dem Kollaps erholt. In diesem Fall würde der *Slow-Start*-Zustand dazu führen, dass sich die Größe des Sendefenster langsam wieder „normalisieren“ würde.

Anders verhält es sich, wenn die Staukontrolle aufgrund einer schnellen Neuübertragung zum Handeln gezwungen wird. In diesem Fall werden die Paramaeter wie folgt gesetzt:

Satz 6.4.7 (Staukontrolle bei schneller Neuübertragung) *Die Staukontrolle reagiert auf das Ereignis der schnellen Neuübertragung mit folgenden Einstellungen für den Schwellenwert*

$$sshtresh := max(\frac{cwnd}{2}, 2 \cdot MTU)$$

und setzt anschließend das Sendefenster auf

$$cwnd := sshtresh.$$

Man erkennt, dass lediglich die Einstellung des Sendefensters von den Einstellungen in Satz 6.4.6 abweicht. Zwar fällt auch hier die Staukontrolle des betreffenden Pfades in den *Slow-Start*-Zustand zurück, allerdings reicht eine einzige minimale Vergrößerung des Sendefensters aus, damit der *cwnd*-Parameter den Schwellenwert überwindet und sich somit die Staukontrolle wieder im *Congestion-Avoidance*-Zustand befindet. Es liegt demnach die bereits angesprochene moderate Variante der Sendefensterkorrektur vor.

6.5 Konzepte zur sicheren und performanten Datenübertragung

Die Konzepte des Multi-Streaming und Multi-Homings wurden bereits mehrfach in dieser Arbeit angesprochen. In diesem Abschnitt werden die einzelnen Methoden und Konzepte zusammengefasst und aus technischer Sicht beleuchtet.

6.5.1 Multi-Streaming

Auf das Streaming-Konzept wurde bereits mehrfach eingegangen. Aus fachlicher Sicht wurde in Abschnitt 3.3 ein Konzept zur performanten Übertragung von gemischten Bild- und Tondaten im medizinischen Umfeld entworfen. In diesem Abschnitt wird das Konzept aus technischer Sicht unter Berücksichtigung der Besonderheiten von SCTP betrachtet.

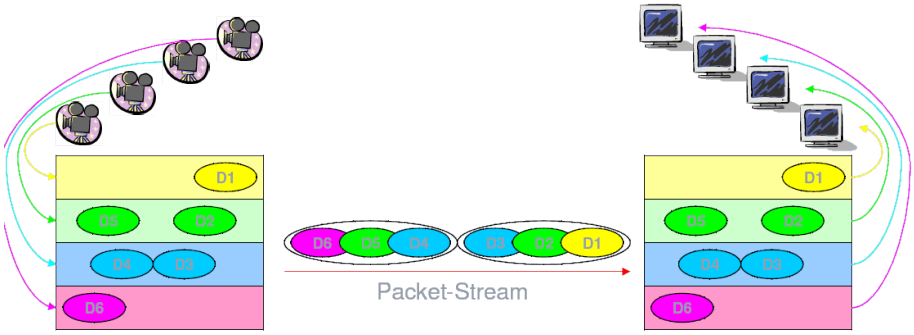


Abbildung 6.13: Logische Sicht auf die Übertragung von Daten über mehrere Streams

Die grundlegenden Begriffe wurden im Abschnitt 6.1 im Rahmen der Einführung in die grundlegenden Begrifflichkeiten von SCTP bereits definiert. So wurde der Stream in Definition 6.1.5 als unidirektionaler, logischer Datenkanal beschrieben. Das besondere am Konzept des Multi-Streaming ist die Tatsache, dass alle verwendeten Streams unter dem Dach einer Assoziation ablaufen und nicht für jeden Stream eine eigene Verbindung initiiert wird.

Da die Parameter für die Staukontrolle und das Fehlermanagement auf Basis der Assoziation fungieren, unterliegen sämtlicihe Streams einer Assoziation einer einzigen gemeinsamen Administration. Der Vorteil besteht darin, dass der Verwaltungsoverhead auf ein Minumum beschränkt werden kann.

Für eine Assoziation muss mindestens ein Stream vorgegeben sein, da die Übertragung von Daten intern über Streams abgewickelt wird. Werden beim Aufbau der Verbindung keine zusätzlichen Streams vereinbart, wird automatisch der Stream mit der $SID = 0$ für die Übertragung verwendet.

Über die *SID*, wie sie in Definition 6.1.6 eingeführt wurde, kann jeder Stream gezielt angesprochen werden. Die Anwendung kann gezielt Daten über einen bestimmten Stream versenden, sodass die Daten nach logischen Gesichtspunkten auf die einzelnen Streams verteilt werden können. In Abbildung 6.13 ist die logische Sicht auf das Konzept des Multi-Streamings schematisch dargestellt.

Man erkennt, dass jedem Stream ein logischer Buffer in Form einer eigenen Sende- bzw. Empfangsschlange zugewiesen wird. Unter diesen Bedingungen erfolgt die Datenübertragung eines Streams unabhängig von der Übertragung auf den anderen Streams. Die logische Trennung der Übertragung ist insbesondere dann von Vorteil, wenn ein geordneter Versand durchgeführt werden soll.

Jeder Daten-Chunk, der über einen bestimmten Stream versendet wird, erhält zusätz-

lich zur *TSN* einen *SSN*-Wert, der über die Definition 6.1.7 eingeführt wurde. Die SSN übernimmt im Rahmen des geordneten Versands dieselbe Aufgabe für den Stream wie die TSN für die gesamte Assoziation. Da die SSN eindeutig aufsteigend vergeben wird, kann der Empfänger anhand der SSN die Reihenfolge der eingehenden Daten-Chunks innerhalb eines Streams eindeutig rekonstruieren.

Ein Argument für die Einführung des Stream-Konzepts bei SCTP beruht sicher auf der Tatsache, dass es bei der klassischen TCP-Verbindung bei der Übertragung von Daten unter Verwendung von HTTP für die Darstellung von Webseiten zu Problemen kommen kann, wenn verschiedene Dateien im Rahmen einer TCP-Verbindung übertragen werden sollen.

Die Darstellung auf einer Webseite setzt sich im Normalfall aus einer Vielzahl von kleinen unabhängigen Objekten zusammen. Insbesondere werden viele kleine Grafiken, zumeist im gif- oder png-Format, verwendet, die einzig zur Anreicherung des Layouts aus ästhetischen Gesichtspunkten dienen. Werden alle Daten über eine einzige TCP-Verbindung ausgetauscht, kann es vorkommen, dass die Auslieferung bereits vollständig übertragener Dateien verhindert wird, da auf ein einzelnes Datensegment einer anderen Teildatei gewartet wird. Dieses Problem wird auch als *Head-of-Line-Blocking* bezeichnet.

Als Alternative könnte man zumindest theoretisch für jedes darzustellende Objekt eine eigene TCP-Verbindung aufmachen. Ein solches Vorgehen bindet aber sehr viele Ressourcen, da mit jeder zusätzlichen TCP-Verbindung auch der komplette Verwaltungsbereich von TCP für jede Verbindung bereitgehalten werden muss. Eine ressourcenschonende Übertragung unter Verwendung einer einzelnen TCP-Verbindung führt aber zu den o.a. Problemen.

Das Head-of-Line-Blocking ist in Abbildung 6.14 veranschaulicht. Obwohl die Datei II, die sich aus den Datenfragmenten $\{4, 5, 6\}$ zusammensetzt, bereits vollständig übertragen wurde, kann sie nicht an die Zielanwendung ausgeliefert werden, da auf das Segment Nr. 3 gewartet wird. Erst wenn das fehlende Segment möglicherweise durch schnelle Neuübertragung beim Empfänger eingeht, erkennt der Empfänger die Vollständigkeit und liefert die Daten aus.

Dieses Problem kann durch Verwendung des Streaming-Konzepts von SCTP vermieden werden. Da jeder einzelne Stream völlig autark agiert, kann es zwar vorkommen, dass eine Datei auf einem bestimmten Stream aufgrund von Problemen auf der Leitung nicht ausgeliefert werden kann, dies hat aber keine Auswirkungen auf die Auslieferung der anderen Dateien.

Die Auslieferung soll anhand der Grafik in Abbildung 6.13 erläutert werden. Beispielsweise würde ein Verlust des Daten-Chunks D1, der auf dem gelb markierten Stream transportiert wird, keinen Einfluss auf die Auslieferung der anderen Daten-Chunks haben. Die Daten, die auf dem grünen, roten und blauen Stream transportiert werden,

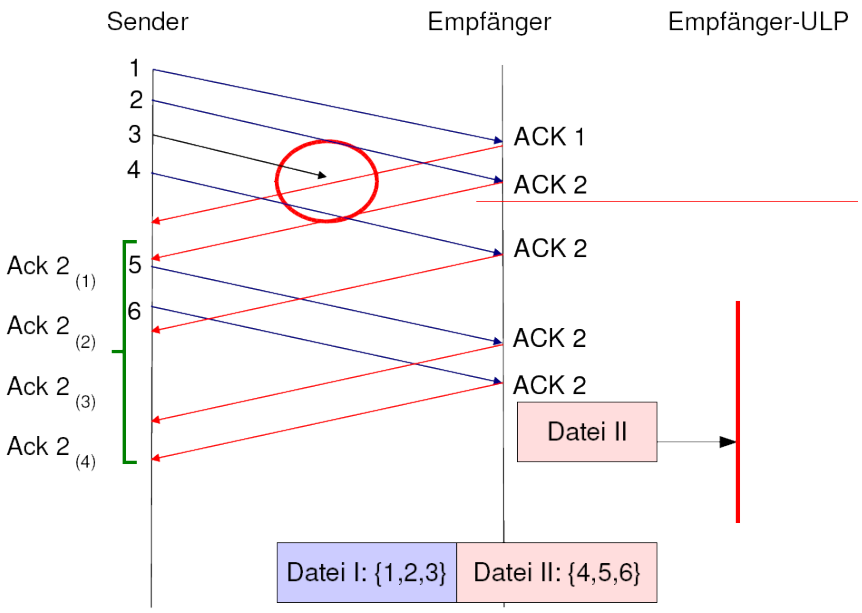


Abbildung 6.14: Head-of-Line-Blocking – Obwohl die Datei II vollständig übertragen wurde, kann sie nicht an die Zielapplikation ausgeliefert werden.

können somit ohne Wartezeit an die Zielanwendung ausgeliefert werden.

Auf eine weitere Besonderheit des Streamings wird in Teil III dieser Arbeit eingegangen, da man bei der Erweiterung von SCTP durch Secure-SCTP in der Lage ist, nicht nur die Ordnung eines einzelnen Streams festzulegen, sondern auch die kryptographische Behandlung vom verwendeten Stream in Abhängigkeit zum Stream festgelegt werden kann.

6.5.2 Multi-Homing

Im einleitenden Abschnitt 5.1 wurde das Konzept des Multi-Homings, also die Möglichkeit, einem Endpunkt mehrere Adressen zuzuordnen, bereits durch Angabe der Definition 5.1.1 eingeführt. Allerdings wurde der Schwerpunkt auf die Verwendung der zusätzlichen Pfade zum parallelen Datentransfer gelegt. In diesem Teilabschnitt wird das Multi-Homing aus Sicht von SCTP betrachtet, das im Grundsatz den Multi-Pfad-Transfer von Daten nicht vorsieht.

In Abbildung 6.15 ist die grundsätzliche Konstellation dargestellt. Die Endpunkte x und y einer einzigen Assoziation können über mehrere Pfade, im Beispiel über (IPx_1, IPy_1) und (IPx_2, IPy_2), erreicht werden. In der Spezifikation von SCTP werden die Pfade

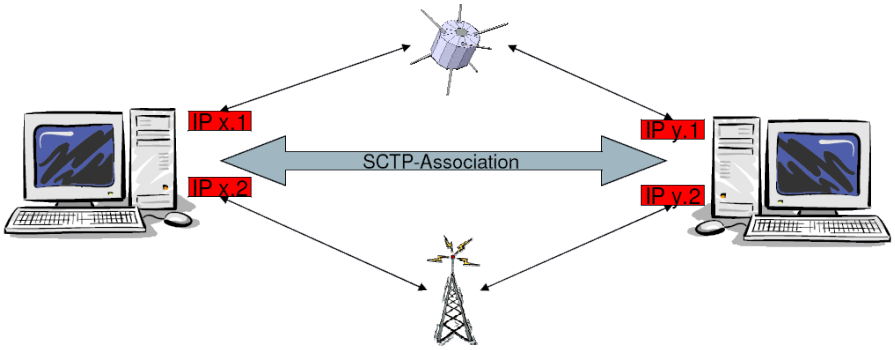


Abbildung 6.15: Die Multi-Homing-Option von SCTP

nicht als gleichwertig betrachtet. Ein Pfad wird herausgehoben und grundsätzlich für den Datentransport verwendet.

Definition 6.5.1 (Primärpfad – Primäradresse) *Wenn einem Endpunkt mehrere Adressen zugewiesen werden, wählt dieser eine Adresse als* primäre Zieladresse *aus, an die sämtliche Daten-Chunks gesendet werden. Der zugehörige Pfad wird als* Primärpfad *bezeichnet.*

Sämtliche anderen, nicht besonders ausgezeichneten Pfade werden als *Sekundärpfade* bezeichnet.

Definition 6.5.2 (Sekundärpfade – Sekundäradressen) *Alle Adressen eines Endpunkts einer multihomed SCTP-Assoziation werden als* sekundäre Zieladressen *bezeichnet. Die zugehörigen Pfade heißen* Sekundärpfade.

Multi-Homing – Erreichbarkeit

Ein Ziel des Multi-Homings besteht darin, auf einen Ausfall des Primärpfades reagieren zu können. In einem solchen Fall werden die Daten über die Sekundärpfade übertragen. Ein Ausweichen auf einen Sekundärpfad ist immer nur temporär möglich, d.h. wenn SCTP feststellt, dass der Primärpfad wieder aktiv ist, wird auf den ursprünglichen Primärpfad zurückgeschaltet.

Die Anwendung, die sich dem SCTP-Dienst zunutze macht, kann eingreifen und einen neuen Primärpfad festlegen. Dies ist die einzige Möglichkeit, regulär einen Sekundärpfad „dauerhaft“ als Primärpfad zu etablieren.

Es stellt sich die Frage nach der *Erreichbarkeit* der Sekundärpfade. Die Erreichbarkeit von aktiven Pfaden kann einfach ermittelt werden, da aufgrund des Datentransfers und der eingehenden SACK-Quittungen die Rundenlaufzeiten sowie die Sendefenster

des Senders und das Empfangsfenster des Empfängers bestimmt werden können. Im Abschnitt 6.4 wurden die zugehörigen Mechanismen und Berechnungen ausführlich dargelegt. Für alle inaktiven Pfade kann dieses Verfahren nicht angewendet werden, da hier aktuell keine Daten übertragen werden.

Um die Erreichbarkeit der inaktiven Pfade zu testen, werden regelmäßig Heartbeat-Chunks auf diesen Pfaden ausgetauscht, die es wiederum erlauben, eine Rundenlaufzeit zu bestimmen und damit die Möglichkeit bieten festzustellen, ob ein Pfad erreichbar ist. Dieses Konzept wurde im Abschnitt 6.1 im Zusammenhang mit der Einführung der Heartbeat-Chunks erläutert. Somit hat ein Sender alle Informationen zur Verfügung, die er benötigt. Muss ein Primärpfad aufgrund einer Störung temporär gewechselt werden, so kann – falls vorhanden – aufgrund der Heartbeat-Prüfungen auf einen erreichbaren Pfad gewechselt werden.

Ein wichtiger Parameter, insbesondere wenn Echtzeitanwendungen unter Verwendung von SCTP realisiert werden sollen, ist die *Umschaltzeit* t_{fail}.

Definition 6.5.3 (Umschaltzeit) *Als* Umschaltzeit t_{fail} *wird die Zeit betrachtet, die notwendig ist, bis der Sender den Ausfall des Primärpfads als solchen erkennt und auf einen Sekundärpfad umschaltet.*

Wie kann die Umschaltzeit abgeschätzt werden?

Die notwendigen Parameter sind im Abschnitt 6.3.5 über Neuübertragung eingeführt worden. Dies ist zum einen die RTO, die festlegt, wann es zu einer Neuübertragung kommt bzw. wann der entsprechende Timer abläuft. Die RTO kann sich nur innerhalb eines festgelegten Intervalls bewegen, d.h.

$$RTO_{min} \leq RTO \leq RTO_{max}.$$

Eine direkte Festlegung der Umschaltzeit ist nicht möglich, da RTO variiert und sich zusätzlich bei jedem Ablauf eines Timers bis zum Maximalwert verdoppelt. Zusätzlich wird folgende Konstante von SCTP für die Berechnung herangezogen:

Definition 6.5.4 (Path-Retransmission-Limit) *Unter dem* Path-Retransmission-Limit (PRL) *versteht man eine Konstante, die festlegt, wann davon auszugehen ist, dass der Primärpfad inaktiv ist. Die PRL gibt die Anzahl der hintereinander auftretenden Fehler an, die ausreichen, um den Pfad als inaktiv zu erkennen. In der Literatur wird ein Wert von* $PRL = 5$ *empfohlen.*

Die beschriebenen Prameter sind geeignet, die Umschaltzeit t_{fail} zu bestimmen.

Satz 6.5.1 (Abschätzung der Umschaltzeit) *Sei* PRL *das Path-Retransmission-Limit und die Werte* RTO_{min} *und* RTO_{max} *gegeben, so kann die Umschaltzeit* t_{fail} *als*

$$\sum_{i=0}^{RPL-1} 2^i \cdot RTO_{min} \;\leq\; t_{fail} \;\leq\; PRL \cdot RTO_{max}$$

abgeschätzt werden.

Die Berechnung wird an einem Beispiel verdeutlicht:

Beispiel 6.5.1 *Setzt man RTO_{min} bei einer Sekunde und lässt man den RTO-Wert bis zu $RTO_{max} = 60sec$ steigen, dann ergibt sich eine Umsschaltzeit von*

$$\sum_{i=0}^{RPL-1} 2^i \cdot RTO_{min} \leq \quad t_{fail} \quad \leq PRL \cdot RTO_{max} \tag{6.1}$$

$$(1+2+4+8+16) \cdot 1 \leq \quad t_{fail} \quad \leq 5 \cdot 60 \tag{6.2}$$

$$31 \leq \quad t_{fail} \quad \leq 300 \tag{6.3}$$

wenn man 5 aufeinanderfolgende Fehler zur Erkennung der Inaktivität des Pfades als ausreichend betrachtet. Es werden also bis zu 300 Sekunden benötigt, um festzustellen, dass ein Pfad gar nicht mehr erreichbar ist.

Die Abschätzung aus Beispiel 6.5.1 verdeutlicht die Problematik beim Einsatz in Echtzeitszenarien. Insbesondere in medizinischen Echtzeitszenarien, wie das in Abschnitt 3.1 beschriebene erweiterte Notfallszenario, reichen diese Umschaltzeiten nicht aus, da es bei der Versorgung von Notfallpatienten auf jede Sekunde ankommt. Um eine zeitnahe Versorgung zur garantieren, muss sichergestellt sein, dass ein Ausweichpfad möglichst schnell zur Verfügung steht, damit die Übertragung der wichtigen medizinischen Daten ohne Verzögerung unverzüglich fortgesetzt werden kann.

Im Folgenden wird eine Methodik auf Basis des IN entwickelt, die den Versand der Daten über mehrere Pfade erlaubt. Da die Pfadwahl immer den aktuell leistungsstärksten Pfad auswählt, wird ein inaktiver Pfad automatisch aussortiert. Daten-Chunks, die sich auf diesem Pfad befinden und nicht mehr beim Empfänger eingehen werden, werden durch die schnelle Neuübertragung erkannt und auf einem gararantiert aktiven Pfad neu übertragen. Die Verzögerung, die sich daraus ergibt, ist demnach minimal. Die Umschaltzeit ist damit nicht existent.

Multi-Homing – Neuübertragung

Neben der Erhöhung der Erreichbarkeit fällt den Sekundärpfaden eine weitere Aufgabe im Zusammenhang mit Neuübertragungen von verloren gegangenen Daten-Chunks zu. Das Thema Neuübertragung wird in Abschnitt 6.3.5 besprochen. Die Sekundärpfade werden wenn möglich für die Neuübertragung von Daten-Chunks verwendet.

Auf diese Art und Weise versucht SCTP, aufgetretene Lücken im Ablauf möglichst schnell zu schließen, damit der Empfänger die eingegangenen Daten an die Zielanwendung ausliefern und die reservierten Ressourcen freigeben kann. Solange diese belegt sind, können keine bzw. nur sehr wenig neue Daten in das Netz eingespeist werden (vgl. Abschnitt 6.4).

Wenn mehrere Sekundärpfade zur Verfügung stehen, wird versucht, alle Pfade für Neuübertragungen einzusetzen und gleichmäßig auszulasten. In [Stewart and Xie 2001] wird ein einfaches *Round-Robin*-Verfahren vorgeschlagen, um zwischen den aktiven Sekundärpfaden zu rotieren.

7 Klassifikationsverfahren

7.1 Kurze Beschreibung des Problems

Um herauszufinden, ob ein gegebenes Problem überhaupt mittels Data Mining gelöst werden bzw. ob ein bereits etabliertes Verfahren für dieses Problem zur Anwendung kommen kann, muss das Problem zumindest sprachlich abgefasst sein. Aus dieser informellen Beschreibung lassen sich bereits die möglichen Verfahren herleiten.

Allgemein kann das Problem wie folgt beschrieben werden:

Problembeschreibung 1 *Bei der Datenübertragung über mehrere Kanäle soll für jedes Paket der optimale Pfad – der Pfad mit der maximalen Dienstgüte – für die Übertragung genutzt werden.*

Die „aktuelle“ Situation im Netz ist aufgrund von physikalischen Gegebenheiten und Auslastung der Pfade ständigen Schwankungen unterzogen. Somit ändert sich das zu lösende Problem zeit- und umweltabhängig. Um einen lernfähigen Verteilalgorithmus zu entwickeln, muss die Netzauslastung überwacht und dem Lernalgorithmus in Form von Netzparametern zur Verfügung gestellt werden. Da dieser Verteilalgorithmus direkt Einfluss auf den Datentransfer nimmt, ist er in die Transportschicht eines Netzprotokolls zu integrieren. Nur hier stehen die notwendigen Informationen für den Datenversand zeitnah zur Verfügung. Diese „intelligente“ Komponente wollen wir im Weiteren als „Intelligentes Netz“ bezeichnen und mit IN abkürzen.

Definition 7.1.1 (Intelligentes Netz – erster Ansatz) *Eine Komponente in einem Transportprotokoll, die anhand von aktuellen Netzparametern den optimalen Pfad zum Datentransport aufgrund von gelernten Regeln auswählt, wird als* Intelligentes Netz *IN bezeichnet.*

Neben der Auswahl eines geeigneten Lernalgorithmus besteht das Problem darin, ausreichende Informationen in Form von Parametern aus dem Transportprotokoll zu bestimmen. Etablierte Protokolle wie TCP oder UDP stellen nur eine begrenzte Auswahl an Informationen zur Verfügung. Weiterhin sollte das zu verwendende Protokoll in der Lage sein, von sich aus mehrere Pfade zu bedienen. Beide Forderungen werden vom SCTP-Protokoll erfüllt, sodass dieses im weiteren Verlauf der vorliegenden Arbeit als Basis herangezogen wird. Für eine Assoziation können mehrere Pfade für den Datentransport eingestellt werden, wobei die Daten grundsätzlich über einen sogenannten Primärpfad gesendet werden und weitere Pfade, die Sekundärpfade, lediglich als Ausfalllösung genutzt werden. Als Abwandlung hiervon wurde u.a. in [Jungmaier 2005] und [Iyengar

et al. 2006] die Verwendung von SCTP in einem Multipfad-Szenario (MP-SCTP) untersucht. Basierend auf diesen Arbeiten, die den wichtigen Aspekt der Pfadwahl nicht in die Untersuchung mit einbezogen haben, soll in der vorliegenden Arbeit eine optimierte MP-SCPT-Variante aufgrund von Verfahren entwickelt werden, die dem maschinellen Lernen zuzuordnen sind. Aufgrund dieser Betrachtung können wir den Begriff des IN jetzt konkret fassen:

Definition 7.1.2 (Intelligentes Netz – Endversion) *Eine Komponente im SCTP-Tranportprotokoll, die anhand von Netzparametern, die ihrerseits vom SCTP-Kern ermittelt werden, den optimalen Pfad zum Datentransport innerhalb einer Assoziation einer MP-SCTP-Anwendung aufgrund von gelernten Regeln auswählt, wird als* Intelligentes Netz *IN bezeichnet.*

7.2 Klassifikationsverfahren

In diesem Abschnitt werden mögliche Verfahren aus dem Bereich der Statistik beschrieben, die für eine optimale Pfadwahl in Frage kommen. Nicht jedes Verfahren ist gleich gut geeignet, da wir aufgrund der speziellen Gegebenheiten im Netzwerk besondere Anforderungen an ein mögliches Verfahren stellen.

Da das Verfahren in einem Echtzeitszenario, d.h. in die Transportschicht eines bestehenden Protokolls eingebunden werden soll, muss die Zeit für die Berechnung der Klassifikationsfunktion möglichst kurz sein. Die zeitliche Komplexität ist demnach ein wichtiges Kriterium für die Auswahl eines geeigneten Lern-Algorithmus. Es bieten sich somit klassische statistische Verfahren an, die von ihrer Struktur sehr einfach gehalten sind. In der Praxis hat sich gezeigt, dass in vielen Fällen diese Verfahren bereits sehr gute Ergebnisse liefern (siehe [Witten 2001]). Statistische Verfahren sind darauf ausgelegt aus einer gegebenen Stichprobe Schlussfolgerungen zu ziehen. Daher nutzen die hier vorgestellten Verfahren grundsätzlich als Basis statistische Verfahren, um die benötigten mathematischen Modelle zu konstruieren.

Bei dem hier gestellten Problem der Pfadwahl ist die „präzise“ Klassifikation nicht entscheidend, da lediglich der vermeintlich bessere Kanal zur Übertragung gesucht wird. Die Vorhersagegenauigkeit ist somit nicht als Kernkriterium zu verstehen. Das später vorgestellte Zwei-Klassen-Problem wird aufgebrochen und auf ein n-Klassen-Problem (mit $n \rightarrow \infty$) transformiert. Dies lässt sich nur bzw. besonders gut mit linearen Verfahren erreichen, sodass wir als weitere Bedingung die Linearität zu untersuchen haben.

Weiterhin sollte die Lernphase von der Anwendung entkoppelt werden. Es soll kein „Instanzbasiertes Lernen“ zum Einsatz kommen. Beim *instanzbasierten Lernen* findet der eigentliche Lernvorgang erst zum Zeitpunkt der Klassifizierung einer neuen Instanz statt. Für unser Problem würde dies bedeuten, dass vor Versand für jedes Paket der „Lernalgorithmus“ aufgerufen wird, um den optimalen Pfad zu ermitteln. Ein solches Verfahren ist besonders gut geeignet, sich auf Änderungen im System sprich im Netzwerk anzupassen,

da sich auch stark ändernde Voraussetzungen wie beispielsweise eine extrem hohe bzw. niedrige Bandbreite des Kanals auf den Entscheidungsprozess auswirken. Allerdings hat diese Vorgehensweise auch Nachteile. Neben dem Problem, eine geeignete Gewichtung der Attribute aus den Trainingsdaten abzuleiten, ist die Speicherung der Instanzdaten höchst problematisch. Welche Instanzdaten sollen bzw. werden gespeichert? Speichert man möglichst viele Daten, so ist das Ergebnis ggf. sehr gut, dagegen hat man ein sehr schlechtes Laufzeitverhalten des Lernalgorithmus zu erwarten. Speichert man hingegen zu wenig Daten, ist das Ergebnis des Lernprozesses ggf. nicht ausreichend, um die Pfadwahl optimal durchzuführen.

Um diesem Problemen zu begegnen, wird in einer ersten Phase, der *Lernphase*, basierend auf Trainingsdatensätzen, eine Regel bzw. Faktoren einer mathematischen Gleichung „gelernt“, die dann dem IN als Regel zur Verfügung gestellt wird. Die Trainingsdaten werden konkret durch Testläufe unter verschiedenen Voraussetzungen ermittelt. Hierfür wird das in Abschnitt 8.3.1 beschriebene Testtool verwendet. Von der Parameterwahl bzw. der gewählten Testumgebung ist das Ergebnis der Lernphase entscheidend abhängig, sodass der Datenanalyse eine gesonderte Erörterung in Abschnitt 8 gewidmet wurde.

Zusammenfassend muss der Lernalgorithmus folgende Kriterien erfüllen:

- geringe zeitliche Komplexität
- Anwendung beruht auf linearen Gesetzmäßigkeiten
- Lernphase basierend auf Trainingsdatensätzen
- eine gelernte Regel muss sich in das IN integrieren lassen

7.2.1 Definitionen und Vorbereitung

Um einen eindeutigen Sprachgebrauch zu garantieren, werden einige Begriffe aus dem Bereich des Data Mining aufgeführt bzw. definiert.

Das vorgestellte Problem greift nur in einem Teilbereich des Data Minings, dem sogenanntem „maschinellen Lernen“ ein. In [Alpaydin 2008] wird maschinelles Lernen folgendermaßen definiert:

Definition 7.2.1 (Maschinelles Lernen) Maschinelles Lernen *heißt, Computer so zu programmieren, dass ein bestimmtes Leistungskriterium anhand von Beispieldaten oder Erfahrungswerten aus der Vergangenheit optimiert wird.*

Geht man von dieser Definition aus, so sind die Trainingsdaten, die mit Hilfe des Testtools (vgl. Abschnitt 8.3.1) ermittelt werden, die Beispieldaten, und das zu optimierende Leistungskriterium ist die optimale Pfadwahl.

Dieser Begriff kann weiter zerlegt werden, wobei man grundsätzlich vier verschiedene Arten des Lernens (vgl. [Witten 2001]) unterscheiden kann.

- das klassifizierte Lernen
- das assoziierende Lernen
- das Clustering
- die numerische Vorhersage

Das klassifizierende Lernen ist für das Pfadwahlproblem am interessantesten. Beim klassifizierenden Lernen wird basierend auf einer Beispielmenge von bereits klassifizierten Daten eine Lernregel abgeleitet, die es ermöglicht, unbekannte Beispiele zu klassifizieren. Die Trainingsdaten entsprechen dabei den Beispieldaten, die aufgrund der vorhandenen Informationen vollständig korrekt klassifiziert werden. Diese Informationen sind grundsätzlich korrekt klassifizierbar, da in den Trainingsdaten zusätzliche Informationen enthalten sind, die erst nach einem vollständigen Durchlauf zur Verfügung stehen. Als mögliche zusätzliche Information wird im Abschnitt Datenanalyse der Chunkdelay herangezogen. Für die genaue Definition, was unter dem hier verwendeten Begriff *Delay* zu verstehen ist, wird auf Abschnitt 8 verwiesen. Bei der Übertragung von Daten im „Echtszenario“ stehen diese zusätzlichen Informationen nicht mehr zur Verfügung, sodass die „unbekannten Beispiele“ nur die Netzsituation vor dem Paketversand berücksichtigen, die jetzt über die gelernte Regel zur Pfadwahl herangezogen wird.

Das zu lösende Problem lässt sich nicht auf den Zwei-Klassenfall – guter Pfad, schlechter Pfad – reduzieren, sondern es wird eine numerische Größe gesucht, die mehrere Pfade hinsichtlich ihrer Dienstgüte bewertet. Ausgehend von diesem numerischen Wert kann ein direkter Vergleich der Pfade erfolgen. Wird keine diskrete Klasse vorhergesagt, sondern ein numerischer Wert, so spricht man auch von einer *numerischen Vorhersage*. Die numerische Vorhersage kann als Variante des klassifizierenden Lernens aufgefasst werden.

Beim *assoziierenden Lernen* werden sämtliche Assoziationen zwischen den Attributen berücksichtigt, d.h. es gilt, Assoziationen zwischen Merkmalen zu entdecken. Diese Form des Lernens spielt für das IN keine Rolle und ist nur der Vollständigkeit halber erwähnt. Ebenso die *Clusteranalyse*, bei der keine zusätzlichen Informationen herangezogen werden, sondern in den Trainingsdaten, die den „Echtdaten“ bei einer realen Übertragung entsprechen, nach zusammengehörigen Beispielsätzen gesucht wird. Ähnliche Instanzen werden in sogenannten Clustern gruppiert.

Speziell im Bereich des maschinellen Lernens gibt es für das Ziel des Lernens, das Ergebnis des Lernalgorithmus und die Eingabeparameter eigene Begriffe, die hier eingeführt werden sollen, um die folgende Einführung der verwendbaren Algorithmen im Fachkontext beschreiben zu können.

Als Erstes wird das Ergebnis eines Lernverfahrens definiert. Unabhängig von der Form des Lernverfahrens wird für die Ausgabe immer die Konzeptbeschreibung erzeugt.

Definition 7.2.2 (Konzept und Konzeptbeschreibung) *Unter einem* Konzept *versteht man das „was“ gelernt werden soll, während man die Ausgabe des Lernverfahrens als* Konzeptbeschreibung *bezeichnet.*

Als Eingabe für das Lernverfahren wird eine Menge von unabhängigen Beispielen, die Trainingsmenge oder Instanzmenge, verwendet.

Definition 7.2.3 (Instanz und Instanzmenge) *Individuelle, unabhängige Beispiele eines Konzepts werden als* Instanz *bezeichnet. Die Instanzen, die einem Verfahren als Eingabe dienen, werden auch als* Instanzmenge *bezeichnet.*

Die Merkmale einer Instanz werden durch so genannte Attribute beschrieben. Im konkreten Fall sind dies die Netzparameter, die vom SCTP protokolliert werden.

Definition 7.2.4 (Attribute) *Ein* Attribut *ist ein Messwert für Aspekte einer Instanz. Man unterscheidet zwischen ordinalen und nominalen Werten.*

Die Auswahl der relevanten Attribute ist ein besonders wichtiger Aspekt bei der Betrachtung eines bestimmten Lernverfahrens. Zum einen ist das Verhältnis der Attribute untereinander entscheidend, zum anderen sollen redundante und damit abhängige Attribute eliminiert werden. Für das Problem der Pfadwahl wird hierauf in Abschnitt 8 speziell eingegangen.

7.3 Lineare Diskriminanzanalyse

Die *Diskriminanzanalyse* kann zur Klassifikation von Daten verwendet werden. Ausgangspunkt ist ein Trainingsdatensatz mit bekannter Gruppenzugehörigkeit, auf dessen Basis die Gruppenzugehörigkeit neuer Objekte, deren Gruppenzugehörigkeit demnach nicht bekannt sind, vorhergesagt werden. Um die Klassifikation vornehmen zu können, wird nach Merkmalen in den Trainingsdaten gesucht, die die Gruppenzugehörigkeit signifikant beschreiben. Im Gegensatz dazu ist bei der *Clusteranalyse* keine Gruppenzugehörigkeit bekannt, diese wird erst durch die Analyse ermöglicht, indem „ähnliche“ Objekte zusammengefasst werden. Aufgrund der Komplexität der Clusteranalyse ist ein solches Vorgehen für das IN nicht praktikabel.

Der Begriff Diskriminanzanalyse leitet sich vom Begriff *Diskrimination* – Trennen – ab, da die einzelnen Datensätze aufgrund der Gruppenzugehörigkeit getrennt werden. Ziel ist es somit eine Funktion (Trenn- oder Diskriminanzfunktion) zu finden, die eine optimale Trennung der Gruppen ermöglicht. Die Anzahl der Gruppen ist dabei nicht begrenzt, wobei der einfachste Fall das Zwei-Gruppen-Problem ist, welches in Form von zweidimensionalen Diagrammen anschaulich abgebildet werden kann. Von einer *linearen Diskriminanzanalyse* spricht man, wenn die Gruppen sich mittels einer linearen Funktion trennen lassen.

Definition 7.3.1 (Diskriminanzfunktion) *Es wird eine Abbildung aus dem n-dimensionalen Merkmalraum in den eindimensionalen Raum der Diskriminanzvariablen gesucht, wobei die* lineare Diskriminanzfunktion *als*

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_n \cdot x_n \tag{7.1}$$

geschrieben wird. Der Wert y wird als Diskriminanzvariable *bezeichnet. Jeder Wert x_i beschreibt ein Merkmal der n gegebenen Merkmale des Datensatzes. Die* Diskriminanzkoeffzienten β_i *sind aus den Trainingsdaten zu schätzen.*

Die Schreibweise aus Formel 7.1 findet sich als Teil des Modells in Formel 7.2 wieder und ist demnach eine Kurzschreibweise für die Trennfunktion.

Die Diskriminanzanalyse gehört zu den *multivariaten* Verfahren, da ein einzelner Datensatz durch mehr als ein Merkmal beschrieben wird.

Beispiel 7.3.1 *Im Abschnitt 8 wird ausführlich auf die Merkmale und Parameter von SCTP eingegangen, die für die Diskriminanzanalyse verwendet werden können, auf die aber zur Veranschaulichung vorgegriffen werden soll.*

Annahme: Ein Pfad soll durch seine Rundenlaufzeit srtt *und einen Parameter* verh*, der die auf dem Kanal befindlichen Daten ins Verhältnis zu der maximal möglichen Datenmenge in Relation stellt, ausgeprägt sein.*

Gegeben: Die Qualität eines Pfades soll als Zwei-Klassenproblem mit den Klassen gut *und* schlecht *beschrieben werden. Für die Trainingsdaten muss die Klasseneinteilung gegeben sein. Aufgrund eines aus den Protokolldateien der Versuchsreihen abgeleiteten* Chunk-Delays *wird die Einteilung vorgenommen.*

Gesucht ist eine Trennfunktion y, genauer gesagt die Koeffizienten β_1 und β_2

$$y = \beta_1 \cdot srtt + \beta_2 \cdot verh,$$

die es ermöglicht, nur in Kenntnis aktueller Werte für die srtt *und* verh *die Diskriminanzvariable y anzugeben.*

Eine konkrete aus den Trainingsdaten abgeleitete Funktion ist beispielsweise in Formel 8.2 dargestellt.

Da es sich um eine lineare Funktion handelt, können mehrere Diskriminanzvariablen direkt miteinander verglichen werden, sodass ein Kriterium zur Verfügung steht, mit dem entschieden werden kann, welcher Kanal von n Kanälen aktuell der leistungsfähigste ist.

Praktisch werden die Diskriminanzkoeffizienten durch Maximierung eines Diskriminanzmaßes gefunden.

7.4 Einschub: Modelle und Formelnotation

In der Statistik wird immer wieder auf die Modellbildung zurückgegriffen. Für die Programmierung in *R* hat sich deshalb eine *Formelnotation* durchgesetzt, die es ermöglicht, selbst komplexe Zusammenhänge einfach zu beschreiben. Mit der Formelnotation können sämtliche in der vorliegenden Arbeit verwendeten Modelle, angefangen bei der linearen Regression über die Lineare Diskriminanzanalyse bis hin zu den Entscheidungsbäumen, beschrieben werden. Daher wird an dieser Stelle eine kurze Einführung gegeben.

Eine Formel kann in der Form

$$y \sim modell \qquad \text{oder auch} \qquad zielgroesse \sim einflussgroessen$$

angegeben werden.

Der Zusammenhang zwischen der *abhängigen Variablen* y und den Einflussgrößen, dem Modell, wird unter Verwendung von mathematischen Operatoren beschrieben. Ein lineares Modell

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \epsilon \tag{7.2}$$

wird folgendermaßen angegeben:

$$y \sim x1 + x2$$

Weitere Variablen werden mit dem Plus-Operator hinzugenommen bzw. mit dem Minus-Operator entfernt. Im obigen Beispiel ist der Achsenabschnitt β_0 enthalten. Soll dieser nicht im Modell modelliert werden, so kann er mit -1 entfernt werden.

7.5 Klassifikations- und Regressionsbäume

Klassifikationsbäume stellen eine direkte Verallgemeinerung der nichtlinearen Diskriminanzanalyse dar. Anstelle einer einzelnen Funktion zur Klassifikation wird eine Folge von Entscheidungen festgelegt, die sukzessive unter Verwendung von Partitionierungsregeln die Trainingsdatenmenge X in kleinere Teilmengen X_i zerlegen. Ziel ist es, auf der untersten Ebene möglichst homogene Gruppen von Elementen zu erhalten. Die Partitionierungsregeln werden auf die Merkmale x_i angewendet.

Liegen anstelle einer nominalskalierten metrisch skalierte Diskriminanzvariablen vor, so bezeichnet man einen Klassifikationsbaum als *Regressionsbaum*. Eine grundlegende Einführung in die Theorie der Regressionsbäume kann in [Breiman et al. 1984] nachgelesen werden, an dieser Stelle soll lediglich das Beispiel aus dem vorherigen Abschnitt fortgeführt werden. Da die SCTP-Parameter in Form von Messwerten vorliegen, wird die Klassifikation über Regressionsbäume vorgenommen.

Beispiel 7.5.1 *Greift man das Beispiel 7.3.1 auf, so gilt es, die Trainingsdatenmenge über die Parameter $x_1 = srtt$ und $x_2 = verh$ so zu zerlegen, dass die Blätter des Baums jeweils einen möglichen Delaybereich abdecken. In Abbildung 8.14 sind konkrete Regressionsbäume abgebildet. Jeder Knoten enthält eine Regel der Form $x_j < Messwert$ und zerlegt die Trainingsdaten in disjunkte Teilmengen.*

Für das Beispiel wird der Baum in Abbildung8.14(a) verwendet.

Seien die Messergebnisse für Kanal K_1 mit $M_1 = \{x_1 = srtt = 300 \quad und \quad x_2 = verh = 400\}$ gegeben, so ergibt sich ein geschätzter Delay von $d_{M_1} = 878,9$, wogegen für einen Kanal K_2 mit der Messung $M_2 = \{x_1 = srtt = 1000 \quad und \quad x_2 = verh = 500\}$ ein Delay von $d_{M_2} = 920$ geschätzt wird. Das IN würde die Daten demnach auf Kanal $K1$ senden.

Durch den Regressionsbaum werden Regeln abgeleitet, die dann im IN zur Pfadwahl angewendet werden. Die Lernphase kann somit aufgrund der Trainingsdaten im Vorfeld erfolgen. Zur Erzeugung der konkreten Bäume, wie sie auch im Abschnitt 8.7.3 beschrieben sind, wurde die vom Statistikprogramm *R* zur Verfügung gestellte Funktion *rpart* aus der gleichnamigen Mathematischen-Bibliothek verwendet.

8 Datenanalyse

8.1 Welche Parameter stellt SCTP zur Verfügung?

Als Parameter für die Pfadwahl und somit als Grundlage für die durchzuführende Klassifizierung werden Werte herangezogen, die direkt von SCTP bereitgestellt bzw. leicht aus den bereitgestellten Parametern berechnet werden können. SCTP ermittelt für die Fluss- und Überlastkontrolle[1] bzw. für die Ausfallsicherung[2] bereits wichtige Informationen, die für die Analyse genutzt werden können. Da diese Informationen ständig aktualisiert werden, spiegeln sie die aktuelle Situation im Netz sehr genau wider.

8.2 Welche Parameter sind für die Pfadwahl geeignet?

Die Parameter von SCTP können grob in zwei Gruppen eingeteilt werden. In Parameter, die Informationen nur für die gesamte Assoziation bereitstellen, und in solche Parameter, die für jeden Pfad gesondert zur Verfügung stehen. Eine Auswahl der zur Verfügung stehenden Parameter ist in Abbildung 8.1 nach Aufgaben der Parameter bei der Datenübertragung sortiert angegeben.

Die im Folgenden beschriebenen SCTP-Parameter geben die Gesamtsituation im Netz sehr breitgefächert wieder.

Beschreibung der von SCTP gegebenen Parameter

Der Parameter *flightsize* gibt die Menge an Daten in Byte an, welche im Netzwerk unterwegs sind. Dies sind die Daten, die in das Netzwerk zu einer spezifischen IP-Adresse eingespeist wurden, deren Empfang jedoch noch nicht vom Empfänger bestätigt worden ist. Der Parameter *flightsize* ist für jeden Pfad einzeln definiert. Wenn die gesamte Menge an Daten bezogen auf sämtliche Pfade benötigt wird, kann man diese unter dem Parameter *Outstanding-Bytes* fassen. Die Parameter *flightsize* und *Outstanding-Bytes* werden von SCTP zur Staukontrolle genutzt.

Ein weiterer wichtiger Parameter ist die Größe des *Empfangsfensters*[3] (*rwnd*), das die Größe des benutzbaren Eingangsbuffers auf der Empfängerseite angibt. Bei einem Ver-

[1] flowcontrol
[2] failover mechanism
[3] receiver window

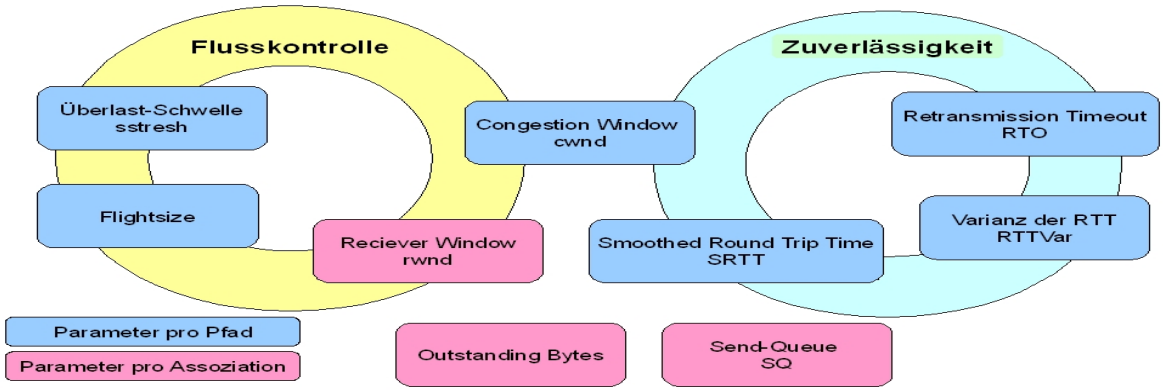


Abbildung 8.1: Wichtige SCTP-Parameter unterteilt in Parameter für Zuverlässigkeit und Flusskontrolle sowie eine Einteilung nach Assoziationsparametern und Pfadparametern

bindungsaufbau legt der Empfänger den für diese Assoziation von ihm zur Verfügung gestellten Speicherplatz fest und übermittelt diese Größe dem Kommunikationspartner. Dies ist Bestandteil des 4-Wege-Handshakes zum Aufbau der SCTP-Assoziation. Dieser vom Empfänger festgelegte Wert wird mit Hilfe des INIT-ACK-Chunks übermittelt. Der *rwnd*-Wert für den aktuellen Zeitpunkt wird beim Sender durch Informationen, die ihm durch die SACKs zugänglich gemacht werden, kontinuierlich neu berechnet. Hierfür ist im SACK das Feld *advertised rwnd* (a_rwnd) reserviert, in welches der derzeitig vorhandene Speicherplatz des Empfängers hinterlegt wird. Mittels dieser Information berechnet der Sender den neuen *rwnd*-Wert. Der Parameter *rwnd* ist pro Assoziation gegeben und bei der Staukontrolle sehr hilfreich.

Der Parameter, der auf die Größe des Sendefensters[4](*cwnd*) abstellt, gibt an, wie viele Daten zu dem aktuellen Zeitpunkt über einen bestimmten Pfad in das Netzwerk eingespeist werden dürfen.

Weiterhin gibt es den Parameter *sendqueue*, der die Größe der Daten angibt, die auf ihr Absenden aus der Sende-Schlange[5] warten.

[4] congestion window
[5] sendqueue

Nachdem die wichtigsten Parameter für die Staukontrolle erwähnt wurden, kann auf spezielle Parameter der Staukontrolle eingegangen werden. Da spezielle Parameter nur in Kenntnis der grundsätzlichen Mechanismen der Staukontrolle erörtert werden können, wird nun kurz auf die Mechanismen der Staukontrolle eingegangen. Eine präzise Beschreibung der Staukontrolle ist Thema des in SCTP einführenden Abschnitts 6.4.

Die *Staukontrolle* hat zur Aufgabe, bei Anzeichen einer überlasteten Leitung die Menge der in das Netzwerk eingespeisten Daten zu reduzieren und somit einer vollständigen Überlastung des Netzwerkes vorzubeugen. Als Kriterium für eine überlastete Leitung wird der Verlust von Datenpaketen betrachtet. Wenn es zu einer Neuübertragung von Datenchunks kommt, wird das Sendefenster drastisch reduziert. Zu Neuübertragungen kann es dabei auf zwei verschiedenen Weisen kommen. Zum einen kann der Timer für die Neuübertragung abgelaufen sein, zum anderen kann es durch das Auftreten von Lücken auf der Empfängerseite zu einer schnellen Neuübertragung, der sogenannten *Fast-Retransmission* kommen. Bei dem MP-SCTP-Szenario stellt die schnelle Neuübertragung eine besondere Herausforderung dar und wird daher in Abschnitt 9.1 in diesem Zusammenhang beschrieben.

Der Sender wartet auf eine Bestätigung der Ankunft des Chunks. Wenn die Bestätigung im SACK den Sender erreicht, wird der Timer zurückgesetzt. Nach Ablauf des Timers wird der Chunk erneut übertragen. Im zweiten Fall werden in den SACKs Informationen über aufgetretene Lücken in der TSN Ankunftsreihenfolge dokumentiert. Falls dem Sender das Fehlen von fünf fehlenden TSN in einer Reihe bekannt wird, werden die entsprechenden Chunks erneut übertragen. Im Vergleich zu der timerbasierten Neuübertragung wird das Fehlen von Chunks früher erkannt und die Chunks somit schneller neu übertragen.

Dies führt zu dem Parametern *rtx_3_Timer*. Beim Versenden von Datenpaketen wird jeder Chunk mit einer aufsteigenden eindeutigen Nummer versehen, der TSN[6]. Weiterhin wird ein Timer gestartet, der sogenannte *rtx_3_Timer*.

Die *Rundenlaufzeit*[7] *RTT* ist die Zeit, die eine Nachricht und die entsprechende Empfangsbestätigung benötigt, um das Netzwerk vom Sender zum Empfänger und zurück zu durchlaufen. Berechnet wird die *RTT*, indem die Ankunftszeit der Empfangsbestätigung von dem Zeitpunkt der Versendung abgezogen wird. Ein ähnlicher Mechanismus wird vom Netz-Tool *Ping* benutzt, das ICMP-Timestamps setzt. Hier werden die Zeitstempel für das Auslösen[8], das Empfangen[9] und das Senden[10] gesetzt. Die RTT ist somit die Differenz zwischen dem Empfangen- und Auslöse-Zeitstempel. In SCTP wird die RTT vom Primärpfad mittels gesendeter Nachrichten bestimmt, meist Datenchunks, die durch den

[6] transmission sequence number
[7] round trip time
[8] orginate
[9] receive
[10] transmit

entsprechenden SACK-Chunk bestätigt wurden. Die RTT der Sekundärpfade wird durch das Versenden von Heartbeat-Chunks und deren Bestätigung[11] ermittelt. Die RTT ist variabel und von der aktuellen Netzwerkbelastung abhängig.

Darüber hinaus gibt es noch die *SRTT*[12], hier als Parameter *srtt*. Die *SRTT* ist der geglättete Schätzwert der Rundenlaufzeiten[13] und deren Schwankung *RTT-Variation*[14].

Die Dauer des bereits eingeführten *rtx_3_Timers* ist variabel und wird durch das *Retransmission-Time-Out*, abgekürzt *RTO*, festgelegt. Dieser wird auf Basis der SRTT folgendermaßen berechnet:

$$RTO = SRTT + 4 \cdot RTTVAR$$

Bei einem Multihomed-Empfänger werden diese drei Werte für jede Empfangs-Adresse benötigt.

Ableitung von Regeln aus den Parametern zur Bewertung einer bestimmten Netzsituation

Aus den Definitionen der einzelnen Parameter lassen sich bereits intuitiv Regeln für das Versenden bzw. Hinweise auf die Netzwerksituation ableiten. Somit weist ein Anstieg der Sendqueue-Länge auf einen Engpass bei der Datenübertragung hin, ebenso wie das Verkleinern des *cwnd*.

Für die Untersuchungen sind die Parameter von Interesse, die auf einen einzelnen Pfad heruntergebrochen werden können, da jeder einzelne Pfad hinsichtlich seiner Güte bewertet werden soll. Von den zur Verfügung stehenden Parametern werden die *srtt*, *flightsize* und das *cwnd* herausgegriffen.

Die *srtt* erscheint besonders gut geeignet zu sein, da sich aus der Rundenlaufzeit direkt eine Aussage über die Leistungsfähigkeit des Kanals ableiten lässt. Dieser Wert für sich genommen ist aber nicht ausreichend, um die komplexe Situation im Netz zu beschreiben. In Abschnitt 8.6 wird dies u.a. u.V. der Abbildung 8.5 veranschaulicht.

Die bereits gesendeten aber noch nicht als angekommen markierten – *geackten* – Chunks sind von besonderem Interesse, da man an diesem Wert erkennen kann, ob es ggf. größere Probleme auf der Leitung gibt, insbesondere wenn dieser Wert die Kapazität der Leitung übersteigt. Allerdings ist dieser Parameter in der Rohform nur schwer zu handhaben, da je nach Bandbreite des Kanals dieser Wert sehr unterschiedlich ausgelegt werden muss. Ein schmaler Kanal kann bereits mit n Kbytes völlig überlastet sein, während ein

[11] heartbeat ACK chunks
[12] Smoothed RTT
[13] RoundTripTime
[14] RTTVAR

breiter Kanal locker $50 \cdot n$ Kbytes verkraften kann. Es gilt also, die *flightsize* möglichst kapazitätsneutral abzubilden. Hierfür kann man einen gemischten Parameter verwenden, der ein Verhältnis zwischen *cwnd*, also der maximal möglich zu sendenden Datenmenge und der tatsächlich auf dem Kanal befindlichen Datenmenge, also der *flightsize*, herstellt. Dieser Verhältniswert, im Folgenden mit *verh* abgekürzt, kann beispielsweise wie folgt berechnet werden:

$$verh = \frac{cwnd}{flightsize \cdot 100} \tag{8.1}$$

Parameter, die zur Beschreibung des Modells verwendet werden, werden auch *Attribute* genannt. Bisher wurden nur Attribute betrachtet, also Werte, die als sogenannte *Einflussgrößen* Teil des Modells sind. Neben diesen Attributen muss auch die Zielgröße, also ein Entscheidungskriterium, festgelegt werden. Speziell für das Problem der Pfadwahl wird ein Kriterium gesucht, dass es ermöglicht, die konkrete Qualität eines Kanals zu beschreiben. Diese Zielgröße steht im späteren Einsatz des IN nicht mehr zur Verfügung, sie wird lediglich zum Trainieren des IN benötigt. Falls diese Zielgröße als Parameter zur Verfügung stehen würde, würde das IN zur Pfadwahl nicht benötigt werden, da dann der günstigste Pfad direkt abgelesen werden könnte.

Eine solche Zielgröße steht demnach nicht direkt zur Verfügung. Unter Verwendung der vom Testtool geschriebenen Protokolldateien, die auf Sender- und auf Empfängerseite die Datenübertragung vollständig beschreiben, kann eine solche Zielgröße nach Ablauf der Übertragung berechnet werden.

Als Zielgröße wird ein *Chunkdelay* angenommen, der eine Aussage darüber macht, wie lange ein einzelner Chunk bzw. eine festgelegte Gruppe von Chunks vom ersten Senderaufruf bis zur Ausgabe beim Zielsystem benötigt. Wesentlich für das IN ist, dass dieser Chunkdelay so gewählt wird, dass die Werte für die verschiedenen Szenarien vergleichbar sind. Im folgenden Abschnitt 8.3 wird konkret erläutert, wie sich der Begriff des Chunkdelays aus den Testdaten ableiten lässt.

In den folgenden Abschnitten wird zu prüfen sein, ob sich die theoretisch durchgeführte Parameterwahl in der Praxis behaupten kann. Voraussetzung hierfür ist die Generierung von aussagekräftigen Test- bzw. Trainingsdaten, die zur Konditionierung des IN genutzt werden können. Von der Qualität der Trainingsdaten sind die Ergebnisse der späteren Modellierungs- bzw. Testphase abhängig.

8.3 Wie sind die Daten entstanden?

In diesem Unterabschnitt wird die Erzeugung des Trainingsdatensatzes erläutert. Neben der Auswahl von repräsentativen Attributen ist die „korrekte“ Generierung der Trainingsdaten ein wesentlicher Aspekt für das Ausgabeergebnis. Die Daten müssen so umfangreich sein, dass sämtliche Facetten des Problembereichs abgedeckt werden, soll-

ten aber auch so klein wie möglich gehalten werden, dass die Auswertung der Daten in angemessener Zeit erfolgen kann.

8.3.1 Einschub: Das Testtool und die SCTPLib

Das Testtool orientiert sich an NetPerf, einem weit verbreiteten Testtool für die Netzwerkleistungsmessung in klassischen TCP/IP-Netzen. Das Testtool besteht aus einer Server- und einer Client-Komponente. Die Client-Komponente generiert Daten einer festen Größe, die dann in bestimmten Zeitabschnitten in das Netz eingespeist werden. Der Server hingegen läuft auf dem Zielrechner und liefert dementsprechend Informationen über die eintreffenden Datenchunks. Um den realen Traffic, d.h. den Traffic incl. Paketheader und Kontrollinformationen, in die Betrachtung einfließen zu lassen, kann der Datentransfer zusätzlich über einen Netzwerksniffer, hier wurde *Wireshark* verwendet, abgefangen und zur weiteren Auswertung in eine Textdatei exportiert werden.

SCTP ermöglicht eine Event-gesteuerte-Programmierung, indem sogenannte *Notifications*, also Benachrichtigungen zu bestimmten Ereignissen, gesendet werden, die dann vom Anwenderprogramm abgefangen und ausgewertet werden können. Hierzu definiert das Anwenderprogramm sogenannte *Callback-Funktionen*, die nach Bekanntgabe beim SCTP-Kern automatisch aus den SCTP-Funktionen heraus aufgerufen werden. Über diese Eventsteuerung ist es möglich, den gesamten Datentransfer mit seinen einzelnen Phasen zu protokollieren.

Folgende Notifications werden von der SCTPLib angeboten:

dataArriveNotif Zeigt an, dass neue Daten eingetroffen sind. Als Parameter erhält die aufgerufene Funktion neben den Informationen zur Assoziation, wie die Assoziations-ID und die zugehörige Stream-ID, auch Informationen zu den eingegangenen Daten. So wird die Länge der Nutzdaten mitgeteilt, mit deren Hilfe die Daten ausgelesen werden können. Hierfür erhält die Funktion einen Zeiger auf eine Upper-Layer-Data-Struktur. Mit dieser Benachrichtigung ist der Transfer der Daten aus Sicht von SCTP abgeschlossen.

sendFailureNotif Daten konnten nicht gesendet werden. Die in der Sendeschlange befindlichen User-Messages warten auf den Assoziations-Startup.

networkStatusChangeNotif Der Status der SCTP-Assoziation hat sich geändert. Status bezieht sich hierbei auf die verwendeten Pfade, so wird beispielsweise angezeigt, wenn ein Pfad nicht mehr verwendet werden kann und auf einen alternativen Pfad umgeschaltet wurde.

communicationUpNotif Eine neue SCTP-Assoziation steht für den Datentransfer zur Verfügung.

communicationLostNotif Die Assoziation wurde beendet.

communicationErrorNotif Es ist ein schwerwiegender Fehler beim Transfer aufgetreten.

restartNotif Die SCTP-Assoziation wurde neu aufgesetzt.

shutdownCompleteNotif Die SCTP-Assoziation wurde über einen ordentlichen Shutdown heruntergefahren und ist somit beendet.

Für das Testtool wurde der communicationUp-Notif auf Senderseite und der dataArrive-Notif auf Empfängerseite implementiert. Der grundsätzliche Verlauf einer Testmessung ist in Abbildung 8.2 dargestellt.

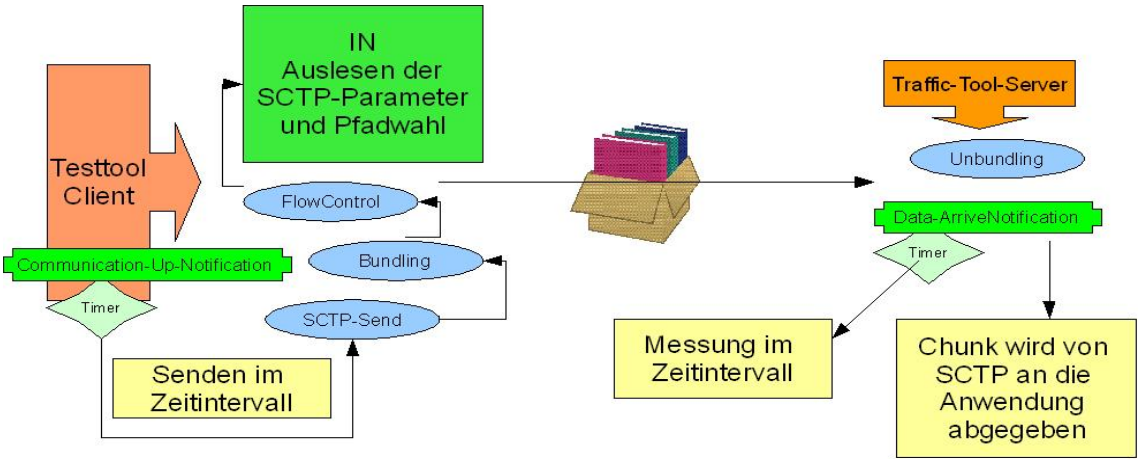


Abbildung 8.2: Genereller Aufbau des Testtools zur Generierung der Trainings-Daten auf Basis der SCTPLib

Neben dem o.a. Benachrichtigungssystem bietet die SCTPLib Timer zur Steuerung und Überwachung der Netzaktivität an. Auf Senderseite erfolgt der Datenversand in vorher festgelegten timergesteuerten Zeitintervallen. Standardmäßig wird jede Sekunde eine bestimmte Anzahl von Chunks einer vorher festgelegten Größe gesendet. Versenden heißt in diesem Fall, dass die Daten der SCTPLib-Funktion *SCTP_Send* übergeben werden. Ab diesem Zeitpunkt ist die SCTPLib für den Versand verantwortlich, d.h. wie viele Chunks zu einem Paket zusammengefasst und wann die Daten tatsächlich – physikalisch – auf den Pfad ausgegeben werden, wird von der SCTPLib festgelegt. Soll in diesen Prozess eingegriffen werden, muss demnach auch in den Programmcode der SCTPLib eingegriffen werden. Das IN wird demnach in die SCTPLib integriert.

Zurück zu der Timersteuerung. Zusätzlich zu den Informationen, die von SCTP für den Versand benötigt und im zugehörigen Paket codiert werden, wird in den Daten die aktuelle TSN des Chunks und der Zeitpunkt des Versands codiert, sodass auf Empfängerseite diese Information zur Bestimmung des *Chunkdelays* herangezogen werden kann. Neben dem dataArriveNotif wird auf der Empfängerseite ebenfalls ein Timer implementiert,

der es ermöglicht, die eingegangenen Daten eines bestimmten Zeitintervalls zu ermitteln.

Zusätzlich werden zu diesen Informationen, die bereits standardmäßig von der SCTPLib bereitgestellten werden, die Parameter *srtt* und *verh* sowie die Zielgröße *delay* für die Auswertung der Daten benötigt. Alle SCTP-Parameter können im Programmlauf über speziell dafür vorgesehene Funktionen ausgelesen werden, sodass die Frage zu klären ist, wann der ideale Zeitpunkt zur Bestimmung und Ausgabe der Parameter erreicht ist. Während der Flusskontrolle (vgl. Abschnitt 6.4) findet im Standard-SCTP die Pfadwahl statt. Allerdings wird hier lediglich festgelegt, ob auf dem Primärpfad oder, falls dieser nicht mehr erreichbar sein sollte, die Daten auf einem Sekundärpfad gesendet werden sollen. An dieser Stelle kann das IN eingreifen und die Pfadwahl übernehmen und somit werden für jeden Testlauf hier auch die Informationen der SCTP-Parameter sowie des aktuell zu sendenden Chunks, der über die TSN eindeutig angesprochen werden kann, ausgegeben.

Als Zielgröße wurde der Chunkdelay festgelegt. Dieser kann allerdings erst nach Eingang auf Empfängerseite, d.h. als Folge des dataArriveNotif, ermittelt werden. Wesentlich für die Erfassung des Chunkdelays ist weniger der absolute Zahlenwert, sondern die Vergleichbarkeit bei Anwendung auf verschiedene Szenarien, Bandbreiten und Nutzdatengrößen. Nach Abschluss einer Testreihe wurden die Logdateien der Messungen vom Sender und vom Empfänger zusammengefügt und für jeden Chunk über seine TSN die Zeitverzögerung durch die Übertragung berechnet. Um vergleichbare Chunk-Delay-Werte zu erhalten, wurden immer die ersten zwanzig eingehenden Chunkdelays pro Zeiteinheit gemittelt, sodass pro Sekunde ein repräsentativer Delay-Wert vorliegt. Die SCTP-Parameter sind relativ schwerfällig, sodass jedem Delay-Wert genau ein Parameterpaar *(srtt,verh)* zugeordnet werden kann.

Die Trainingsdaten sollen ein breites Spektrum an möglichen Netzsituationen abdecken, um so die Vorhersage von unbekannten nicht in den Testszenarien enthaltenen Situationen zu ermöglichen. So wurden verschiedene Testläufe mit unterschiedlich konfigurierten Pfaden durchgeführt. Die verschiedenen Bandbreiten wurden über Nistnet (vgl. Abbildung 5.1) eingestellt. Es wurden drei verschiedene Grundpfade $(B1, B2, B3) = (6000, 12500, 25000)$ eingestellt. Als Erstes wurde die mögliche maximale Auslastung der jeweiligen Kanäle ermittelt. Auf jedem Pfad wurden Daten übertragen, und zwar so, dass die zu übertragenden Daten knapp über dem maximal Möglichen liegen sollten. So wurde erreicht, dass am Anfang der Übertragung gute bis sehr gute Übertragungsraten erreicht werden konnten, die im Verlauf des Experiments zwangsläufig immer ungünstiger wurden. Somit ergab sich ein breites Spektrum an guten und schlechten Übertragungen mit den dazugehörigen SCTP-Parametern bzw. den Delay-Werten als Zielgröße.

Konkrete Versuchsdurchführung:

- Datenübertragung timergesteuert über einen Zeitraum von jeweils 60 Sekunden pro Grundpfad B_i.

- Datengenerierung erfolgt über einen Generator-Timer, der jede Sekunde eine feste Anzahl an Datenchunks absendet. Es wurde eine feste Chunkgröße von 250 Bytes festgelegt. Die Anzahl der gesendeten Chunks richtete sich nach der vorher festgelegten bzw. ermittelten maximal möglichen Datenmenge auf dem Kanal.

Sämtliche Daten wurden in einer Datei als Trainings-Datensatz zusammengeführt, indem die verwendete Bandbreite in den Datensatz aufgenommen wurde. Da die Trainingsdaten als Eingabe für die Trainingsphase der Algorithmen verwendet werden sollen, wurde eine grobe Klassifizierung durchgeführt, indem ein fester Schwellwert des Delays zur Trennung der Klassen *gut* und *schlecht* eingeführt wurde.

Ein Auszug aus der Datei ist zum besseren Verständnis in Tabelle 8.1 aufgetragen.

nr	srtt	verh	delay	klasse	band
1	1545.00000	308.28083	441.72222	gut	sechs
2	90.00000	388.45609	473.05556	gut	sechs
3	90.00000	422.29635	430.94444	gut	sechs
4	140.88889	813.21068	467.44444	gut	sechs
5	347.05556	1216.89994	464.61111	gut	sechs
6	438.33333	1620.52156	461.77778	gut	sechs
7	491.72222	2023.77024	459.00000	gut	sechs
116	6864.00000	141.35432 1	4479.16667	schlecht	sechs
117	6864.00000	138.96913 1	4727.55556	schlecht	sechs
118	1545.00000	130.08661	211.31250	gut	zwoelf
119	90.00000	218.99455	351.25000	gut	zwoelf
120	90.00000	332.59767	516.37500	schlecht	zwoelf
121	319.75000	94.93414	652.31250	schlecht	zwoelf
307	621.00000	1117.32107	349.50000	gut	zwanzig
308	606.00000	1006.72932	392.00000	gut	zwanzig
309	678.00000	916.26597	434.56250	gut	zwanzig
310	643.00000	840.85104	477.12500	gut	zwanzig
311	700.00000	807.65122	522.06250	schlecht	zwanzig
312	790.00000	748.59189	562.12500	schlecht	zwanzig
313	840.00000	697.63014	604.68750	schlecht	zwanzig

Tabelle 8.1: Auszug aus den Trainingsdaten

8.4 Wissensentdeckung

Bevor die Modelle und Verfahren aus dem aus Abschnitt 7 zur Anwendung kommen können, müssen die Daten sich in einem „auswertbaren“ Zustand befinden. Daher wer-

den in einer vorangestellten *Analysephase* die Daten analysiert und einer Vorverarbeitung zugeführt. Gerade für den hier benötigten Bereich der interaktive Analyse von vorher gesammelten Trainingsdaten mit Hilfe von Lernverfahren hat sich der Begriff der *Wissensentdeckung* etabliert. Grundsätzliche handelt es sich um eine universell einsetzbare, prozessgesteuerte Vorgehensweise. Folgt man der Definition von Fayyad et.al (vgl. [Fayyad et al. 1996]) so kommt man zu folgender Definition.

Definition 8.4.1 (Wissensentdeckung) *Unter* Wissensentdeckung *bzw.* Knowledge Discovery in Databases*(KDD) ist der nicht triviale Prozess der Identifikation gültiger, neuartiger, unter Umständen nützlicher und schlussendlich nachvollziehbarer Muster in den Daten zu verstehen.* [15]

Wissensentdeckung ist somit ein mehrstufiger Prozess, wobei bei der Anzahl der notwendigen Prozessschritte und deren Ausformulierung die Meinungen in der Literatur leicht abweichen. So wurde in [Lindner und Studer] das sechsstufige CRISP-Prozessmodell vorgestellt. In [Fayyad et al. 1996] und [Grob 1999] werden fünfstufige Modelle favorisiert, wobei sich die einzelnen Phasen allerdings unterscheiden. Eine mögliche Aufteilung in einzelne Prozessschritte ist in Abbildung 8.3 visualisiert.

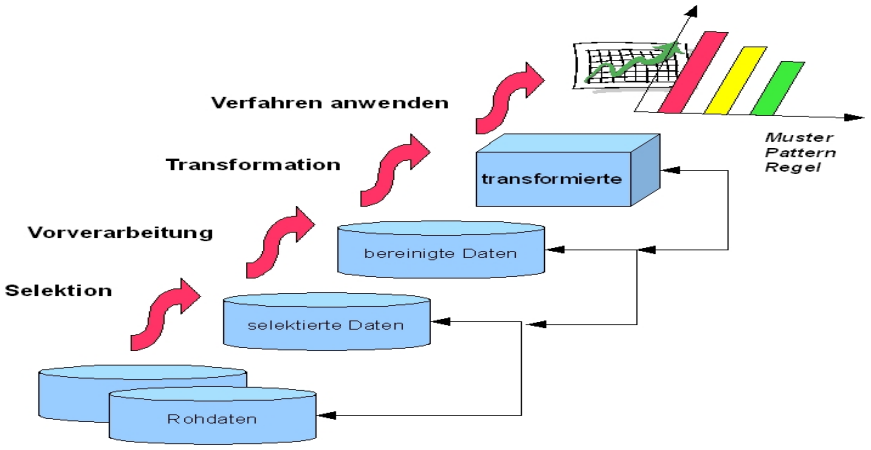


Abbildung 8.3: Prozess der Wissensentdeckung

In einer ersten Phase werden die zu untersuchenden Daten aus einer Datenquelle *extrahiert* und *selektiert*. Im konkreten Fall werden die Daten unter Zuhilfenahme des in

[15] KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimateandly understandable patterns in data

Abschnitt 5.2 festgelegten Testaufbaus inklusive des hierfür geschriebenen Testtools unter verschiedenen Bedingungen im Netz generiert. Wir erhalten einen Rohdatensatz, der sich aus sämtlichen Testreihen zusammensetzt. Allerdings sind die so entstandenen Daten selten direkt für die Anwendung des maschinellen Lernens geeignet. Es findet eine Auswahl der relevanten Attribute, die sogenannte *horizontale Selektion*, statt. Welche Datensätze endgültig als Grundlage für das ausgewählte Verfahren verwendet werden, wird ebenfalls entschieden. Bei dieser *vertikalen Selektion* werden u.a. redundante, ungeeignete, fehlerhafte und unvollständige Daten aussortiert bzw. in der folgenden *Vorverarbeitungsphase* mittels zusätzlicher Information korrigiert. Hierbei ist besonders darauf zu achten, dass der Informationsgehalt der Daten nicht signifikant verändert wird. Dieser Aspekt wird in den folgenden Abschnitten an den konkreten Beispielen ebenfalls Teil der Arbeit sein.

Die *Datentransformation* überführt die Daten in ein Schema, sodass sie von dem gewählten Verfahren ausgewertet werden können. Häufig ist dies die Ablage in einer Datenbank bzw. die Ablage in einer Datei mit fest vorgegebener Syntax (z.B. das in Java geschriebene WEKA-Tool). Die Form der Transformation ist demnach von der Wahl des Auswertungssystems abhängig. Die Daten aus dem Testtool werden mit dem Statistikprogramm „Gnu R“ weiterverarbeitet, welches über eine eigene Programmiersprache verfügt, mittels der die Daten in das entsprechende Format transformiert werden können.

Nach diesen umfangreichen Vorbereitungsphasen kann im nächsten Schritt ein geeignetes Modell gewählt und durchgeführt werden. Hierzu zählt auch die Auswertung und insbesondere die Bewertung und Evaluierung der Ergebnisse. In der klassischen statistischen Mustererkennung ist eine entsprechende Fehlerabschätzung notwendig, um festzustellen, inwieweit die „erkannten Muster“ in der Realität überhaupt gegeben sind. Im vorliegenden Fall kann die Fehlerabschätzung durch die praktische Verifikation der Ergebnisse unterstützt werden. Entweder das IN liefert unter Verwendung der ermittelten Regeln ein brauchbares Ergebnis oder das Netz verhält sich eben nicht wie gewünscht.

8.5 Daten ohne Bereinigung – Teil I

In diesem Abschnitt wird davon ausgegangen, dass die Trainingsdaten, wie in Abschnitt 8.3 beschrieben, aufgezeichnet und für die Auswertung vorverarbeitet – selektiert – wurden. In einer ersten Untersuchung werden zunächst keine weiteren Bereinigungen an den Daten durchgeführt. Da der Gesamtprozess, wie er in Abschnitt 8.4 beschrieben ist, einen „Prüfe-und-fange-zur-Not-wieder-von-vorne-An“-Charakter besitzt, bietet es sich an, zunächst die ersten Ergebnisse zu bewerten und dann den Versuch zu unternehmen, diese zu verbessern. Dieser Schritt wird dann in Abschnitt 8.7 nachgeholt.

Anhand des Studiums der Streudiagramme in Abbildung 8.4 kann man bereits einige Informationen gewinnen.

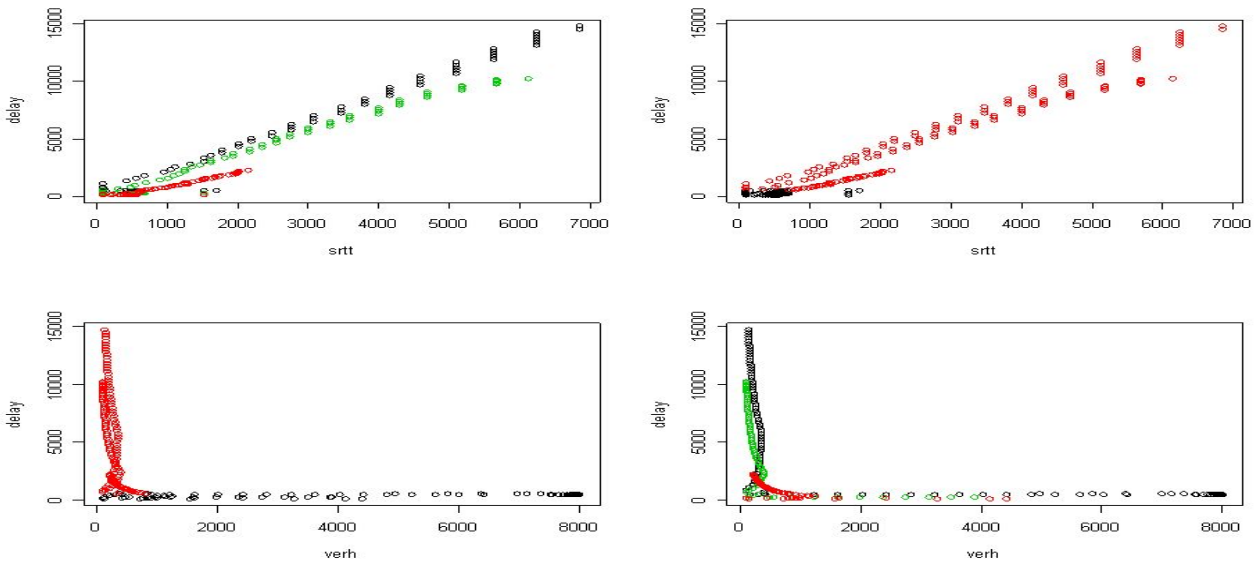


Abbildung 8.4: Streudiagramme der Trainingsdaten mit Markierung der Bandbreiten bzw. der Güteklassen (a) Der Parameter *srtt* im Verhältnis zur Zielgröße *delay* (b) Der Parameter *verh* im Verhältnis zur Zielgröße *delay*

Für einen ersten Überblick wurden in Abbildung 8.4 die Attribute *srtt*, also die von uns ermittelte Smoothed Round Trip Time und der Verhältniswert *verh* gem. Gleichung 8.1 in Abhängigkeit zum aufgetretenen Delay (vgl. Beschreibung der Generierung der Trainingsdaten in Abschnitt 8.3.1) aufgetragen.

Die oberen Teilabbildungen von Abbildung 8.4 stellen den Zusammenhang zwischen der *srtt* und dem *delay* grafisch dar. Man erkennt den *linearen Zusammenhang* dieser beiden Parameter. Dies ist für einige Verfahren ein entscheidendes Kriterium – insbesondere falls die *lineare Regression* verwendet wird. Dass der *delay* bei einer großen Umlaufzeit auch entsprechend groß ist, sollte nicht weiter überraschen. Man könnte annehmen, dass die *srtt* bereits völlig ausreicht, um die Leistungsfähigkeit des Kanals zu beschreiben. Praktische Tests, bei denen lediglich die *srtt* zur Pfadwahl verwendet wurde, haben diese Annahme widerlegt. Ein Problem dabei stellt der enge Zusammenhang der *srtt* zur Bandbreite dar. Man kann dies deutlich der linken Teilgrafik entnehmen, bei der jede im Test verwendete Bandbreite einer Farbe zugeordnet wurde. Die jeweiligen Kurven gehen deutlich auseinander. Somit kann ein Kanal mit kleiner Bandbreite und ein Kanal mit hoher Bandbreite bei prozentual gleicher Auslastung deutliche Unterschiede aufweisen.

Für die Ausgabe in der rechten oberen Teilgrafik von Abbildung 8.4 wurden die Trainingsdaten manuell in zwei Güteklassen *gut und schlecht* aufgeteilt. Für die gewählte

Testumgebung erscheint ein Delay von 500 noch ausreichend zu sein, daher wurde dieser Wert als Schwelle zur Trennung der Klassen herangezogen. Alle schwarz markierten Werte entsprechen dabei den Testdatensätzen der Klasse *gut* und die roten Werte dementsprechend der Klasse *schlecht*. Man kann deutlich erkennen, dass die durchgeführten Tests sehr schnell zu einer Überlastung des Netzes und damit zu langen Übertragungen geführt haben. Dieses Übergewicht an „Negativsätzen“ hat dazu geführt, dass die angewendeten Verfahren nur suboptimale Ergebnisse lieferten (vgl. Abschnitt 9). Im nächsten Unterabschnitt werden die Daten durch die Vorverarbeitung eingeschränkt, was zu deutlich besseren Ergebnissen führte.

Die untere Zeile der Abbildung 8.4 stellt den Zusammenhang zwischen Delay und dem Verhältniswert grafisch dar. Auch hier wurden die Datensätze entsprechend ihrer Bandbreite bzw. ihrer Güteklasse eingefärbt. Man erkennt, dass der Verhältniswert ein sehr „sensibler“ Parameter ist. Wenn er sehr hoch ist, liegt ein niedriger Delay und damit eine geringe Netzauslastung vor. Wird ein bestimmter Wert unterschritten, hat das Netz seine Leistungsgrenze erreicht. An der Bandbreitenabbildung kann man die geforderte „möglichst hohe“ Unabhängigkeit in Bezug auf die Bandbreite erkennen. Nachdem die Daten durch die Vorverarbeitung bearbeitet wurden, sind die Bandbreiten zwar klarer abgrenzbar, aber eine direkte Trennung ist nicht ersichtlich. Durch diese Eigenschaften ist dieser Parameter besonders gut geeignet, um eine möglichst allgemeingültige Funktion zur Beschreibung des IN zu finden.

Um einen besseren Überblick über die Verteilung der Güteklassen innerhalb der Daten zu erhalten, wurden die Daten weiter zerlegt. Für Abbildung 8.5 wurden die Trainingsdaten in sechs Güteklassen {ausreichend, befriedigend, gut, mangelhaft, perfekt, ungenügend} aufgespalten, wobei jede Güteklasse – bis auf ungenügend – einen Delay-Bereich von 300 abdeckt.

8.5.1 Einschub – Boxplot

In diesem Einschub wird kurz die oben verwendete Boxplotdarstellung erläutert. Der Boxplot ergibt einen knappen, aber informativen Überblick über die Testdaten. Folgende mathematischen Kennzahlen werden hierfür verwendet bzw. benötigt:

Definition 8.5.1 (Geordnete Stichprobe) *Sei $X = x_1, \ldots, x_n$ eine Stichprobe, so liegt eine* geordnete Stichprobe *vor, wenn $x_i \leq x_{i+1} \forall x \in X$. Geschrieben $x_{(1)}, \ldots, x_{(n)}$.*

Aus dem Begriff der geordneten Stichprobe lassen sich die Kennzahlen ableiten.

Definition 8.5.2 (Median) *Sei X eine geordnete Stichprobe, dann ist der* Median *definiert als*

$$\tilde{x} = \begin{cases} x_{\frac{n+1}{2}} & \text{, falls} \quad n \equiv 1 \pmod 2 \\ \frac{1}{2}(x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}) & \text{, sonst} \end{cases}$$

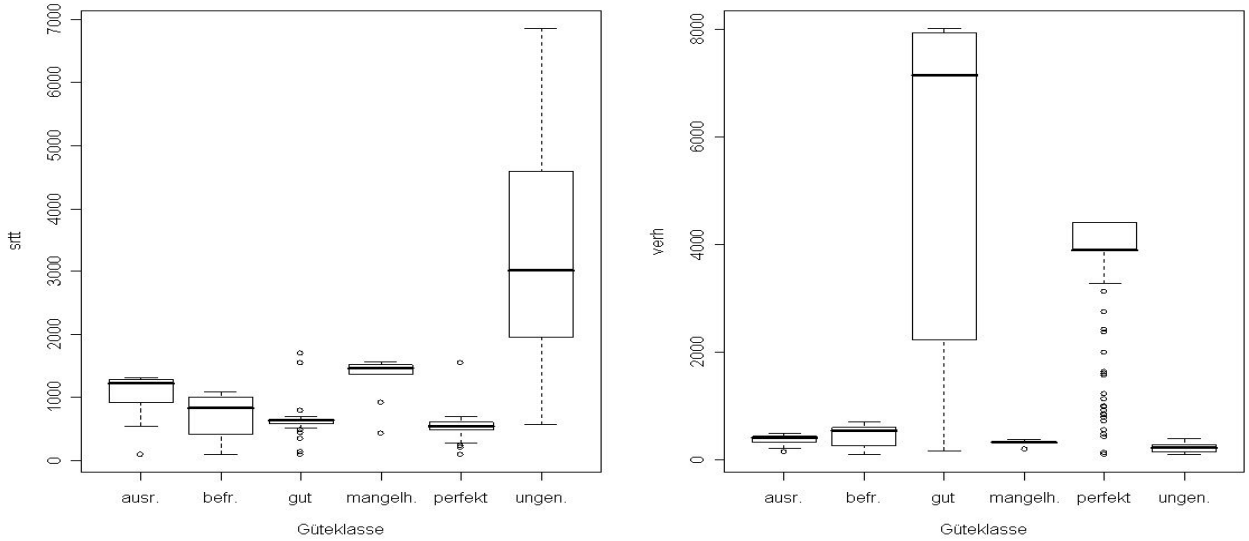


Abbildung 8.5: Güteklasse der Trainingsdaten – Aufteilung in sechs Klassen (a) Darstellung des Parameters *srtt* (b) Darstellung des Parameters *verh*

Der Median gibt somit den Wert x_i, bei dem 50 % der Daten unterhalb und 50 % der Daten oberhalb liegen. Der Median fungiert als Mittelwert, ist aber im Gegensatz zum arithmetischen Mittel resistent gegen Ausreißer und daher für Trainingsdaten, die aufgrund von Messungenauigkeiten zu Ausreißern neigen, besonders gut geeignet.

Verallgemeinert man den Begriff des Medians, so kommt man zum Begriff der p-Quantile. Der Wert der p-Quantile x_p gibt den Wert der Stichprobe X an, bei dem ein Anteil p der Daten unterhalb von x liegen.

Definition 8.5.3 (p-Quantile) *Sei n die Anzahl Elemente einer geordneten Stichprobe X.*

$$x_p = x_{([np+0,5])} \qquad \textit{für } 0 < p < 1$$

Der Wert $x_{0,25}$ *wird als* unteres Quartil*, und* $x_{0,75}$ *wird als oberes Quartil bezeichnet.*

Der Median ist somit der Sonderfall $p = 0.5$ der p-Quantile $\tilde{x} = x_{0,5}$ und wird manchmal auch als *mittleres Quartil* bezeichnet.

Die Kennwerte $\tilde{x}, x_{0,25}, x_{0,75}$ sowie die Extremwerte x_{min} und x_{max} werden als Basis für den Boxplot verwendet. Die zentralen 50 % der Daten werden durch einen Kasten gekennzeichnet, der sich demnach vom unteren Quartil $x_{0,25}$ bis zum oberen Quartil $x_{0,75}$ erstreckt. Weiterhin werden der Median $\tilde{x}$ sowie die Extremwerte x_{min} und x_{max}

durch „Striche“ gekennzeichnet. Ausreißer werden als Punkte bzw. Kreise aufgetragen. Der Boxplot wird gerne zum schnellen Vergleich von Datensätzen angewendet.

8.6 Daten ohne Bereinigung – Teil II

Unter Verwendung der Definitionen von Unterabschnitt 8.5.1 wird jetzt die Beschreibung von Abbildung 8.5 fortgesetzt.

An der Grafik lassen sich einige Probleme der korrekten Klassifizierung eines gegebenen Kanals bereits ablesen.

Problematisch ist beispielsweise die *Spannbreite* zwischen dem Maximal- und dem Minimalwert der Boxen. Es sollen einige Werte exemplarisch betrachtet werden.

Betrachtet man den Verhältniswert, so ergibt sich für die Klasse *gut* ein Minimalwert $x_{min} = 167,8$ und ein Maximalwert von $x_{max} = 8024,7$. Es gibt eine große Anzahl von Daten gerade im unteren Bereich, die in sämtlichen Klassen vorkommen, sodass eine eindeutige Beschreibung über diesen Parameter alleine nicht möglich ist. Auch die als *perfekt* klassifizierten Daten fallen häufiger in diesen Bereich. Sie sind hier zwar nicht signifikant, da es lediglich Ausreißer sind, aber die gewünschte Eindeutigkeit ist nicht gegeben. Vorteilhaft ist dem entgegen, dass die zentralen Daten, die bereits 50 % der Gesamtdaten ausmachen, für die Klassen *gut* und auch *perfekt* eindeutig im oberen Bereich angesiedelt sind, was die Qaurtilen für die Klasse *perfekt* mit $x_{0,25} = 3902,23$ und $x_{0,75} = 4414,92$ bzw. für die Klasse *gut* mit $x_{0,25} = 2225,2$ sowie $x_{0,75} = 7936,8$ belegen. Auch der Median liegt mit $\tilde{x}_{perfekt} = 3902,23$ bzw. $\tilde{x}_{gut} = 7154$ eindeutig im oberen Bereich.

Ein ähnliches Verhalten kann man für den Parameter *srtt* ausmachen. Hier ist die Spannweite von *ungenügend* mit $x_{min} = 569$ und $x_{max} = 6864$ kritisch. Auch hier kommt es zu Überlagerungen der Klassen, sodass eine Trennung nicht eindeutig möglich ist.

Zusammenfassend kann man feststellen, dass

- ein einzelner Parameter nicht ausreicht, um die Klassifikation vorzunehmen
- die Daten in der Vorverarbeitungsphase so aufbereitet werden sollten, dass die Trennung der einzelnen Klassen und somit die Bewertung der Dienstgüte eines Kanals eindeutiger vorzunehmen ist.

Diese Punkte werden in Abschnitt 8.7 nochmals aufgegriffen und an einem Histogramm verdeutlicht.

8.6.1 Anwendung der linearen Diskriminanzanalyse

Die im vorherigen Abschnitt gemachten Aussagen beruhen zum Teil auch auf den praktischen Versuchsreihen (vgl. Abschnitt 9), daher soll hier kurz auf die LDA basierend auf den Original-Trainingsdaten X_{ges} eingegangen werden.

In Abbildung 8.6 ist das Ergebnis der LDA auf dem ursprünglichen Trainingsdatensatz festgehalten.

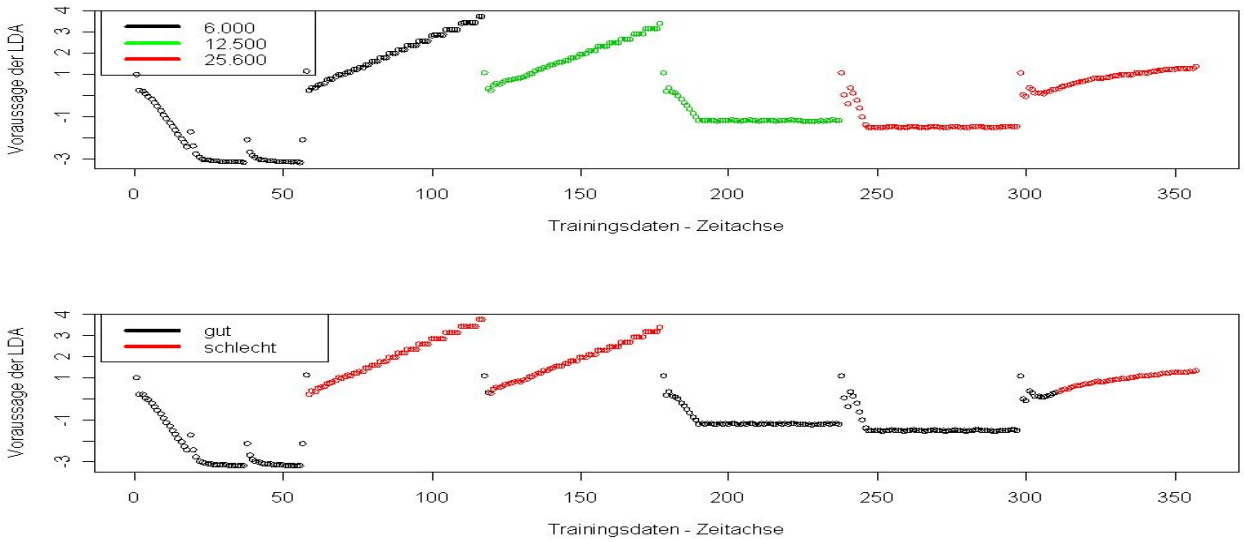


Abbildung 8.6: Auswertung – Prädiktion der Trainingsdaten gegen die LDA (a) Markierung der verwendeten Bandbreiten (b) Markierung der Güteklassen

Als Basis wurde folgendes Modell – für die Formeldarstellung vgl. Abschnitt 7.4 –

$$klasse \sim srtt + verh$$

angenommen, wobei *klasse* der im Abschnitt 8.5 angegebenen Klassifizierung in *gut* und *schlecht* entspricht. Führt man die LDA auf Basis dieses Modells aus, ergeben sich die Koeffizienten (LD1)

```
Coefficients of linear discriminants:
                LD1
srtt  0.0005018529
verh -0.0004812147
```

und damit die *Diskriminanzfunktion*

$$\begin{aligned} y &= \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 \\ &= 0,0005018529 * srtt + (-0,0004812147) * verh \end{aligned} \tag{8.2}$$

Führt man jetzt eine Klassifizierung der Trainingsdaten durch, indem man die Werte der Trainingsdaten in die Diskriminanzfunktion (8.2) einsetzt – also eineSchätzung der Daten gegen sich selbst – erhält man eine Übersicht über die Verteilung der Klassen bzw. eine erste Abschätzung hinsichtlich der Güte der LDA. Gibt man das Ergebnis grafisch aus, so ergibt sich Abbildung 8.6. Die Grafik wurde wieder zwei mal produziert, einmal mit Fokus auf die Bandbreiten und einmal mit Fokus auf die Klasseneinteilung.

Man kann den Experimentverlauf anhand der Grafik noch erkennen, so wurde der Versuch mit drei verschiedenen Bandbreiten hintereinander durchgeführt, wobei die Netzqualität mit zunehmender Zeit immer mehr nachgelassen hat. Die Klassifizierung der LDA ist auf der Ordinate aufgetragen. Ein negativer y-Wert bedeutet eine Zuordnung zur Klasse *gut*, während ein positiver Wert den entsprechenden Datensatz der Klasse *schlecht* zuordnet. An der Grenze zwischen den Klassen stellt man fest, dass nicht alle Datensätze korrekt klassifiziert werden. Dies stellt an sich für das gestellte Problem keine Einschränkung dar, da auf die *numerische Vorhersage* abgestellt wird, also lediglich ein Vergleichswert gebildet werden soll.

8.7 Analyse mit bereinigten Daten

Um die im vorherigen Abschnitt beschriebenen Probleme mit den Daten zu manifestieren, wurden die Histogramme für die Parameter *srtt*, *verh* und der Zielgröße *delay* ausgewertet (vgl. Abbildung 8.7).

Betrachtet man Abbildung 8.7(a) und (b), so stellt man fest, dass die meisten Informationen in den niedrigen Werten, $srtt = \{0 \ldots 4000\}$ und $verh = \{0 \ldots 3000\}$, liegen. Daher wird in einem Vorverarbeitungsschritt die Trainingsdatenmenge reduziert, indem nur die folgende Teilmenge X_{mod} betrachtet wird, die durch Reduktion auf Grundlage der Modellparameter *srtt* und *verh* beruht.

$$X_{mod} = \{x \in X_{ges} \quad | \quad srtt < 4000 \text{ und } verh < 3000\} \tag{8.3}$$

In Abbildung 8.7(c) ist das Histogramm für die Zielgröße *delay* aufgetragen. Wenn man davon ausgeht, dass ein Delay bis zu 500 als *gut* klassifiziert wird und demgegenüber die Spanne von Delay-Werten bis zu 15.000 vorkommen, wird ein großer Bereich nicht relevanter bzw. redundanter Daten ausgewertet. Bei Delay-Werten größer als 5.000 ist an

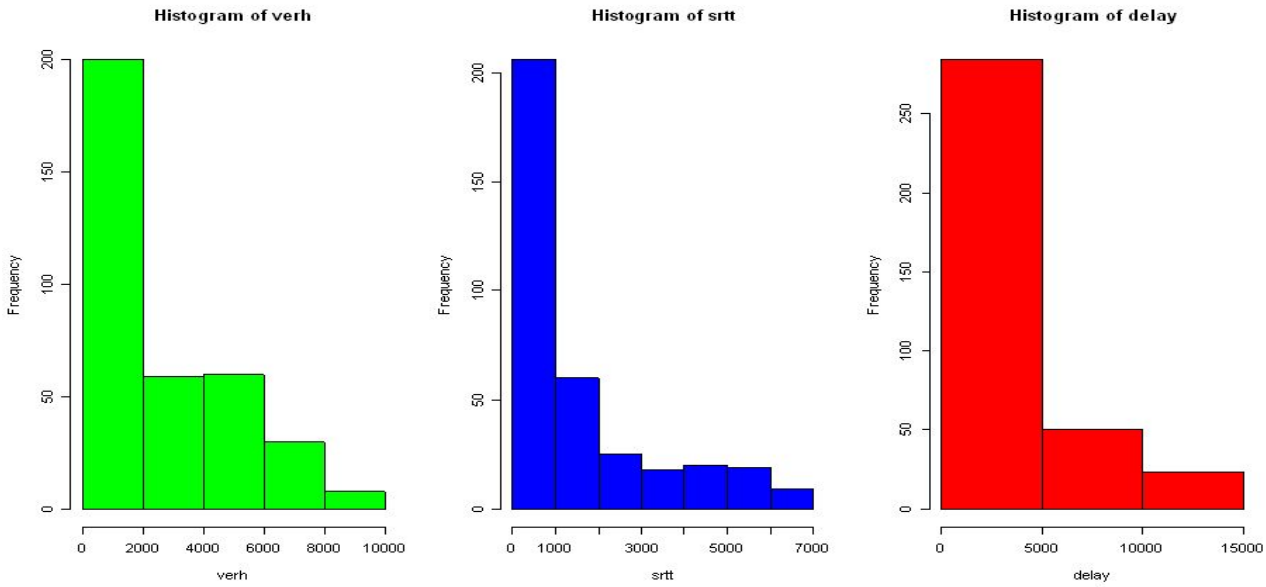


Abbildung 8.7: Histogramme des ursprünglichen Trainingsdatensatzes (a) Histogramm des Parameters *srtt* (b) Histogramm des Parameters *verh* (c) Histogramm der Zielgröße *delay*

sich keine performante Datenübertragung zu erwarten. Daher wird als weiterer Vorverarbeitungsschritt die Teilmenge X_{delay} gebildet, die dieser Feststellung Rechnung trägt.

$$X_{delay} = \{x \in X_{ges} \quad | \quad delay < 5000\} \tag{8.4}$$

In den folgenden Teilabschnitten werden die Ergebnisse der Anwendung der Verfahren auf die bereinigten Daten X_{mod} und X_{delay} untersucht.

8.7.1 Bessere Ergebnisse durch bereinigte Daten?

In praktischen Versuchen gilt zu prüfen, ob durch die Bereinigung der Daten bessere Ergebnisse erzielt werden können. In diesem Abschnitt werden hierfür die Grundlagen gelegt, indem die LDA für jeden Datensatz durchgeführt bzw. zugehörige Regressionsbäume abgeleitet werden.

Im Streudiagramm 8.8 ist das Verhältnis der Attribut-Werte *srtt* und *verh* basierend auf der bereinigten Datenmenge X_{mod} aufgetragen.

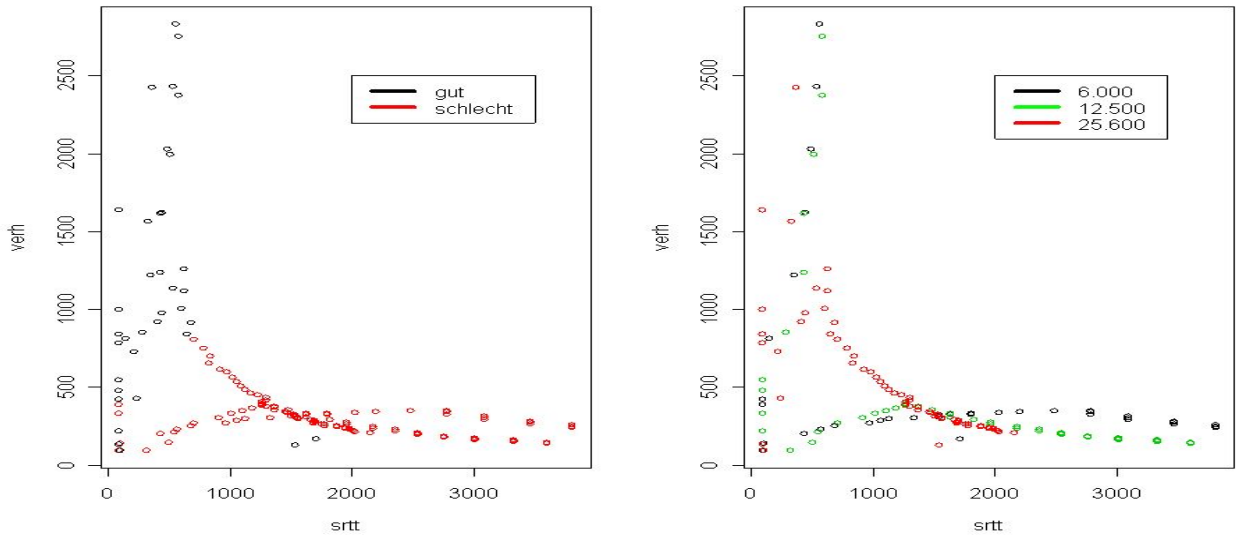


Abbildung 8.8: Streudiagramm der Datenmenge X_{mod} hier srtt und verh (a) Darstellung der Klassenzugehörigkeit (b) Darstellung unter Berücksichtigung der Bandbreiten

Wie zu erwarten war, lässt sich im Zusammenspiel der Parameter die Klassenzugehörigkeit mit hoher Genauigkeit bestimmen. Im Vergleich zu Abbildung 8.4 ist ein deutlich größeres Spektrum an Wertepaaren auszumachen, sodass sich die Bewertung einer bestimmten Netzsituation präziser darstellen und bestimmen lässt.

Aus Teilabbildung 8.8(b) lässt sich auch die „relative“ Unabhängigkeit der Werte von der verwendeten Bandbreite ablesen, sodass die Anwendbarkeit nicht nur für Kanäle mit gleichartiger Struktur – sprich Bandbreite und Kapazität – gegeben ist.

Der Vollständigkeit halber ist das Streudiagramm der Teilmenge X_{delay} in Abbildung 8.9 beigefügt. Die Ausführungen für die Teilmenge X_{mod} haben auch für die Menge X_{delay} Bestand.

8.7.2 Anwendung der linearen Diskriminanzanalyse

Wie sieht die LDA für die bereinigten Datensätze aus? In Abbildung 8.10 ist die Prädiktion auf Basis der LDA zum Datensatz X_{mod} visualisiert. Die grundsätzliche Vorgehensweise zur Generierung und Auswertung, wie sie in Abschnitt 8.6.1 eingeführt wurde, findet hier ebenfalls Anwendung.

In Abbildung 8.10 ist hingegen die LDA für den zweiten Datensatz durchgeführt worden.

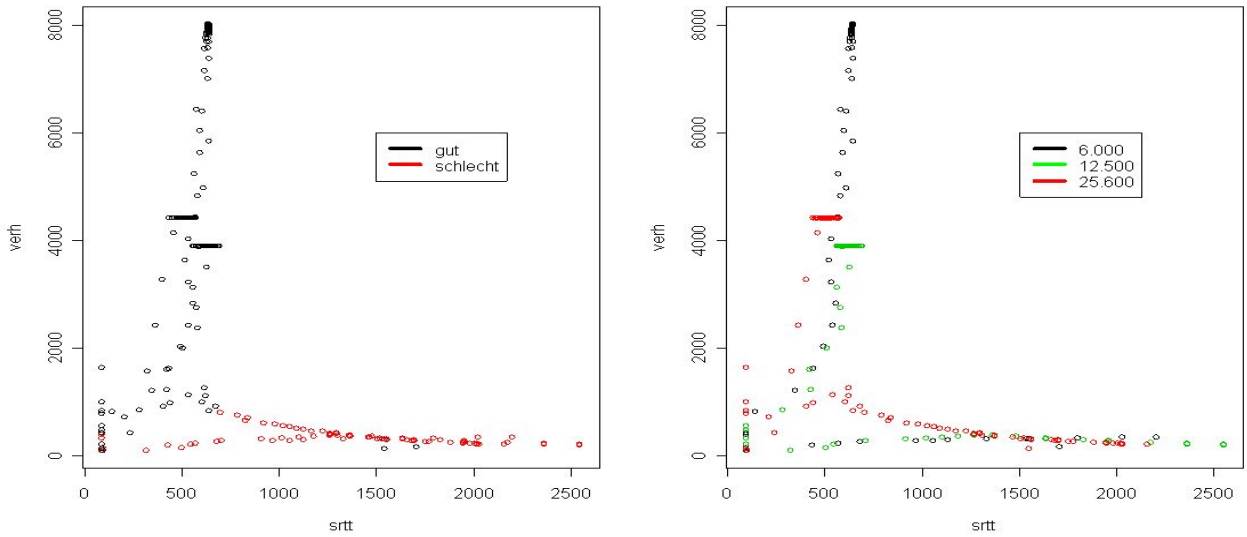


Abbildung 8.9: Streudiagramm der Datenmenge X_{delay} hier srtt und verh (a) Darstellung der Klassenzugehörigkeit (b) Darstellung unter Berücksichtigung der Bandbreiten

Wertmäßig ergeben sich die in Abbildung 8.12 und 8.13 aufgetragenen Auswertungsfunktionen, die in dieser Form direkt in die SCTPLib integriert werden können. Man erkennt, dass durch die Zerlegung der Trainingsmenge in die jeweilige Klasse keine disjunkten Mengen entstehen. Die Bewertung einer Netzsituation ist demnach nicht immer eindeutig.

Die Trennung erfolgt bei beiden Abbildungen bei Null. Alle Werte kleiner Null werden der Klasse *gut* zugeordnet, alle größeren Werte der Klasse *schlecht*. Liegen Werte, die einer Klasse zugeordnet wurden, nicht in dem dafür vorgesehen Bereich, sind diese nicht korrekt zugeordnet worden. Klassisch ist diese Unsicherheit an den Klassengrenzen deutlich zu erkennen. Allerdings gibt es auch größere Ausreißer, beispielsweise bei dem LDA-Wert 1 in Abbildung 8.12 oder beim Wert 2 in Abbildung 8.13.

Die Ausreißer können sicherlich vernachlässigt werden, da das System sich durch die stetige Änderung der Netzparameter auf einem günstigen Niveau einpendelt. Problematischer wiegt die unscharfe Zuordnung im Grenzbereich. Solange die Pfade deutlich voneinander zu unterscheiden sind (vgl. Abschnitt 9.2.1 und Abschnitt 9.2.4) kam diese Problematik nicht zum Tragen, da die Auswertung der einzelnen Pfade gar nicht oder nur sehr selten in diese Grenzbereiche fiel.

Anders sieht es bei der Betrachtung von Pfaden aus, deren Kapazitäten sehr eng an-

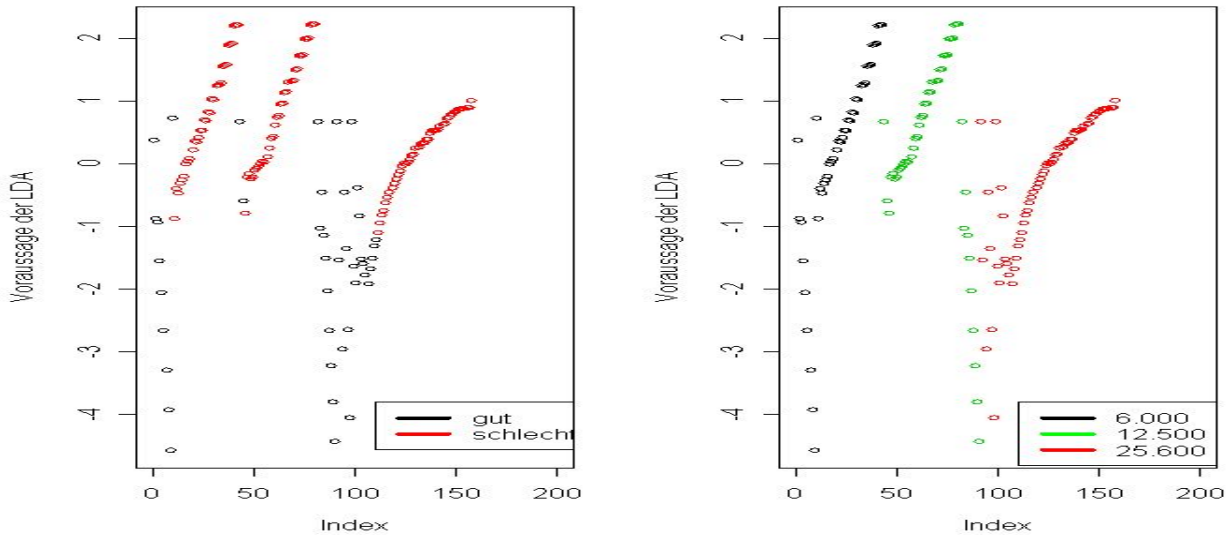


Abbildung 8.10: Prädiktion der Datenmenge X_{mod} mit (a) farblicher Darstellung der Klassenzugehörigkeit und (b) unter Berücksichtigung der verwendeten Bandbreiten

einanderliegen oder im Spezialfall gleich groß sind. Bei Testreihen in diesem speziellen Szenarium (vgl. Abschnitt 9.3) lieferte die Übertragung unter Verwendung des IN nicht die erwarteten sehr guten Ergebnisse. Zwar zeigen die Ergebnisse, dass die LDA praktisch angewendet werden kann, aber auch, dass hier ein Optimierungspotenzial besteht. Bei Verwendung von gleich großen Kanälen fallen die LDA-Werte häufig in den beschriebenen nicht eindeutig klassifizierten Grenzbereich, sodass es zu Fehlentscheidungen bei der Pfadwahl kommen kann.

Abschließend werden die konkreten Diskriminanzfunktionen angegeben, die sich anhand der Auswertung der Trainingsdaten ergeben und in das IN Eingang gefunden haben.

Ergebnis 8.7.1 (Ermittelte Diskriminanzfunktionen) *Die LDA liefert für die Datenmenge X_{mod} folgende Koeffizienten:*

```
Coefficients of linear discriminants:
              LD1
srtt  0.000766357
verh -0.001661036
```

und damit die Diskriminanzfunktion

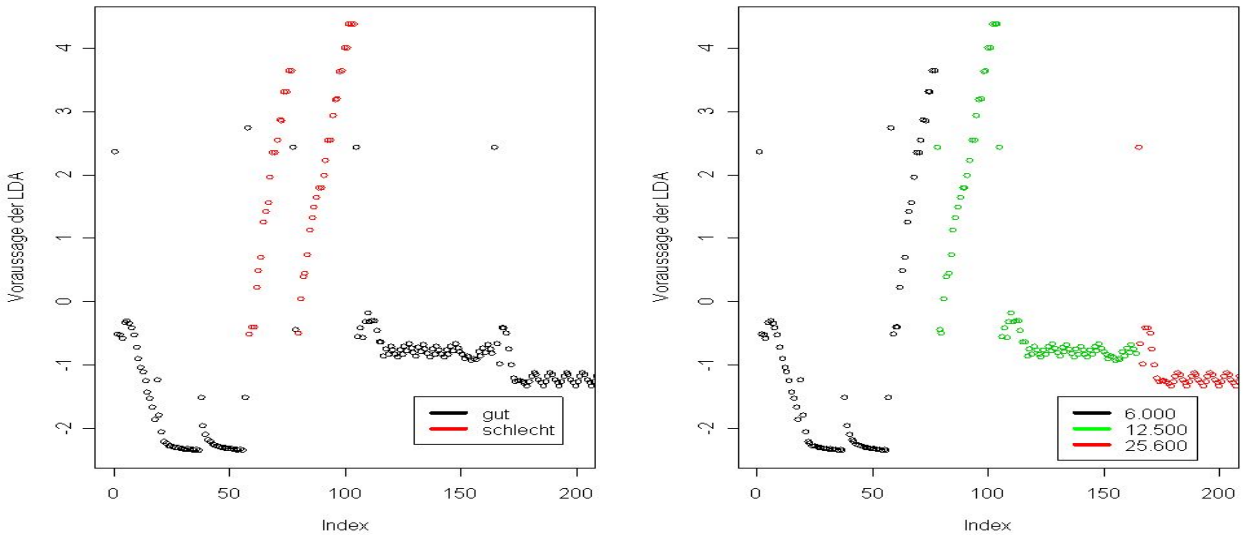


Abbildung 8.11: Prädiktion der Datenmenge X_{delay} mit (a) farblicher Darstellung der Klassenzugehörigkeit und (b) unter Berücksichtigung der verwendeten Bandbreiten

$$\begin{aligned} y_{x_{mod}} &= \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 \\ &= 0,000766357 \cdot srtt + (-0,001661036) \cdot verh \end{aligned} \tag{8.5}$$

Für die Datenmenge X_{delay} wurden die folgenden Koeffizienten ermittelt:

```
Coefficients of linear discriminants:
               LD1
srtt  0.0019584988
verh -0.0003805504
```

Es folgt die Diskriminanzfunktion

$$\begin{aligned} y_{x_{delay}} &= \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 \\ &= 0,0019584988 \cdot srtt + (-0,0003805504) \cdot verh \end{aligned} \tag{8.6}$$

8.7.3 Anwendung eines Regressionsbaums

Als weiteres Verfahren kommen Entscheidungs- und Regressionsbäume, wie sie in Abschnitt 7.5 eingeführt wurden, zur Anwendung.

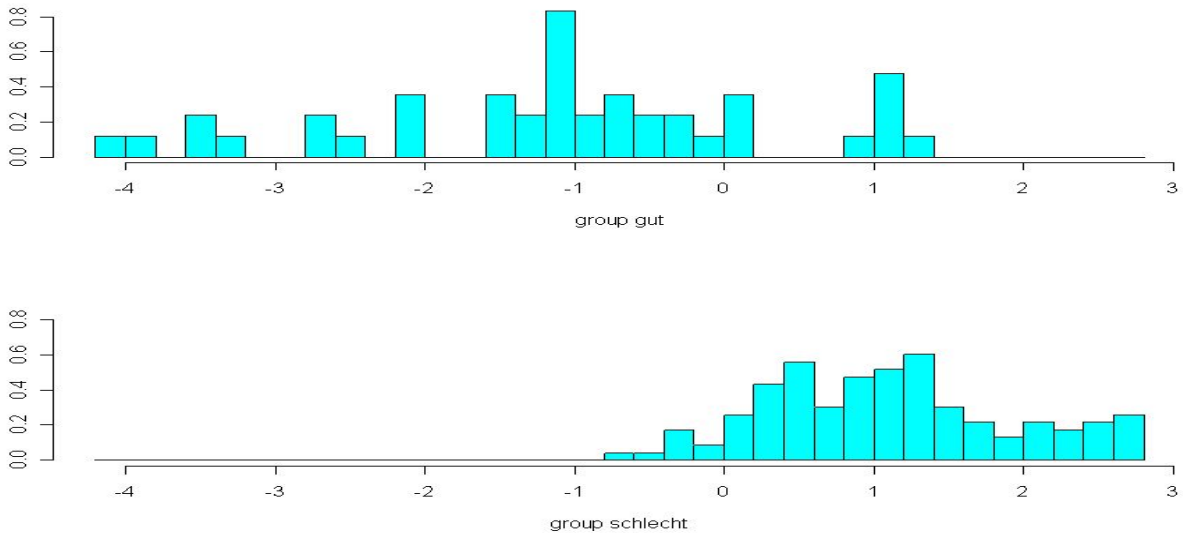


Abbildung 8.12: Verteilung der guten und schlechten Pfadeigenschaften – Anwendung auf die Datenmenge X_{mod} (a) Alle Trainingsdatensätze der Klasse *gut* (b) die entsprechenden Datensätze der Klasse *schlecht*

Der Ablauf des Algorithmus zerfällt in folgende Schritte. Als Erstes ist ein *Baum-Modell* an die vorgegebenen Daten anzupassen. Im vorliegenden Szenario müssen die Trainingsdaten an das Baum-Modell angepasst werden. Zur Anwendung kommt ein sogenannter *Rekursiver-Partitionierungs-Algorithmus*[16]. Jeder Baumknoten führt eine Zerlegung der Daten in zwei Gruppen durch, wobei ein Parameter X und ein Wert t gewählt wird, die ein vorgegebenes Kriterium minimieren. Der Knoten kann dann als Trennregel $L = \{X < t\}$ für den linken Teilbaum und $R = \{X \geq t\}$ für den rechten Teilbaum formuliert werden.

Wendet man die R-Funktion *rpart* auf unsere Datensätze X_{mod} sowie X_{delay} an, ergeben sich die in Abbildung 8.14 abgebildeten Baumstrukturen. Der zugehörige Aufruf aus R gibt auch das verwendete Modell an:

```
Call:
rpart(formula = delay ~ verh + srtt, method = "anova")
```

[16]Recursive Partitioning Algorithm

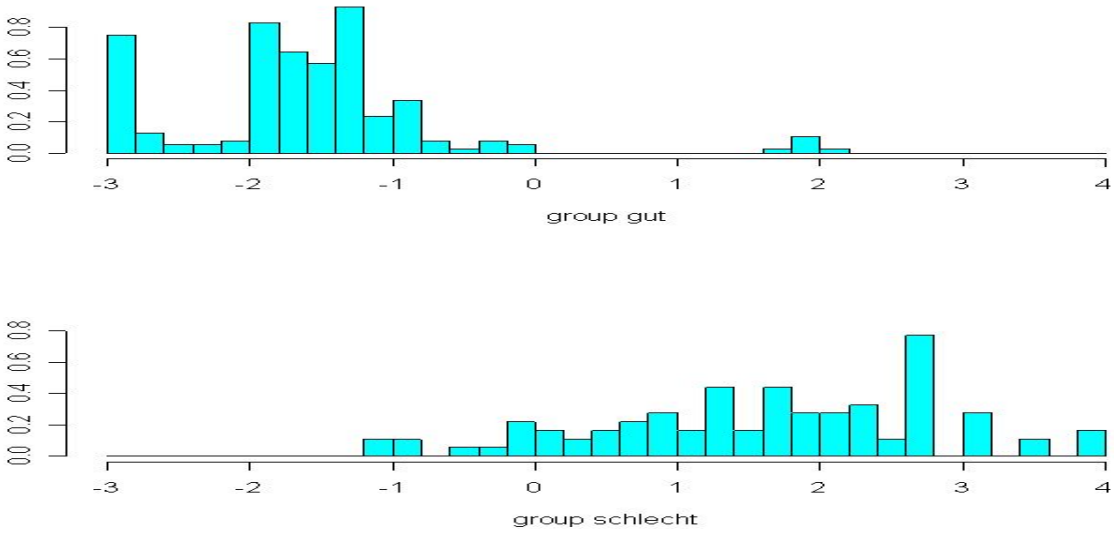


Abbildung 8.13: Verteilung der guten und schlechten Pfadeigenschaften – Anwendung auf die Datenmenge X_{delay}(a) Alle Trainingsdatensätze der Klasse *gut* (b) die entsprechenden Datensätze der Klasse *schlecht*

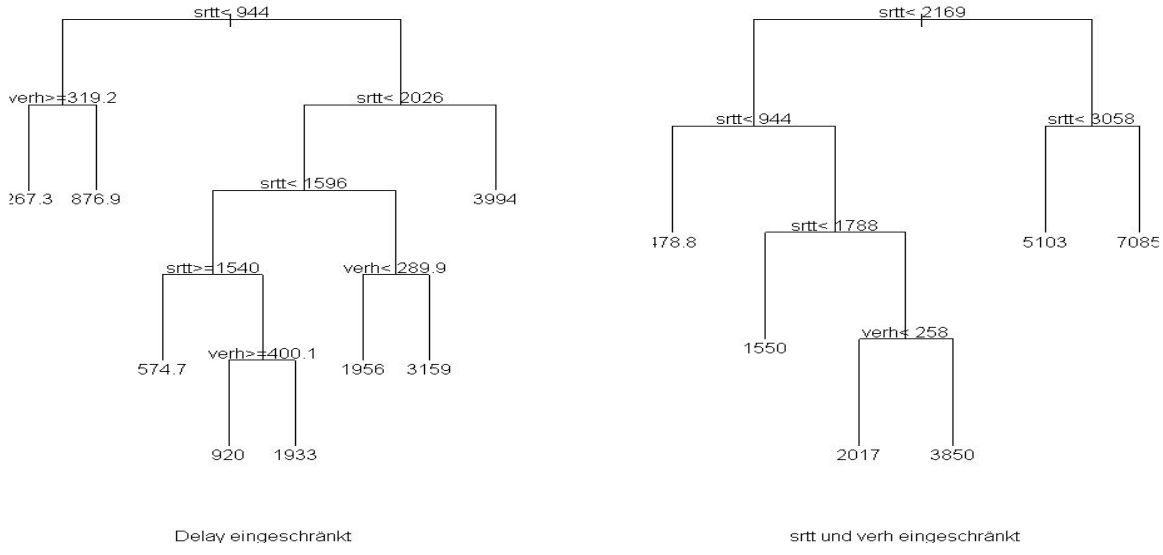


Abbildung 8.14: Regressionsbäume der Trainingsdaten (a) Verwendung der Datenmenge X_{mod} (b) Verwendung der Datenmenge X_{delay}

9 Auswertung der Daten

Nachdem die Problemstellung formalisiert und die Verfahren sowie das Modell ausgewählt wurden, müssen die Ergebnisse erprobt werden. Hierfür steht die bereits für die Generierung der Trainingsdaten genutzte Testplattform zur Verfügung.

Grundsätzlich kann beim Test an sehr vielen Parametern und Einstellungen geschraubt werden, die alle die Ergebnisse beeinflussen. Es ist somit sinnvoll, feste Eckwerte für diese Parameter festzulegen und alle Testreihen unter denselben Rahmenbedingungen ablaufen zu lassen.

Für die folgenden Testreihen werden folgende Einstellungen vorausgesetzt:

- Die grundsätzlich konfigurierbaren SCTP-Parameter werden auf die Standard-Werte gesetzt.
- Die Chunkgröße wird fest vorgegeben. Sämtliche eingespeisten Chunks werden auf 250 Bytes festgesetzt.
- Die Kanalkapazitäten bzw. Bandbereiten werden unter Verwendung von NistNet emuliert. Diese werden vor dem Test fest eingestellt und während des Tests nicht variiert.
- Ebenso wird bei NistNet ein allgemeiner Delay für die Verbindung festgesetzt.

Basierend auf den in der vorliegenden Arbeit gewonnenen Ergebnissen ist es interessant zu erfahren, welchen Einfluss diese zusätzlichen Parameter auf das Verhalten und die Leistungsfähigkeit des IN haben. Insbesondere das Verhalten bei anderen Chunkgrößen, das zu einem anderen Bundling der Pakete führt, ist von Interesse. Weiterhin ist die Frage zu klären, welchen Einfluss ein geografisch vorgegebener Delay auf das Verfahren hat. Aufgrund der möglichst allgemeinen Grundstruktur des IN sollte es Änderungen der hier beschriebenen Parameter zu keinen negativen Auswirkungen auf die Leistungsfähigkeit des IN kommen.

Während der Testphase wurden die im vorherigen Abschnitt aus den Trainingsdaten abgeleiteten Regeln und Gleichungen als Basis für das IN verwendet. Die Ergebnisse der verschiedenen Testreihen weichen stark voneinander ab. Die besten und stabilsten Ergebnisse wurden durch Anwendung der LDA auf der Datenmenge X_{mod} erzielt. Daher beziehen sich die folgenden Ausführungen auf eine IN-Implementierung auf Basis der zugehörigen LDA, wie sie in dem Ergebnis 8.7.1 aufgeführt ist.

9.1 Datenüberholungen und Fast-Retransmission

In der Einleitung wurden die Probleme des MP-SCTP als

- Unnötige Fast-Retransmissions
- Verringerung des Berechnungszyklus für das Sendefenster (cwnd)
- Anstieg des Traffic durch vermehrtes Senden von Bestätigungen (Acknowledgements)

festgemacht, wobei die Problematik der Neuübertragungen (Problem 1) am schwersten wiegt.

Die schnelle Neuübertragung (Fast-Retransmission) ist Bestandteil der zuverlässigen Übertragung mittels SCTP. Eine Übersicht über die Mechanismen der zuverlässigen Übertragung kann im Abschnitt 6.3 nachgelesen werden.

Eine Möglichkeit dem Problem zu begegnen besteht darin, direkt in den SCTP-Kernel einzugreifen und über abgewandelte Gap-Reports den Neuversand von Chunks zu kontrollieren. Ein entsprechender Algorithmus wurde u.a. von Iyengar in [Iyengar et al. 2006] erfolgreich eingeführt.

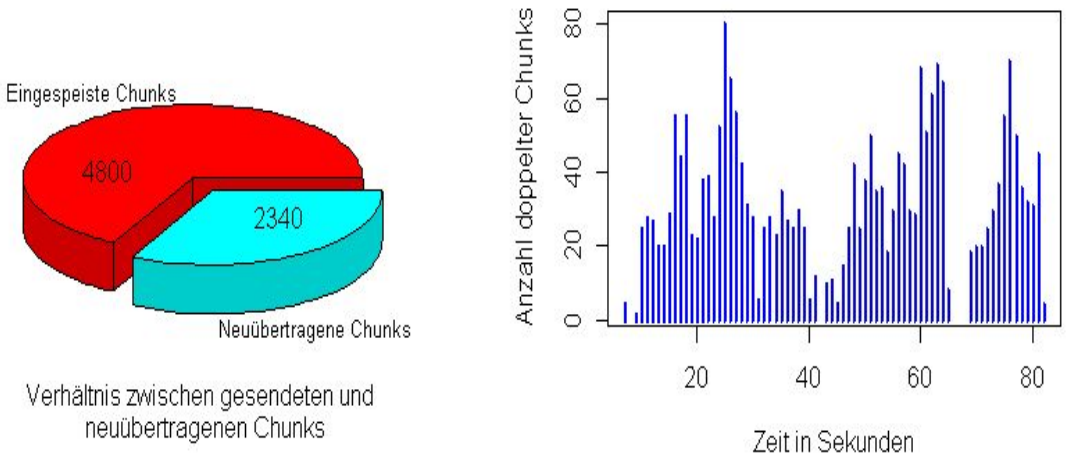


Abbildung 9.1: Doppelte Datenchunks durch schnelle Neuübertragung (a) Durchschnittliche zeitliche Verzögerung von neu übertragenen Datenchunks (b) Anzahl und Zeitpunkt des Eintreffens von neu übertragenen Datenchunks

Für die folgenden Testreihen kann aber auch auf die Erweiterung von SCTP zur partiell zuverlässigen Übertragung, dem sogenannten PR-SCTP[1], wie er in Abschnitt 10.2 beschrieben wurde, zurückgegriffen werden. Der Effekt ist vergleichbar mit dem o.a. Algorithmus, da auf der Teststellung ein zuverlässiger Versand gewährleistet werden kann. In einer konkreten Realisierung sollten sämtliche Lösungsansätze von Iyengar bzw. alternative Ansätze gleicher Intention eingesetzt werden.

Sämtliche Testreihen der folgenden Abschnitte wurden daher unter Verwendung von PR-SCTP durchgeführt. In Abbildung 9.13(b) im Abschnitt 9.3 ist eine Übertragung von 80 Datenchunks pro Sekunde auf zwei gleich große Pfade abgebildet. Die Daten waren nach ca. 60 Sekunden vollständig übertragen. Im Vergleich dazu benötigt eine Übertragung im zuverlässigen Modus von SCTP über 80 Sekunden mit einem erheblichen zusätzlichen Traffic durch doppelte Chunks. Abbildung 9.1 vermittelt einen Eindruck von den negativen Auswirkungen der unnötigen schnellen Neuübertragungen.

Die Teilabbildung 9.1(b) stellt die Anzahl sämtlicher doppelt übertragener Chunks zum Zeitpunkt des Eintreffens beim Empfänger dar. Insgesamt sind 2.340 Chunks doppelt versendet worden. Da ursprünglich $60 \cdot 80 = 4.800$ Datenchunks eingespeist wurden (vgl. Teilabbildung 9.1(a)), wurde somit fast jeder zweite Chunk doppelt übertragen. Unter diesen Voraussetzungen ist die parallele Nutzung der Pfade nicht möglich.

Falls man das Problem der unnötigen Neuübertragungen im Griff hat, können die weiteren Probleme zumindest für die hier vorgenommenen Testreihen vernachlässigt werden. Ein geringfügig größerer Traffic durch die Bestätigungs-Chunks fällt nicht wirklich ins Gewicht. Durch die Verwendung von PR-SCTP wird die Berechnung der Fenstergröße des Sendefensters (cwnd) nicht tangiert. Somit kann PR-SCTP zur Emulation der in [Iyengar et al. 2006] empfohlenen Algorithmen verwendet werden.

9.2 Besondere Betrachtung des IN bei Kanälen unterschiedlicher Kapazität

9.2.1 Betrachtung bei geringem Abstand

In diesem Unterabschnitt werden zwei Kanäle unterschiedlicher Kapazität für die Übertragung der Testdaten verwendet, die gerade in heterogenen Netzen vorkommen. In einer ersten Versuchsreihe wurden zwei Kanäle gewählt, die zwar unterschiedliche Kapazität aufweisen, aber von ihrer Grundstruktur her nicht sehr weit auseinander liegen. Die Testreihen wurden unter den in der Einleitung zu diesem Abschnitt (vgl. 9) gemachten Voraussetzungen durchgeführt.

Hierfür wurden die Kanäle über Nistnet auf 12.500 Bytes/s und 6000 Bytes/s beschränkt. Der Delay wurde auf $45ms$ fest eingestellt und für alle Versuchsreihen beibehalten. Für

[1] SCTP Partial Reliability Extension

die Übertragung wurden die im Vorfeld festgelegten „Max-Werte“ verwendet. Diese Werte sind so eingestellt, dass eine Übertragung unter guten Voraussetzungen zeitnah erfolgen kann. Da das Testtool die zu versendende Nutzlast in Form von Chunkgrößen angibt, kann die tatsächliche Netzlast abweichen. Dies resultiert aus der Bundling-Funktionalität von SCTP, die darauf bedacht ist, die Chunks möglichst effektiv auf die einzelnen SCTP-Pakete zu verteilen. Je nach Grad des Bundlings kommen entsprechende Daten für die Paketheader und Kontrollstrukturen hinzu. Als Datenlast wurden die Daten eine Minute lang in Sekundenabständen versendet. Speziell für diese Versuchsreihe wurde der Maximalwert auf 20 Chunks für den kleineren Pfad bzw. 40 Chunks für den größeren Kanal festgelegt, die mit einer Chunkgröße von 250 Bytes Nutzlast pro Sekunde in das Netz eingespeist wurden.

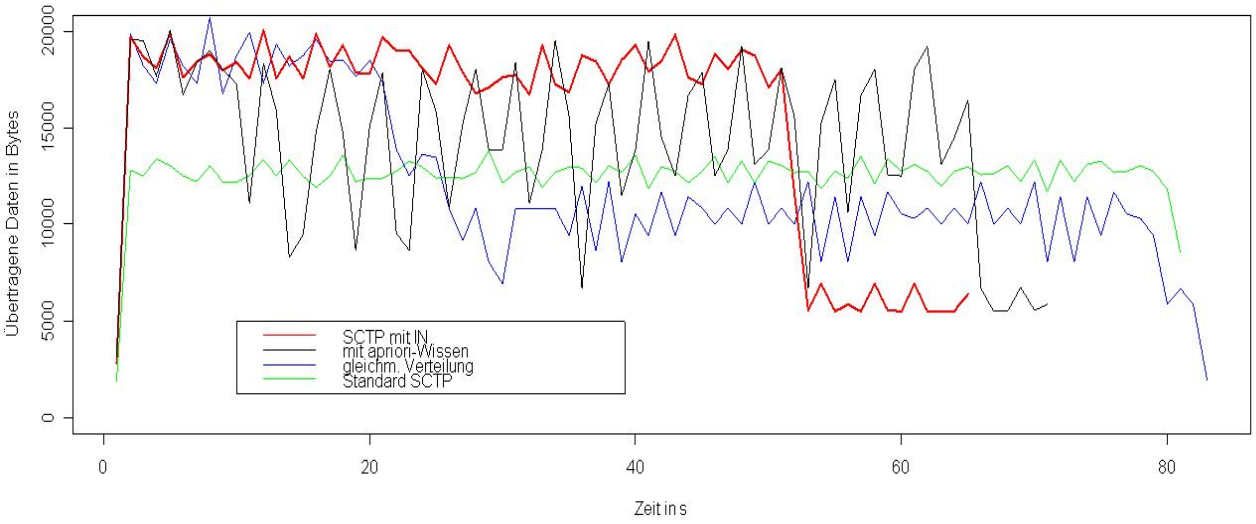


Abbildung 9.2: Vergleich der einzelnen Versuchsreihen – Gesamtübertragung auf beiden Kanälen

Der Gesamtdurchsatz von repräsentativen Übertragungen

Für Abbildung 9.2 wurden für die unterschiedlichen Versuchsszenarien repräsentative Übertragungen herausgegriffen. Inwieweit sich diese Ergebnisse verallgemeinern bzw. reproduzieren lassen, wird am Ende dieses Unterabschnitts Thema sein.

Die Messungen wurden mit Wireshark durchgeführt. Die generierten Capture-Dateien wurden so nachbearbeitet, dass lediglich der reine Datentransfer, sprich: SCTP-Pakete mit Daten-Chunks, bei der Ausgabe berücksichtigt werden. Die Abbildung 9.2 zeigt le-

diglich die Gesamtübertragung, in Abbildung 9.3 werden dagegen die Auslastung der einzelnen Pfade sowie ihre Stabilität berücksichtigt.

Als Erstes soll Abbildung 9.2 näher erläutert werden. Die grün eingezeichnete Kurve beschreibt den *Referenzverlauf* der Datenübertragung mit Standard-SCTP. Hierbei wurde der größere Pfad mit 12500 $Bytes/s$ als Primärpfad gewählt, der zweite Pfad wurde lediglich zur Ausfallsicherheit im klassischen SCTP-Multihoming-Szenario hinzugefügt. Auf diesem Pfad werden die Daten konstant auf sehr hoher Auslastung übertragen. Da die Kapazität des Primärpfades nicht ausreichend ist, um die eingespeiste Datenmenge in Echtzeit zu übertragen, läuft die Übertragung entsprechend nach.

Um die volle Kapazität beider Kanäle zu nutzen, wurde als Erstes ein *Loadsharing* verwendet, welches die Daten gleichmäßig auf beide Kanäle verteilt. Da die Kanäle unterschiedliche Kapazitäten aufweisen, ist hier mit einem nicht optimalen Ergebnis zu rechnen. Die Messergebnisse bestätigen dies. In der Grafik ist das Loadsharing als blaue Kurve vermerkt. Durch die gleichmäßige Verteilung werden beide Kanäle von Anfang an mit Daten versorgt, sodass die Gesamtübertragung auf einem hohen Niveau beginnt. Nach kurzer Zeit bricht die Übertragung aufgrund der ungünstigen Auslastung der Kanäle ein und benötigt ein Vielfaches an Zeit für die Übertragung. Das vorher vermutete ungünstige Übertragungsverhalten wurde durch die Testläufe bestätigt.

Bessere Ergebnisse sind hingegen durch Anwendung von a-priori Informationen zu erwarten. Sind die aktuellen Kapazitäten der Kanäle bekannt, so können die Daten gezielt auf beiden Kanälen platziert werden. Das Testergebnis dieser Methode ist in der Grafik in schwarz aufgetragen. Hierfür wurden die Daten in einem Verhältnis von 1 : 2 auf den kleineren bzw. größeren Kanal verteilt. Die Übertragung kann auf einem hohen Niveau gehalten werden. In der Zusammenstellung in Form von signifikanten Datensätzen (vergl. Abschnitt 9.2.2) konnte dieses Ergebnis bestätigt werden. Die Daten werden relativ optimal so auf beide Kanäle verteilt, dass eine Echtzeitübertragung möglich ist. Diese Übertragung dient als Referenzübertragung für die folgende Betrachtung der Ergebnisse, die mit Hilfe des IN generiert wurden.

Die rote Funktion in Abbildung 9.2 ist unter Anwendung des IN entstanden. In den Testreihen hat sich die LDA auf Basis der Trainingsdaten X_{mod} als am effektivsten erwiesen, sodass im Folgenden diese Ergebnisse für den Vergleich der einzelnen Pfadwahlen verwendet werden. Die Übertragung unter Verwendung des IN weist eine gute Stabilität auf. Lediglich am Ende der Übertragung ist ein Abfall der Leistung zu beobachten. Um zu untersuchen, wie sich dieser Abfall der Leistung auf längere Übertragungen auswirkt, wurden auch längere Testreihen erstellt. Als Ergebnis konnte festgehalten werden, dass der Abfall der Leistung immer am Ende einer Gesamtübertragung vorkam. Dies lässt sich auf die Art der verwendeten Pfade zurückführen.

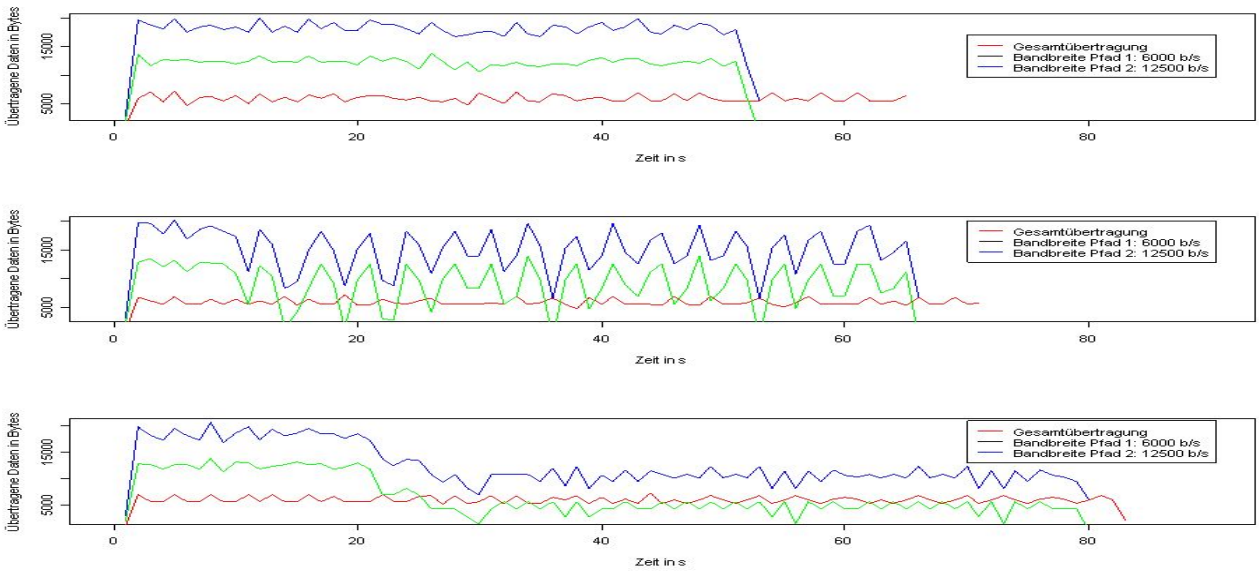


Abbildung 9.3: Verteilung der übertragenen Daten auf die einzelnen Kanäle (a) Pfadverteilung über IN (b) Pfadverteilung mittels a-priori Informationen (c) Verwendung von Loadsharing zur gleichmäßigen Verteilung der Daten auf beiden Kanälen

Auslastung der einzelnen Pfade

Betrachtet man hierfür die Übertragung auf den einzelnen Pfaden, wird dies deutlich. In Abbildung 9.3(a) ist die tatsächlich vorgenommene Verteilung der Datenchunks auf die beiden Kanäle abgebildet. Während der Kanal mit der größeren Kapazität sämtliche Daten, die auf ihm eingespeist wurden, bereits übertragen hat, ist der kleinere Kanal mit zu vielen Chunks bestückt worden, sodass die Übertragung dieser „Überlast“ noch abgearbeitet werden muss.

Die Pfadwahl über Loadsharing ist in Abbildung 9.3(c) aufgetragen. Hier ist deutlich der Abfall der Leistung nach einer sehr stabilen Auslastung beider Kanäle in der Anfangsphase zu sehen. Auffällig ist, dass nicht der kleinere Kanal, der ja mit einer Vielzahl von Daten gefüttert wird, zusammenbricht, sondern der größere Kanal, auf dem grundsätzlich ausreichende Ressourcen zur Verfügung stehen.

Den Flaschenhals stellt hier die *Sendqueue* dar, die nur pro Assoziation gepflegt wird und somit für beide Kanäle zur Zwischenspeicherung der Daten gemeinsam verwendet wird. Werden die Daten für den kleineren Kanal nicht zeitnah abgearbeitet, sendet SCTP die ausstehenden Daten nicht auf dem größeren Kanal, obwohl dieser durch die direkte Adressierung dafür vorgesehen ist. Dieses Negativbeispiel zeigt, dass bei der Pfadwahl

nicht nur auf die aktuelle Leistungsfähigkeit einzelner Pfade abgestellt werden sollte, sondern die gesamte Assoziation berücksichtigt werden muss. Damit der Sendqueue-Algorithmus optimal ablaufen kann, darf kein Kanal ausfallen bzw. durch Überlastung vollständig lahm gelegt werden.

In Abbildung 9.3(b) ist eine Beispielsitzung für die Verteilung unter Ausnutzung der a-priori Information abgebildet. Bei dieser Art der Einspeisung fällt auf, dass der kleinere Kanal stabil ausgelastet ist und der größere Kanal zu Schwankungen neigt. Dies lässt sich ebenfalls mit der zu Abbildung 9.3(c) getroffenen Aussage erklären. Hier kommt es kurzfristig zu einer „Stausituation“, da der kleinere Kanal kurzfristig überlastet ist. Da aber keine größeren Datenmengen auf dem kleineren Kanal nachgeschoben werden, werden die nachfolgenden Daten in der Sendqueue nach Abarbeitung des kurzfristigen Staus unverzüglich auf den dafür vorgesehenen größeren Kanal eingespeist, der dann auch sofort eine große Menge an Daten abtransportieren kann. Der kleinere Kanal ist dabei stabil ausgelastet, während der größere zu periodischen Schwankungen neigt. Dies liegt auch daran, dass die zu übertragenden Daten leicht über den möglichen Datenvolumen liegen. Werden weniger Daten geschickt verteilt eingespeist, sollte auch der größere Kanal sauber übertragen. Betrachtet man den *signifikanten Datensatz* in Abbildung 9.6 i.V.m. Abschnitt 9.2.2, kann man erkennen, dass die Schwankungen ausgeglichen werden und grundsätzlich eine sehr gute Datenübertragung möglich ist.

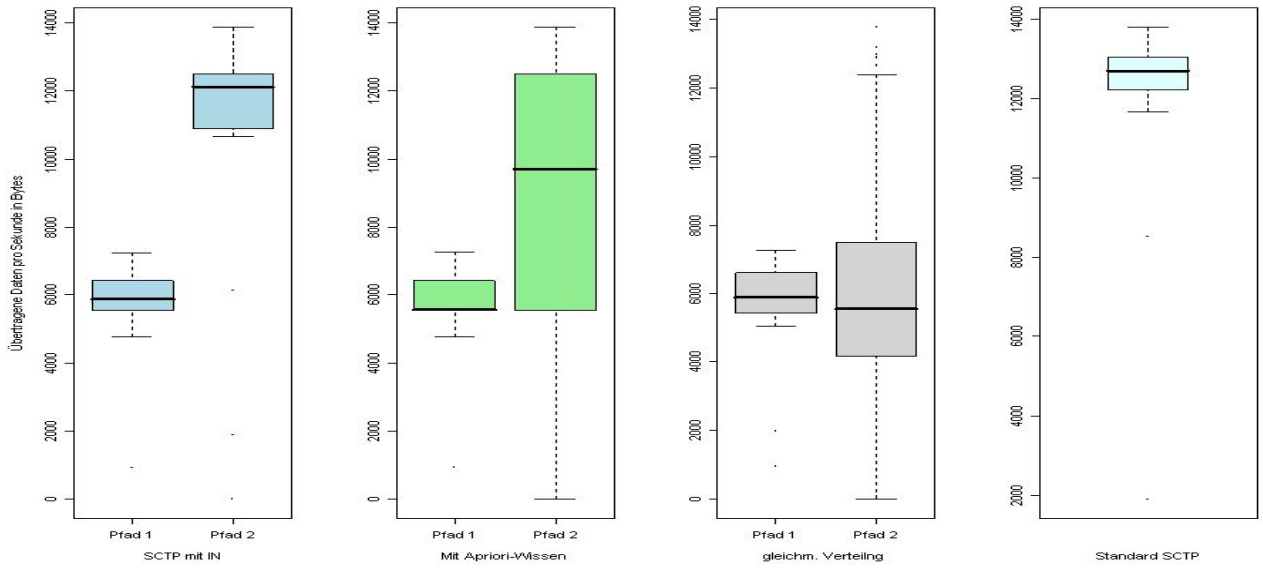


Abbildung 9.4: Vergleich der wesentlichen Kennzahlen der verschiedenen Beispielübertragungen (a) SCTP mit IN (b) Pfadwahl durch a-priori Wissen (c) Pfadwahl durch Loadsharing (d) Standard-SCTP

Kennwerte der Testreihen

Bevor die bereits erwähnte statistische Auswertung eingehender behandelt wird, wird die Stabilität der Übertragung untersucht. Hierzu sind in Abbildung 9.4 die wesentlichen Kennzahlen der o.a. Beispielsitzungen grafisch zusammengestellt. Die hier verwendete Boxplotsystematik entspricht der Beschreibung in Abschnitt 8.5.1.

In Abbildung 9.4(d) sind die Kennwerte der Standard-SCTP-Übertragung aufgetragen, welche die beste und stabilste Übertragung darstellt. Dies kann man direkt an den Kennwerten ablesen. Der Median liegt sehr hoch, bei 12.527 $Bytes/s$. Dadurch, dass der Plot sehr schmal ist, liegen die obere und untere Quartile mit 12.224 und 13.048 Bytes/s eng beieinander, sodass von einer regelmäßigen konstanten Übertragung der Daten ausgegangen werden kann. Auch die Maximal- und Minimalwerte liegen dicht am Median, sodass hier keine Abweichungen aufgetreten sind. Einzelne Ausreißer treten gar nicht bzw. sehr selten auf. Die Übertragung ist als sehr gut anzusehen. Allerdings wird beim Standard-SCTP auch nur ein Pfad genutzt, sodass die mögliche Nutzlast auf diesen einen Kanal beschränkt bleibt.

An die sehr guten Übertragungseigenschaften der Standard-SCTP-Übertragung reichen die Multipfad-Übertragungen nicht heran. Am schlechtesten schneidet hierbei das Loadsharing (vgl. Abbildung 9.4(c)) ab, da hier, wie bereits beschrieben, der größere Pfad nicht ausreichend genutzt werden kann. Dies kann an den statistischen Werten nochmals sehr anschaulich dargelegt werden. Die Median-Werte der beiden Kanäle pendeln sich bei ähnlichen Werten von 5.892 und 5.552 Bytes/s ein, was bedeutet, dass auf dem größeren Kanal pro Zeiteinheit sogar weniger Daten übertragen werden als auf dem kleineren Kanal. Der kleinere Kanal ist seinerseits sehr gut ausgelastet und weist über die Quartilen einen sauberen Verlauf der Übertragung auf. Die Minimal- bzw. Maximalwerte des größeren Kanals weichen stark voneinander ab, auch sind sehr viele Ausreißer sichtbar. Diese Konstellation ist bei solch unterschiedlichen Kanälen nicht geeignet. Die Übertragung über einen Standard-SCTP-Kanal erweist sich als effizienter als die Loadsharing-Methode.

Grundsätzlich gut geeignet ist die Übertragung mit a-priori Informationen, wie sie in Abbildung 9.4(b) dargestellt ist. Im Vergleich zum Loadsharing liegt der Median des größeren Kanals deutlich über dem des kleineren Pfades. Die Quartile liegen zwar weiter auseinander, liegen aber im oberen Leistungsbereich und sind somit Garant für eine sehr gute Datenübertragung. Mit 13.880 Bytes/s wird ein sehr hoher Maximalwert bei der Übertragung erreicht. Das einzige Problem besteht in der Verfügbarkeit der a-priori Informationen. Falls die Pfadeigenschaften zur Verfügung stehen, kann die Verteilung direkt erfolgen. Dies ist normalerweise in heterogenen Netzen nicht gegeben.

Die Pfadvermittlung über das IN bzw. die zugehörige LDA ist in Abbildung 9.4(a) aufgetragen. Ähnlich wie bei der a-priori Verteilung werden sehr gute Übertragungseigenschaften erreicht. Der kleinere Pfad wird konstant und vollständig ausgenutzt. Die

Stabilität lässt sich hier ebenfalls anhand der günstigen Werte für die Quartile bzw. den Median herleiten. Aber auch der größere Pfad wird sehr gut ausgelastet. Der Median liegt hier sehr hoch bei 12.124 Bytes/s, wobei kaum Abweichungen und Ausreißer auftreten. Die Übertragung ist insgesamt als sehr stabil zu bewerten. Bei der Betrachtung der signifikanten Datensätze (vgl. Abschnitt 9.2.5) wird auf diese Übertragung noch im Einzelnen eingegangen.

9.2.2 Verallgemeinerung der Ergebnisse durch Betrachtung von Versuchsreihen

Während in den vorherigen Abschnitten die speziellen Charakteristika der Beispiel-Übertragungen herausgearbeitet wurden, wird jetzt der Frage nachgegangen werden, inwieweit die bisher gewonnenen Erkenntnisse verallgemeinert werden können.

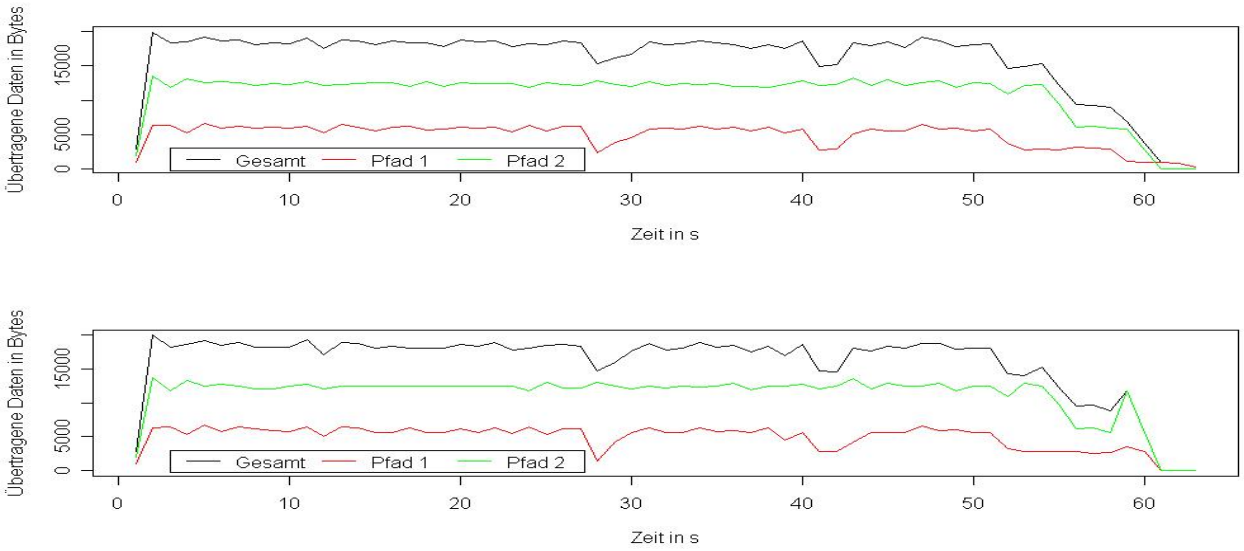


Abbildung 9.5: Signifikanter Datensatz zweier unterschiedlich großer Kanäle mit geringem Abstand und Pfadwahl durch das IN (a) auf Basis von Mittelwerten (b) auf Basis des Medians

Hierzu wurden die verschiedenen Versuchsreihen mehrmals hintereinander ausgeführt und jeweils zu einem für die Übertragung signifikanten Datensatz zusammengefasst. Es kamen zwei verschiedene Methoden zum Einsatz. Zum einen wurden die Mittelwerte, zum anderen die jeweiligen Median-Werte zur Beschreibung des signifikanten Datensatzes herangezogen.

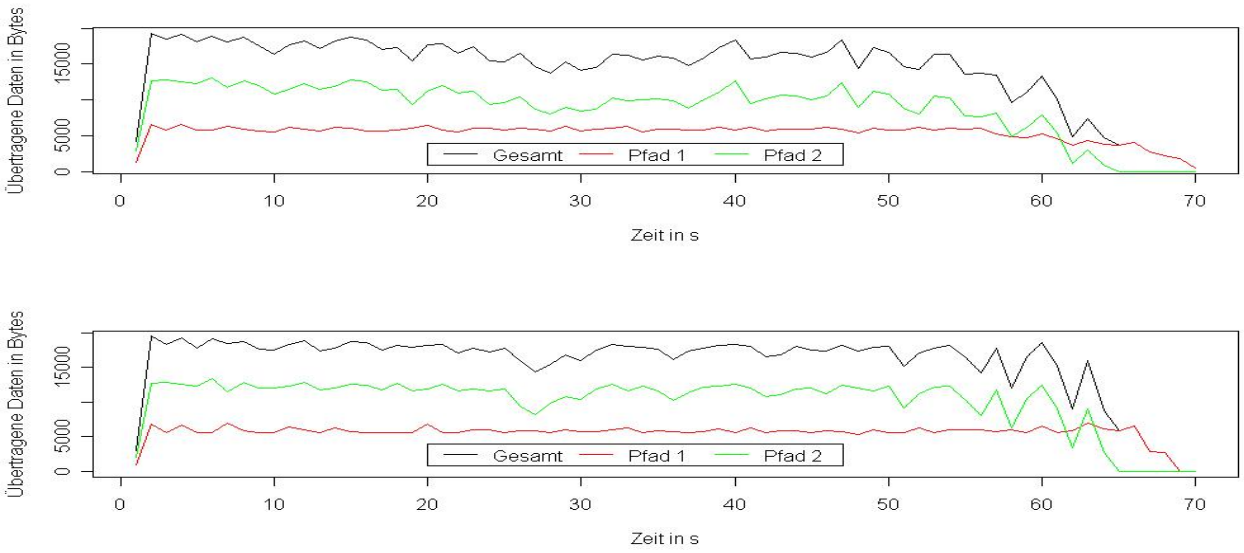


Abbildung 9.6: Signifikanter Datensatz zweier unterschiedlich großer Kanäle mit geringem Abstand und Pfadwahl durch A-priori-Wissen (a) auf Basis von Mittelwerten (b) auf Basis des Medians

Grundlage hierfür bilden die übertragenen Daten pro Sekunde. Auf Empfängerseite wurden die real eingetroffenen Daten abgefangen und auf der Zeitachse aufgetragen. Werden mehrere Versuche durchgeführt, stehen pro Zeitintervall von einer Sekunde mehrere Werte zur Verfügung. Über jeweils diese Werte wurde der Median gebildet bzw. wurde der Mittelwert berechnet. Trägt man die so berechneten Werte auf der Zeitachse auf, entsteht der signifikante Datensatz. Die Abbildungen 9.5 und 9.6 sind auf diese Art und Weise entstanden. Für die Pfadwahl durch Loadsharing wurde auf eine solche Darstellung verzichtet, da sich das Loadsharing – zumindest für unterschiedlich große Kanäle – als nicht geeignet erwiesen hat.

Betrachtet man den *signifikanten Datensatz* für die Anwendung des IN in Abbildung 9.5, so werden die sehr guten Übertragungseigenschaften bestätigt, die bereits im vorherigen Abschnitt herausgearbeitet wurden. Bis auf kleinere Einbrüche in der Übertragung wurden die Daten bei sämtlichen Versuchen in sehr guten Zeiten knapp über 60 Sekunden übertragen. Selbst die Mittelwertdarstellung in Abbildung 9.5(a), die nicht resistent gegenüber Ausreißern ist, weist keine solchen auf. Die kleineren Einbrüche sind bei der Art und Weise der Übertragung nicht zu vermeiden, da kein speziell für die Mehr-Pfad-Übertragung spezialisiertes Protokoll verwendet, sondern lediglich auf SCTP aufgesetzt wird. Insbesondere werden keine grundsätzlichen SCTP-Eigenschaften und Algorithmen ausgehebelt, sondern auf das Standard-Verhalten von SCTP zugegriffen, welches wiederum nicht für den Multi-Pfad-Transfer vorgesehen ist. Durch eine geschickte Verteilung

können diese Nachteile zumindest neutralisiert werden.

Entsprechend wurde für die Übertragung unter Verwendung von a-priori Informationen vorgegangen (vgl. Abbildung 9.6). Man erkennt, dass durch den Median, aber auch durch die Mittelwerte, die Funktion geglättet wird. Die Schwankungen fallen bei einer solchen Vereinheitlichung kaum noch ins Gewicht. Aufgrund der etwas längeren Laufzeit von ca. 70 s sind die Übertragungen insgesamt minimal schlechter als beim IN. Es werden aber beide Pfade konstant genutzt, sodass bei Kenntnis der zugehörigen Pfadeigenschaften diese Form der Übertragung auch ohne zusätzlichen Aufwand durch das IN durchgeführt werden kann.

9.2.3 Kurz-Resümee der ersten Versuchsreihe

Man erkennt an den Ergebnissen, dass durch die verwendete LDA das IN in der Lage ist, aktiv auf Änderungen im Netz zu reagieren und eine optimale Pfadwahl durchzuführen. Dabei hat sich herausgestellt, dass das IN sogar bessere Ergebnisse liefert als die Verteilung aufgrund der bekannten Pfadeigenschaften. Somit kann festgehalten werden, dass unter der Voraussetzung, dass unterschiedlich große Kanäle verwendet werden, das IN eine stabile Übertragung auf beiden Kanälen liefert.

9.2.4 Vergleich bei Verwendung von Pfaden mit größerer Leistungsdifferenz

Die Ergebnisse aus Abschnitt 9.2.1 machen Hoffnung auf ähnlich gute Ergebnisse bei Änderungen der Pfadkonfigurationen. Als nächste Konstellation werden zwei Pfade gewählt, deren Kapazität weiter auseinander liegen, sodass ein deutlich größerer *dominanter* Pfad bereits ohne Verwendung des zweiten Pfades in der Lage sein sollte, die Daten in einer akzeptablen Zeit zu übertragen. Es ist zu vermuten, dass hierbei der Performancegewinn bzw. der Datenvolumen-Benefit nicht ganz so groß ausfallen wird wie bei den Pfaden mit geringeren Kapazitätsunterschieden.

Für die folgenden Testreihen wurde das grundsätzliche Vorgehen aus dem vorherigen Abschnitt beibehalten, lediglich die Kanalkapazitäten und die pro Zeiteinheit zu übertragende Datenmenge wurden variiert. Die Kanäle wurden über Nistnet auf 25.600 Bytes/s und 6.000 Bytes/s beschränkt. Der Delay wurde wie gehabt auf $45ms$ fest eingestellt. Als Datenvolumen wurden 100 Chunks a 250 Bytes pro Sekunde übertragen, wobei von einer Maximallast von 80 Chunks auf dem größeren und 20 Chunks auf dem kleineren Pfad ausgegangen wurde. Insgesamt wurden pro Experiment in einem Zeitraum von 60 Sekunden 6000 Chunks in das Netz eingespeist.

Vergleich von Standard-SCTP und der IN-Übertragung

Da von einer geringeren Differenz der Übertragung mittels IN und einer Standard-SCTP-Übertragung ausgegangen wird, werden diese Testreihen als Erstes betrachtet. In Abbildung 9.7 sind zwei charakteristische Übertragungsverläufe dargestellt. Dabei kenn-

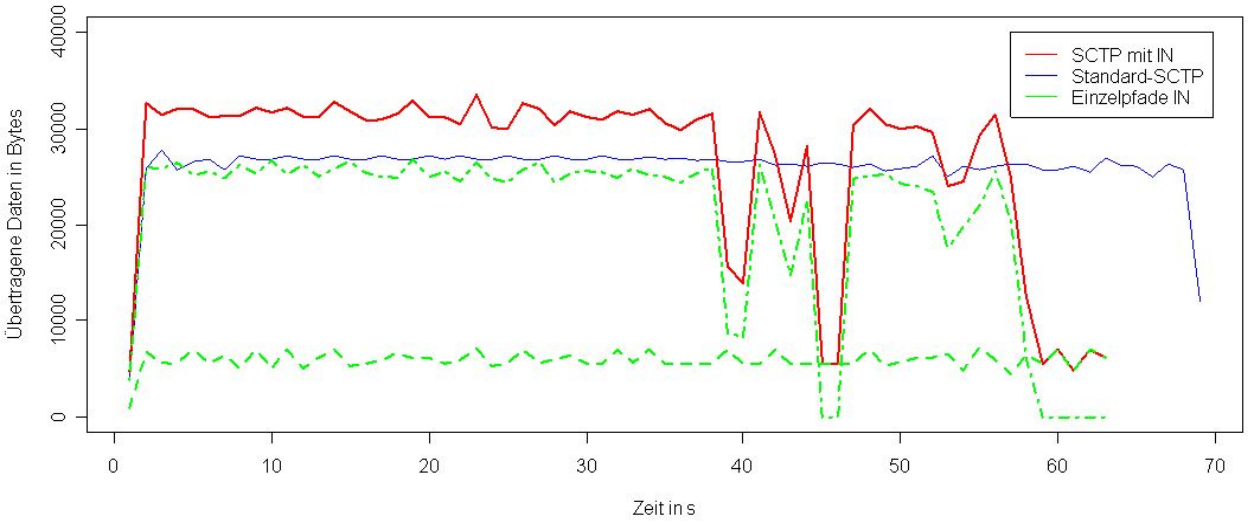


Abbildung 9.7: Vergleich der Testreihen mittels IN und Standard-SCTP

zeichnet die rote Funktion die Übertragung mittels IN, welche über die Daten der LDA gesteuert wurde. Es handelt sich um die Gesamtübertragung, d.h. beide Kanäle wurden berücksichtigt. Die blaue Linie beschreibt die Übertragung mittels Standard-SCTP. Um Aussagen über die Auslastung und Stabilität der einzelnen Kanäle bei Übertragung unter Verwendung des IN machen zu können, sind diese in grün zusätzlich aufgetragen.

Wie zu erwarten war, überträgt Standard-SCTP die Daten auf einem konstant hohen Niveau und hat nach ca. 70 *sek* die Übertragung vollständig beendet. Für die Übertragung wurde ein Primärpfad mit einer Kapazität 25.600 Bytes pro Sekunde verwendet, er entspricht demnach dem größeren Pfad der IN-Übertragung. Vergleicht man den größeren Pfad der IN-Übertragung mit dem Primärpfad von SCTP, stellt man fest, dass die maximale Leistung des Kanals nicht erreicht wird, da die obere grüne Kurve immer unterhalb der blauen SCTP-Kurve liegt. Nimmt man den zweiten Pfad hinzu, der auf hohem Niveau überträgt und während der gesamten Übertragung ausgelastet ist, kommt man allerdings auf einen größeren Durchsatz durch die Verwendung des IN zur Pfadwahl.

Das im vorherigen Abschnitt herausgearbeitete Problem des kurzfristigen Einbruchs der Übertragung auf dem größeren Kanal durch kurzfristige Staubildung in der Sendqueue tritt auch hier auf. So wird nach ca. 50 Sekunden die Übertragung auf dem größeren Kanal deutlich gesenkt. Dieses ist aber nur temporär, da durch die geschickte Verteilung der Stau in der Sendqueue schnell abgearbeitet werden kann. Insgesamt läuft die Gesamtübertragung zwar nicht so flüssig wie beim Standard-SCTP, aber der erhöhte Durchsatz spricht für das IN.

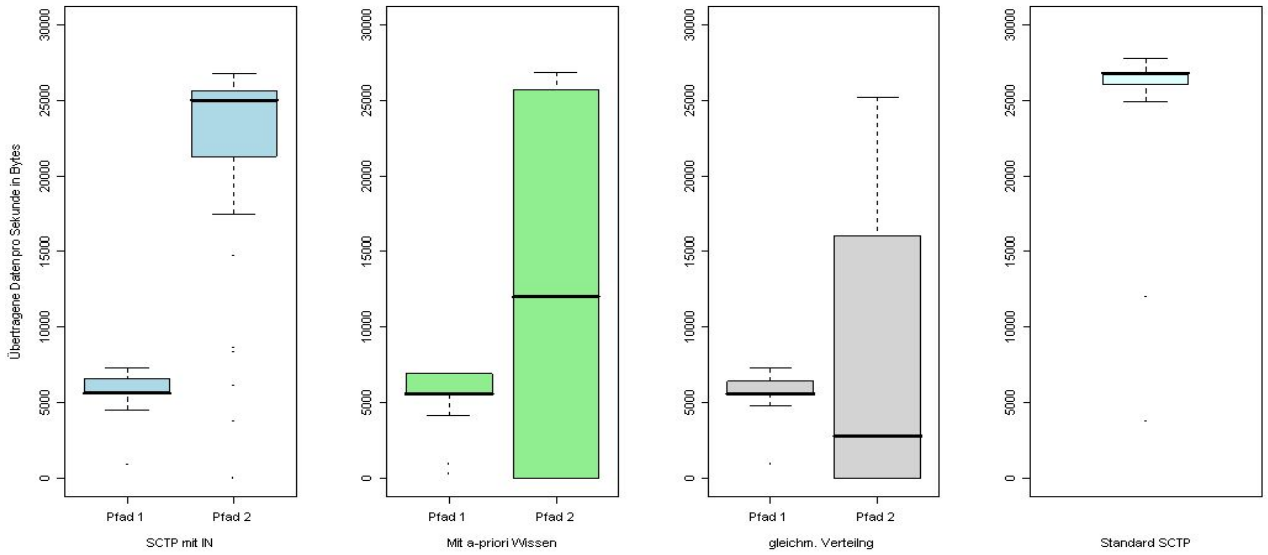


Abbildung 9.8: Vergleich der wesentlichen Kennzahlen der verschiedenen Beispielübertragungen (a) SCTP mit IN (b) Pfadwahl durch a-priori Wissen (c) Pfadwahl durch Loadsharing (d) Standard-SCTP

In Abbildung 9.8 sind die statistischen Kennwerte für die verschiedenen Testreihen grafisch dargestellt, wobei derzeit Teilabbildung (a) für die Übertragung mittels IN und Teilabbildung (d) für die Übertragung mit Standard-SCTP von Interesse sind. Man erkennt deutlich die exzellenten Übertragungseigenschaften des Standard-SCTP. Der Median liegt mit 26.752 Bytes/Sekunde deutlich über dem Median des größeren Kanals des IN, der lediglich 20.724 Bytes pro Sekunde übertragen hat. Hinzu kommen die hochwertigen Quartile mit 26.040 bzw. 26.860 Bytes pro Sekunde, die für einen konstant guten Durchsatz sprechen. Die vorhandenen Ausreißer sind vernachlässigbar.

Aber auch die Boxplots der IN-Pfade sprechen für eine saubere und stabile Übertragung. Zwar trüben einige Ausreißer und die größere Breite des Boxkernbereichs das Gesamtbild, in der Summe liegt aber ein größerer Datendurchsatz als beim Standard-SCTP vor. Der Boxplot bestätigt somit die positiven Ergebnisse, die aus der Auswertung von Abbildung 9.7 bereits gewonnen werden konnten.

Loadsharing und Übertragung mit a-priori Wissen

Nachdem die Auswertung der Pfadwahl durch das IN eine sehr gute Auslastung der Kanäle ergeben hat, sind demgegenüber die gängigen Alternativen *Loadsharing* sowie die Übertragung unter Verwendung der als bekannt vorausgesetzten voreingestellten Ka-

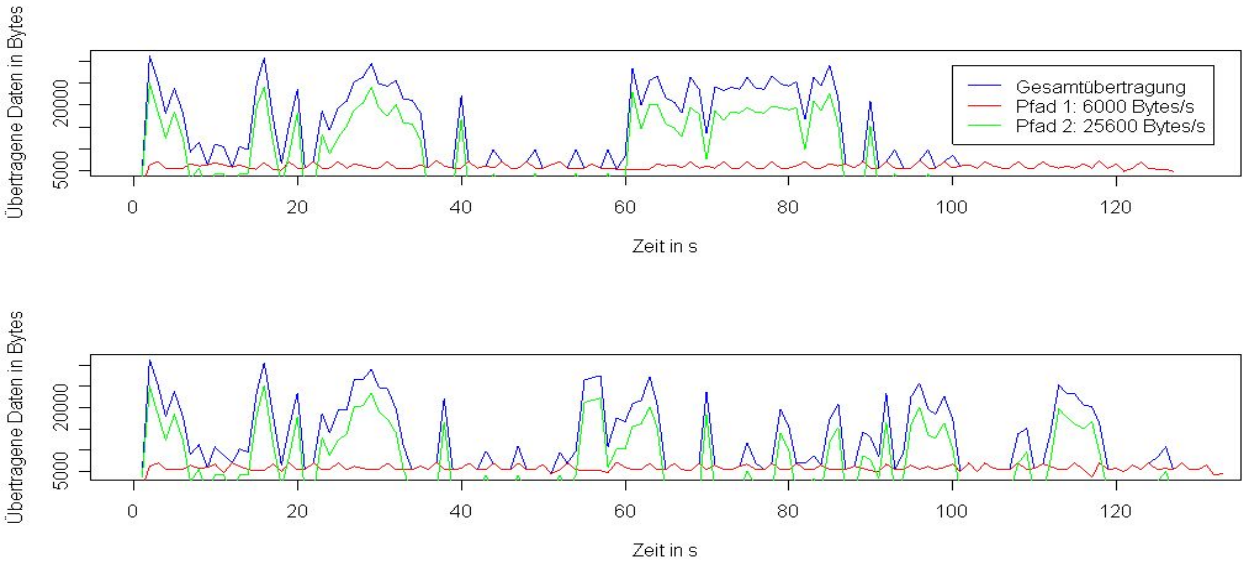


Abbildung 9.9: (a) und (b) Zwei beobachtete Grund-Charakteristika beim Loadsharing

pazitäten zu untersuchen.

Das Loadsharing lieferte bei den Testläufen zwei charakteristische Verläufe der Übertragung, die in Abbildung 9.9 aufgetragen sind. Der Verlauf in der unteren Messung ist ständigen Schwankungen unterworfen, sodass die maximale Datenübertragung nur zeitweise erreicht werden kann. Ursächlich hierfür ist wieder die Verwendung der Sendqueue und der zu stark beanspruchte kleinere Kanal. Zwar wird der Überschuss auf dem Kanal schnell abgebaut, aber genauso schnell wieder aufgebaut, sodass keine konstante Übertragung zustande kommt.

Ähnliches gilt für die Übertragung in der oberen Teilabbildung. Hier werden zwar größere Blöcke an Daten auf hohem Niveau übertragen, diese werden allerdings von mit der Zeit immer größer werdenden Einbrüchen unterbrochen, sodass auch hier keine saubere Übertragung der Daten erreicht werden kann. Besonders negativ fällt auf, dass die letzten 20 Sekunden nur noch der kleinere Kanal für die Übertragung genutzt wird bzw. die dort aufgestauten Daten abarbeitet werden.

So unterschiedlich beide beobachteten Verläufe auch sein mögen, das Ergebnis ist bei beiden identisch. Die Übertragung nimmt trotz der Verwendung von zwei Kanälen sehr viel Zeit in Anspruch. In diesem Fall werden ca. 120 Sekunden benötigt, um die in 60 Sekunden eingespeisten Daten vollständig zu übertragen. Im Vergleich dazu benötigt Standard-SCTP für die Übertragung lediglich ca. 70 Sekunden, wobei nur der Primärpfad zur Verfügung steht. Die Verteilung der Daten mit Hilfe des IN ist sogar noch effektiver,

sodass nach spätestens 65 Sekunden alle gesendeten Daten beim Empfänger eingetroffen sind. Diese Werte werden im Abschnitt 9.2.5 auf Verlässlichkeit überprüft.

Somit stellt das *Loadsharing* in diesem Szenario keine Lösung des Pfadwahlproblems dar, wie sieht es aber mit der Alternative unter Verwendung von a-priori Wissen aus? Bei der a-priori Übertragung wurden die Kanalkapazitäten der beiden Pfade als bekannt vorausgesetzt und die Daten im Verhältnis der Kapazitäten 6.000 : 25.600, also ca. 1 : 4 übertragen.

Um einen charakteristischen Verlauf der Übertragung zu erhalten, wurden auch für dieses Szenario mehrere Durchläufe vorgenommen. Es konnte aber kein eindeutig reproduzierbares Verhalten ermittelt werden, sodass sich kein charakteristischer Verlauf herauskristallisieren konnte. Einige Verläufe sind beispielhaft in Abbildung 9.10 zusammengestellt. In der unteren Bildreihe in Abbildung 9.10 ist zum besseren Vergleich nochmals die bereits beschriebene Funktion mittels IN aufgetragen.

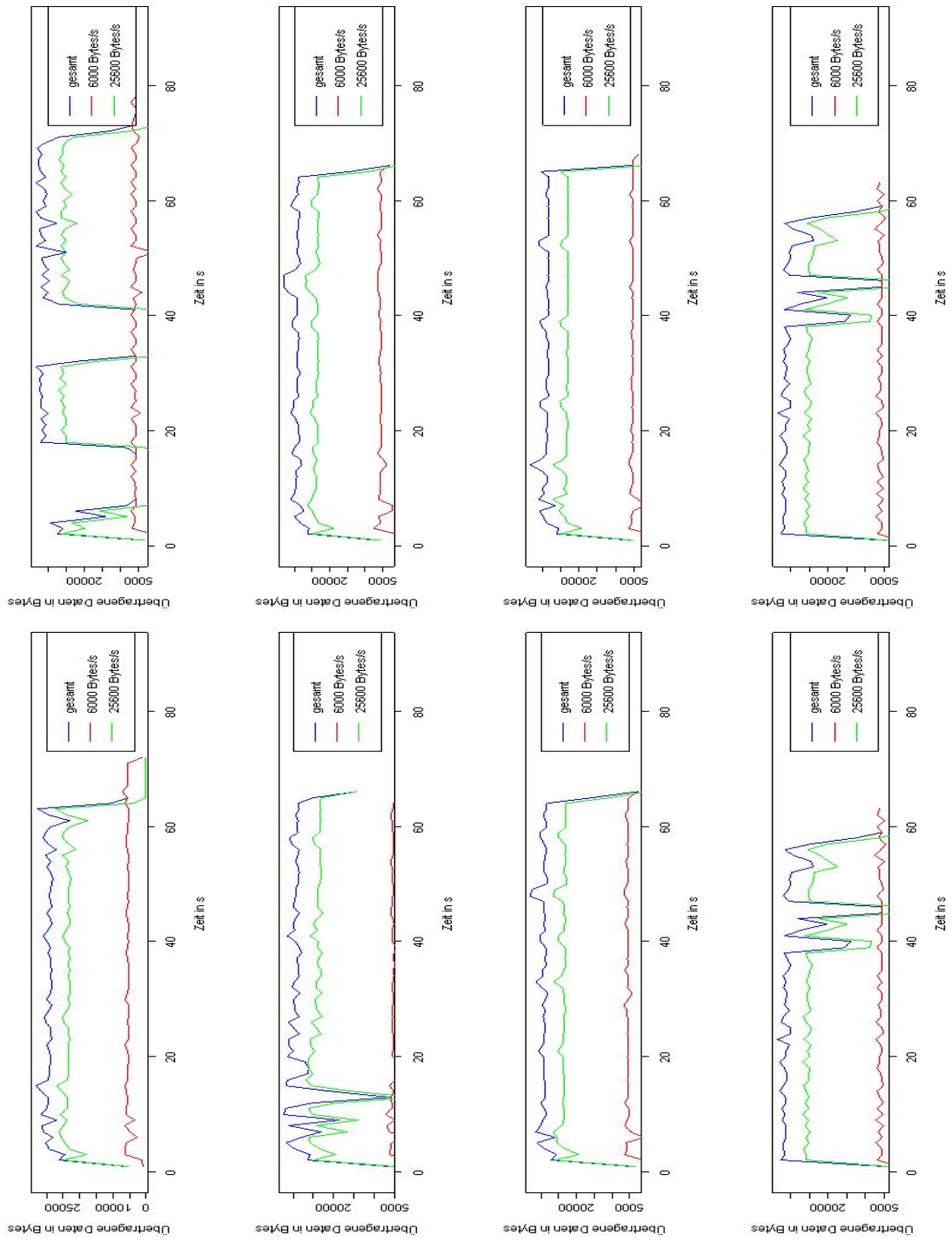


Abbildung 9.10: Varianten der a-priori Versuchsreihe im Vergleich zum IN

Auch bei dieser Übertragung dominiert der kleinere Pfad den größeren derart, dass die Übertragung nicht optimal ablaufen kann. Die Versuchsreihe aus Abbildung 9.10(b) stellt den ungünstigsten Verlauf dar, wobei für die Gesamtübertragung immerhin über 77 Sekunden benötigt wurden. Dies ist wieder erheblich länger als die Übertragung der Daten mit Standard-SCTP über einen Pfad, die lediglich ca. 70 Sekunden benötigt hat. Hierbei macht sich das Größenverhältnis der Pfade besonders bemerkbar. Der Versuch bestätigt aber die Annahme, dass lediglich die Verteilung der Daten auf mehrere Kanäle nicht zielführend ist, sondern während der Übertragung ständig die Netzauslastung beobachtet und ausgewertet werden muss, sodass die Pfadwahl intelligent, ergo zielgerichtet erfolgen muss.

Es handelt sich bei dieser ungünstigen Übertragung nicht um einen Einzelfall, ähnliche Verläufe wurden häufiger beobachtet. Ein besonders ungünstiger Verlauf erreichte sogar Zeiten von über 80 Sekunden für die vollständige Übertragung der Daten. Alle negativen Beispiele weisen ein gemeinsames Verhalten auf. Wenn es während der Übertragung zu Problemen kommt und ein Pfad aufgrund der Einreihung in die Sendqueue einbricht, ist das System nicht mehr in der Lage, dieses Problem zu bereinigen. Das Fehlverhalten setzt sich über den gesamten Zeitverlauf fort. Anders sieht es bei Verwendung des IN aus. Hier kann das System aktiv eingreifen und auf Probleme bei der Übertragung reagieren. In einem solchen Fall würden überschüssige Daten nicht blind in den kleineren Kanal eingespeist werden, sodass sich die Gesamtübertragung nach einem kurzen Versagen wieder auf hohem Niveau einpendeln würde.

Die anderen Versuchsreihen fallen deutlich besser aus, reichen aber an die IN-Variante nicht heran. So bricht in Abbildung 9.10(c) die Übertragung auf dem größeren Kanal nach kurzer Zeit ein, fängt sich aber sofort wieder. Hier muss sich anscheinend das System erst einschwingen. Aber auch bei den restlichen Testreihen, die keinen sichtbaren Einbruch der Leistung eines Kanals aufweisen, sind die Zeiten der Gesamtübertragung deutlich schlechter als beim zum Vergleich anstehenden IN.

Um die Betrachtung abzurunden, werden die restlichen Boxplots aus Abbildung 9.8 ausgewertet. In Abbildung 9.8(c) sind die Kennwerte der gleichmäßigen Verteilung der Daten auf beide Kanäle aufgetragen. Die Probleme der im vorherigen Abschnitt betrachteten Testreihe mit Loadsharing finden sich auch bei diesem Test wieder. So wird auch hier der größere Pfad derartig blockiert, dass sein Median sogar unter dem des kleineren Pfades liegt. Somit wird dieser Pfad nur unzureichend genutzt. Die weit auseinanderliegenden Quartile sowie die große Differenz zwischen Minimal- und Maximalwert zeigen auf, dass eine instabile und stark variierende Übertragung stattgefunden hat.

Eine ähnliche Aussage kann für die Übertragung unter Berücksichtigung von a-priori Wissen, deren Kennwerte in Abbildung 9.8(b) abgebildet sind, getroffen werden. Da keine charakteristische Übertragung abgeleitet werden konnte, wurde eine Übertragung nach dem Zufallsprinzip herausgegriffen. Es liegt eine noch größere Differenz zwischen dem oberen und dem unteren Quartil als beim Loadsharing vor. Somit kann auch hier

von einer nicht konstanten Übertragung gesprochen werden. Allerdings liegt der Gesamtdurchsatz deutlich über dem des Loadsharings, da der Median des größeren Kanals deutlich über den Kapazitätsgrenzen des kleineren Kanals liegt. Es wird, wenn auch nur zeitweise, ein sehr hoher Datendurchsatz von 26.823 Bytes pro Sekunde erreicht.

9.2.5 Verallgemeinerung der Ergebnisse durch Betrachtung von Versuchsreihen

In diesem Abschnitt wird entsprechend den Ausführungen in Abschnitt 9.2.2 geprüft, ob sich die Ergebnisse reproduzieren lassen. Hierzu wurde der *signifikante Datensatz* für die beiden konkurrierenden Übertragungen mittels IN bzw. unter Verwendung von a-priori Wissen jeweils in der Mittelwert- und Mediandarstellung erstellt.

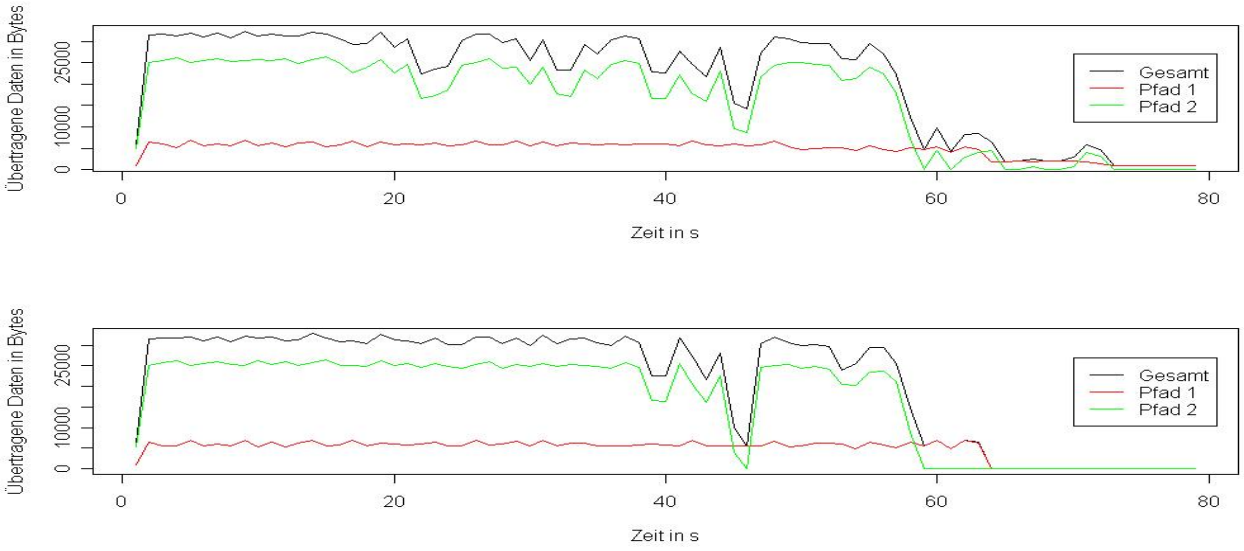


Abbildung 9.11: Signifikanter Datensatz zweier unterschiedlich großer Kanäle mit größerem Abstand und Pfadwahl durch das IN (a) Mittelwert (b) Median

Als Erstes soll der signifikante Datensatz der IN-Übertragung betrachtet werden. Bei den für die Abbildung verwendeten Versuchsreihen hat eine suboptimale Übertragung Eingang gefunden. Daher ist in der Betrachtung der Mittelwerte (vgl. Abbildung 9.11(a)) sogar eine Übertragung mit einer Laufzeit von bis zu 80 Sekunden aufgetragen. Dass es sich bei diesen Werten um einen einmaligen Ausreißer handelt, lässt sich anhand der Medianübersicht (vgl. Abbildung 9.11(b)) erkennen, da dort die längste Übertragung gerade einmal 63 Sekunden beträgt. Die Medianabbildung liefert somit eine realitätsnahe Darstellung, die den Verlauf der Übertragung charakterisiert.

Auffällig ist der Einbruch, der konstant nach ca. 44 Sekunden eintritt und in beiden

Ansichten deutlich hervorsticht. Dies deutet darauf hin, dass die einzelnen Übertragungen bei gleicher Ausgangssituation reproduzierbare Funktionen liefern. An dieser besagten Stelle ist die Verteilfunktion nicht optimal eingestellt, sodass hier das Sendqueue-Problem erkennbar ist. Allerdings reagiert das IN sehr schnell auf dieses Problem und stellt zeitnah die maximale Übertragung auf dem größeren Kanal wieder her. Insgesamt kann zwar ein Zeitverlust festgestellt werden, der aber in der Gesamtbetrachtung vernachlässigbar ist. Der Datendurchsatz liegt dabei deutlich über den Konkurrenzprodukten.

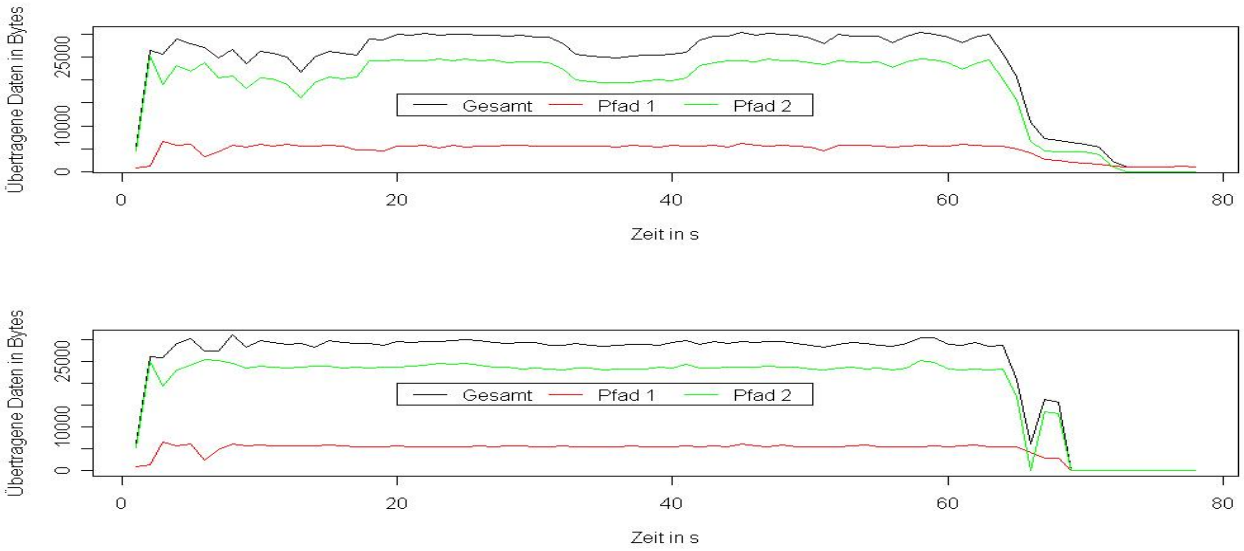


Abbildung 9.12: Signifikanter Datensatz zweier unterschiedlich großer Kanäle mit größerem Abstand und Pfadwahl durch a-priori Wissen (a) Mittelwert (b) Median

Kommen wir zu den Konkurrenzprodukten. In Abbildung 9.12 sind die entsprechenden signifikanten Datensätze der Pfadwahl unter Verwendung der bekannten Kanalkapazitäten aufgetragen. Im vorherigen Abschnitt wurden bereits einige Übertragungsfunktionen exemplarisch untersucht und festgestellt, dass sich kein eindeutiger charakteristischer Verlauf extrahieren lässt. Dies spiegelt sich auch im signifikanten Datensatz wider.

Durch das Aufsummieren der einzelnen – teilweise stark variierenden – Teilübertragungen entsteht der Eindruck einer stabilen Übertragung, der wie bereits erläutert in der Einzelansicht nicht bestätigt werden kann. Allerdings kann aus statistischer Sicht auf einen konstanten Verlauf geschlossen werden, d.h. unabhängig vom tatsächlichen Verlauf der einzelnen Übertragung kann eine Aussage für zukünftige Übertragungen abgeleitet werden. So ist die Gesamtlaufzeit und somit der durchschnittliche Datendurchsatz sehr

wohl für sämtliche Versuchsreihen vergleichbar, da die Übertragung im Normalfall nach 70 Sekunden abgeschlossen ist und nur in Ausnahmefällen längere Zeit in Anspruch nimmt.

9.2.6 Kurz-Resümee der zweiten Versuchsreihe

Man erkennt, dass das IN in der Lage ist, sich auch an extremere Konstellationen anzupassen. So konnte ein im Verhältnis zur Gesamtübertragung sehr klein gewählter Kanal zusätzlich und effizient für die Übertragung genutzt werden. Bei sämtlichen betrachteten Verteilalgorithmen konnte festgestellt werden, dass die reine übertragene Nutzlast deutlich unter der maximal möglichen Auslastung der einzelnen Pfade liegt.

Daraus folgt, dass der Benefit der Multi-Pfad-Übertragung im direkten Verhältnis zu den Kapazitätsdifferenzen der verwendeten Kanäle steht. Wenn die Kapazitäten der zu verwendenden Pfade als bekannt vorausgesetzt und lediglich ein extrem kleiner Kanal dem System hinzugefügt werden kann, sollte geprüft werden, ob es für das Gesamtsystem nicht vorteilhafter ist, eine Standard-SCTP-Übertragung zu verwenden.

9.3 Besondere Betrachtung des IN bei Kanälen gleicher Kapazität

Die positiven Ergebnisse der vorherigen Betrachtungen legen die Vermutung nahe, dass die Übertragung über zwei gleich große Kanäle noch besser abläuft, da die Generierung der Trainingsdatensätze gerade unter Verwendung genau eines festgelegten Kanals stattgefunden hat. Dies konnte allerdings nur begrenzt bestätigt werden, wie die folgenden Untersuchungen zeigen.

9.3.1 Auswertung der Messergebnisse

Auch für dieses Szenario wurden aufgrund der Vergleichbarkeit der Ergebnisse die Messungen nach dem bewährten Schema durchgeführt.

Bei jedem Versuch wurden demnach 60 Sekunden lang 80 Datenchunks pro Sekunde in das Netz eingespeist. Jeder Chunk hat eine feste Größe von 250 Bytes. Da beide Pfade dieselbe Kapazität besitzen, fallen das *Loadsharing* und die Übertragung durch *a-priori Wissen* zusammen und werden im Folgenden unter *Loadsharing* zusammengefasst.

Standard-SCTP und Loadsharing als Referenzübertragung

Bei den Experimenten hat sich gezeigt, dass bei gleich großen Kanälen das *Loadsharing* besonders effektiv ist. Es bietet sich daher an, das *Loadsharing* und die *Standard-SCTP-*

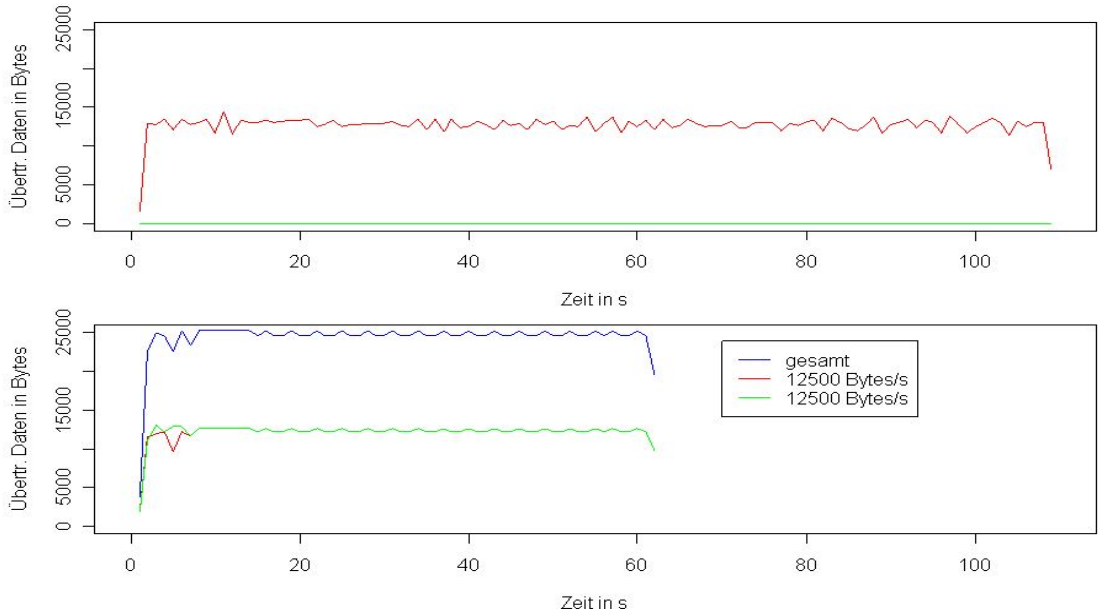


Abbildung 9.13: Referenzübertragungen für die Übertragung über zwei gleich große Kanäle (a) Standard-SCTP (b) Loadsharing

Übertragung im Vorfeld als Referenzübertragung zu betrachten und danach mit den Auswertungen der IN-Übertragung abzugleichen. In Abbildung 9.13 sind zwei charakteristische Verläufe aufgetragen.

Die Übertragung über Standard-SCTP unter Verwendung des Primärpfades ist in Abbildung 9.13(a) aufgetragen. Man erkennt die stabile Übertragung der Daten durch SCTP. Da nur ein Pfad zur Übertragung genutzt wird, dauert die Gesamtübertragung entsprechend länger bzw. ist nach genau 120 Sekunden vollständig abgeschlossen. Es konnte also der maximale Durchsatz des Pfades auch tatsächlich erreicht werden, da in der Testdauer von 60 Sekunden die Datenmenge für zwei Pfade eingespeist wurde.

Bei Gleichverteilung kann die eingespeiste Datenmenge in 60 Sekunden (vgl. Abbildung 9.13(b)) übertragen werden. Beide Pfade zeigen einen fast identischen Verlauf im oberen Kapazitätsbereich der Pfade. Beide Kanäle sind optimal ausgelastet. Dieses Ergebnis konnte auch in der statistischen Auswertung (vgl. Abschnitt 9.3.2) bestätigt werden. Falls für eine Datenübertragung bekannt ist, dass zwei Kanäle gleicher und gleich bleibender Kapazität verwendet werden können, ist sicher das *Loadsharing* die erste Wahl zur Verteilung auf die einzelnen Pfade.

Dieser Eindruck kann durch die Darstellung der statistischen Kennwerte in Form des Boxplots bestätigt werden. Die zugehörigen Grafiken sind in Abbildung 9.14(c) und (d) festgehalten. Die Übertragung mittels Standard-SCTP weist auch in dieser Darstellung

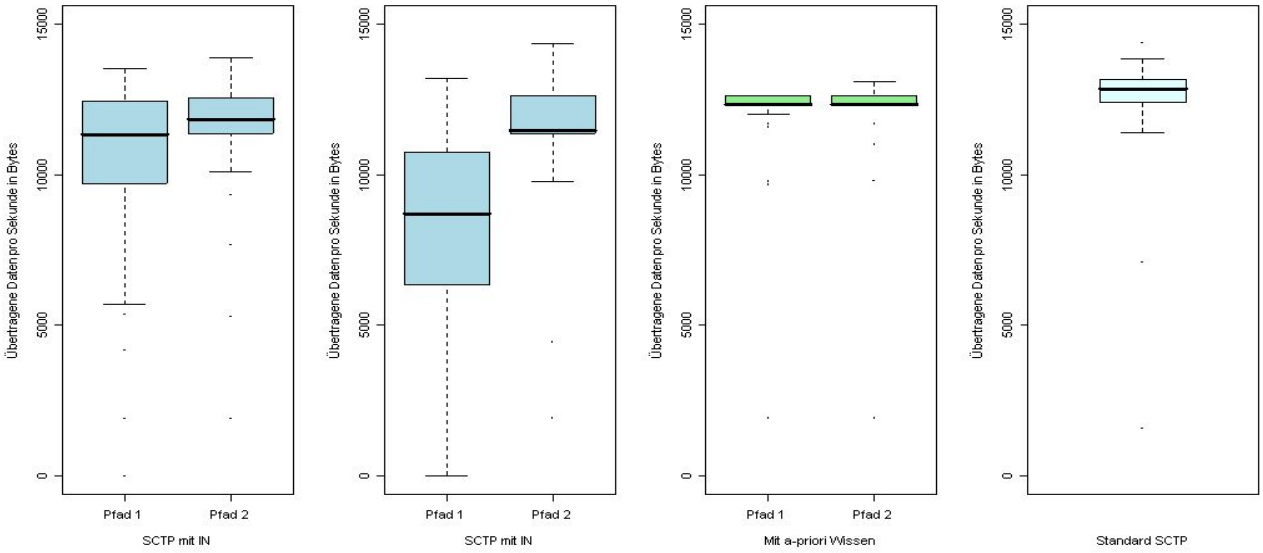


Abbildung 9.14: Darstellung der statistischen Kennwerte für einzelne Messreihen bei Verwendung von gleich großen Pfaden

keine negativen Punkte auf. Der Median liegt auf hohem Niveau, die Abstände der Quartile sind sehr gering, dies gilt auch für die Minimal- und Maximalwerte. Vorhandene Ausreißer entstehen lediglich am Anfang und am Ende der Übertragung und sind zu vernachlässigen. Man kann festhalten, dass die Übertragung bei allen verwendeten Kanalvarianten über Standard-SCTP (vgl. Abbildung 9.14 und Abbildung 9.13) im Rahmen des Primärpfads ordnungsgemäß durchgeführt wurde.

Die Übertragung mittels *Loadsharing* schneidet bei dieser Betrachtung noch besser ab. Die bei Übertragung mittels Standard-SCTP beschriebenen Eigenschaften des Primärpfades sind auf beiden Kanälen wiederzufinden. Der Median liegt bei beiden Kanälen mit 12.324 Bytes pro Sekunde nur minimal unter dem Referenzwert von 12.828 des Primärpfads von Standard-SCTP. Demgegenüber liegen die Quartile und Extremwerte noch dichter beieinander, was für eine extrem gute Auslastung der Kanäle spricht. Eine bessere Übertragung lässt sich kaum erreichen.

Übertragungsfunktion unter Verwendung des IN

Nach diesen überaus positiven Ergebnissen beim *Loadsharing* stellt sich jetzt die Frage, ob das IN ähnlich gute Ergebnisse liefert.

Es hat sich keine eindeutig charakteristische Übertragung herauskristallisiert, sondern es

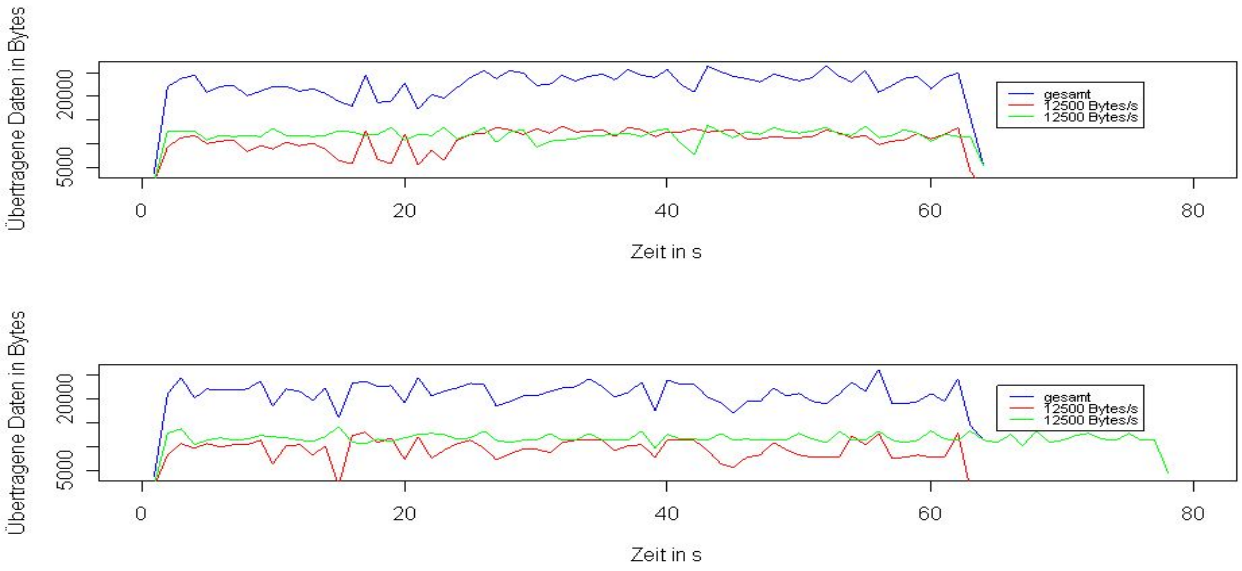


Abbildung 9.15: Zwei repräsentative Übertragungen auf gleich großen Kanälen mit Pfadwahl durch das IN (a) optimale Übertragung (b) problematische Übertragung

haben sich zwei Grundformen, wie sie in Abbildung 9.15 dargestellt sind, ergeben. Die Verteilung über das IN konnte nicht immer an die guten Ergebnisse des Loadsharings heranreichen. Auffällig ist, dass sehr gute Durchläufe wie in Abbildung 9.15(a), aber auch schlechtere wie in Abbildung 9.15(b), vorgekommen sind. Falls sich das System eingeschwungen hat und sich auf beiden Kanälen eine optimale Übertragungsfunktion eingestellt hat, bleibt die Übertragung bis zum Ende auf diesem hohen Niveau stehen, und die Daten können optimal übertragen werden. Die Werte reichen an die Referenzübertragung mittels Loadsharing heran und sind somit als gleichwertig zu betrachten.

Falls ein solcher Einschwingvorgang nicht realisiert werden kann, hat das Gesamtsystem Probleme. Insbesondere werden auf den schwächeren Kanal weniger Daten eingespeist als auf den gut laufenden Kanal. Daraus resultiert ein Nachlauf auf dem konstant laufenden Kanal. Die „restliche“ Datenmenge wird nur noch auf einem Kanal übertragen.

Abbildung 9.16 stellt die Verteilung mittels Loadsharing und das suboptimale Ergebnis unter Verwendung des IN vergleichbar gegenüber.

Sieht man sich die Parameter, die für die LDA herangezogen wurden, genauer an, kann man feststellen, dass diese erwartungsgemäß sehr dicht beieinanderliegen. Beide Kanäle weisen grundsätzlich die gleichen Übertragungseigenschaften auf, sodass dies nicht verwunderlich ist. Hier besteht sicher noch Verbesserungspotenzial. Liegen die Parameter-

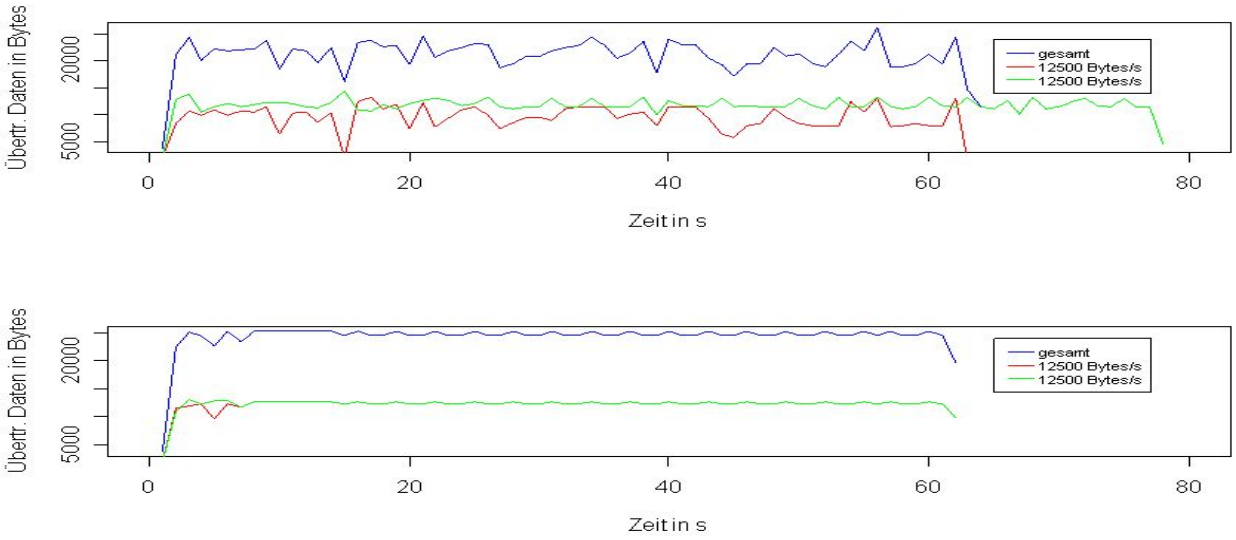


Abbildung 9.16: Vergleich der sehr guten Übertragung mittels Loadsharing (b) und einer nicht optimalen Übertragung mit Pfadwahl durch das IN (a)

werte zu dicht beieinander, kann es zu Fehleinschätzungen dahingehend kommen, welcher Kanal für die Übertragung besser geeignet ist. Um hier Abhilfe zu schaffen, muss man eine größere Trainingsdatenmenge verwenden. Dies ist aber auch nur bedingt möglich, da nicht für jeden möglichen Pfad Daten gesammelt werden, sondern nur repräsentative Kapazitäten ausgewählt werden können. Für alle nicht bekannten Einstellungen soll die LDA gerade diese Werte ermitteln. Als weitere Alternative bietet es sich an, weitere Verfahren und Algorithmen dahingehend zu untersuchen, ob sie für das Pfadwahlproblem geeignet sind und ggf. sogar bessere Ergebnisse liefern. Anhand der theoretischen Ableitung der Diskriminanzfunktion in Abschnitt 8.7.2 wurde bereits auf dieses Problem hingewiesen, da im Grenzbereich zwischen den jeweiligen Klassen ein hoher Anteil an nicht eindeutig klassifizierbaren Werten liegt.

Als Abschluss dieser Auswertung wird auf die statistischen Kennwerte für die IN-Übertragungen eingegangen. Man erkennt deutlich die guten Übertragungseigenschaften der „gelungenen“ Übertragung in Abbildung 9.14(a) und die Probleme der Übertragung in Abbildung 9.14(b). An diesem Beispiel treten die Vorteile der Darstellung durch Boxplots hervor. Man kann direkt an den statistischen Kennwerten eine Aussage zur Stabilität der Übertragung treffen.

Bei der problembehafteten Übertragung ist eindeutig der als Pfad 1 gekennzeichnete Pfad nicht korrekt ausgelastet worden. Der Median liegt deutlich unter dem des zweiten Pfades und erheblich unter den möglichen Werten, die aus den Referenzübertragun-

gen deutlich werden. Auch sind die Schwankungen sehr groß. Die Quartile liegen zwar weit auseinander, dies könnte aber durch eine konstant hohe Übertragung ausgeglichen werden. Problematisch sind hier die sehr großen Abweichungen von Maximal- und Minimalwert und natürlich der viel zu niedrige Median. Bei dieser Übertragung hat sich nie ein konstanter Verlauf der Übertragung eingestellt.

Anders sieht es bei der Übertragung in Abbildung 9.14(a) aus. Zwar werden auch hier nicht die hohen Ansprüche der Referenzwerte erreicht, die Gesamtübertragung ist aber aufgrund der hohen durchschnittlichen Übertragung sehr gut verlaufen.

9.3.2 Verallgemeinerung der Ergebnisse durch Betrachtung von Versuchsreihen

Entsprechend den Ausführungen in Abschnitt 9.2.2 wird geprüft, ob sich die Ergebnisse reproduzieren lassen. Hierzu wurde der *signifikante Datensatz* für die Übertragungen mittels IN und dem Loadsharing jeweils in der Mittelwert- und Mediandarstellung erstellt.

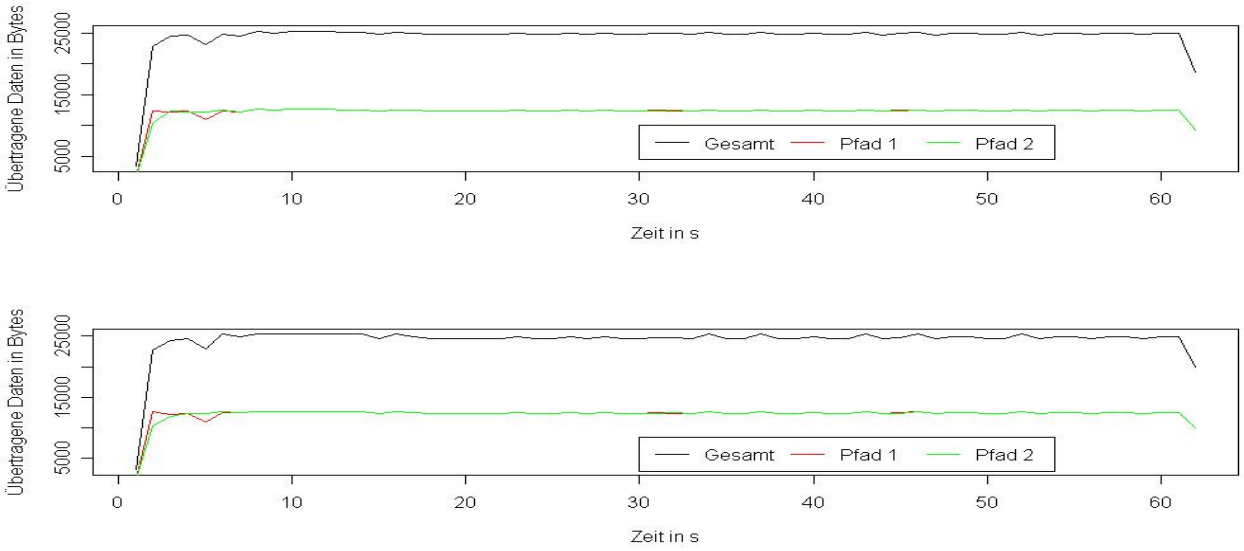


Abbildung 9.17: Signifikanter Datensatz zweier gleich großer Kanäle mit Pfadwahl durch Loadsharing (a) Mittelwert (b) Median

In Abbildung 9.17 sind die *signifikanten Datensätze* für das Loadsharing aufgetragen. Beide Darstellungen sind eindeutig und zeigen auf, dass sämtliche Testreihen gleich gut verlaufen sind. Das Ergebnis ist eindeutig reproduzierbar.

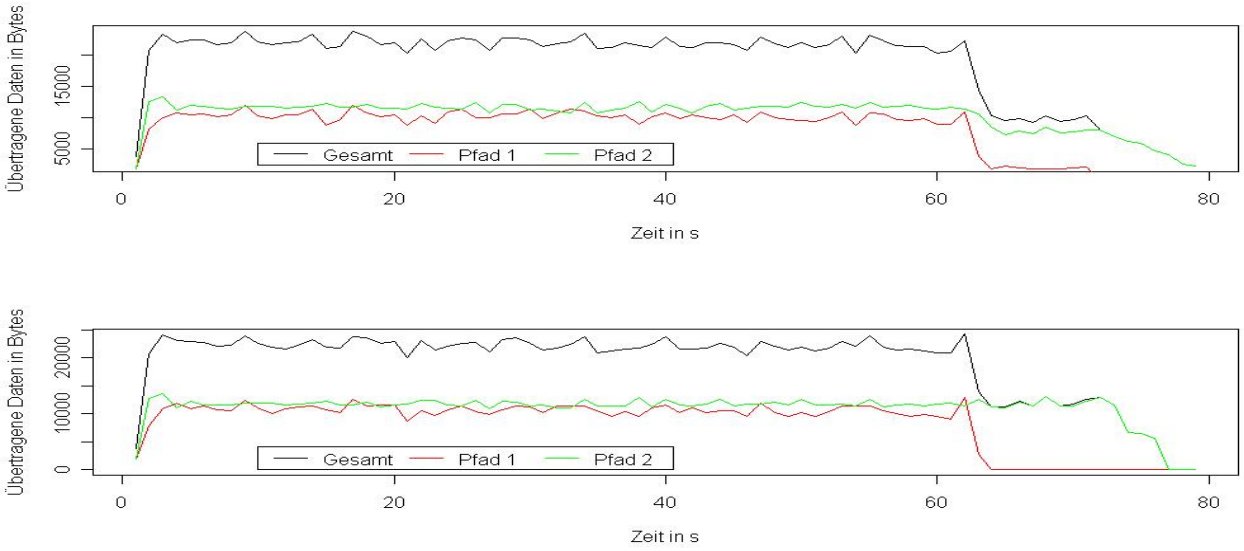


Abbildung 9.18: Signifikante Datensätze zweier gleich großer Kanäle mit Pfadwahl durch das IN (a) Mittelwert (b) Median

Beim *signifikanten Datensatz* des IN verhält es sich anders, da, wie im vorherigen Abschnitt erläutert, gute und weniger gute Übertragungen gemessen wurden. Dies drückt sich auch in Abbildung 9.18 aus. Da auch die Mediandarstellung in Abbildung 9.18(b) den relativ ungünstigen Verlauf der Mittelwertdarstellung (obere Grafik) bestätigt, kann geschlossen werden, dass es sich nicht um einen Ausreißer oder Einzelfall handelt, da dieser durch die Medianabbildung entfernt worden wäre. Es folgt, dass eine Übertragung auf den hier verwendeten gleich großen Kanälen zwischen 70 und 80 Sekunden benötigt und somit eine ca. 30 % längere Laufzeit aufweist.

9.3.3 Beurteilung der Auswertung

Dieser Abschnitt ist geprägt von Zeichnungen, Grafiken und Auswertungen der Testreihen, wobei der Schwerpunkt bei der technischen Auswertung liegt. In diesem zusammenfassenden Unterabschnitt soll die Beurteilung der Ergebnisse und die Anwendbarkeit für *echte* Anwendungsszenarien erläutert werden.

Als erstes soll die Frage erörtert werden, unter welchen Voraussetzungen sich das hier vorgestellte Prinzip der Pfadwahl anwenden lässt bzw. wann andere Verteilmechanismen sinnvoll sind.

In den meisten Anwendungen ist davon auszugehen, dass die Datenverbindung als Black-Box fungiert und der Anwender bzw. das Anwenderprogramm keine Informationen über die Güte der zur Verfügung stehenden Pfade besitzt. In einem solchen Szenario können keine a-priori Informationen genutzt werden. Unabhängig davon, welche der betrachteten Versuchsreihen herangezogen werden, ist die gezielte Verteilung unter Kenntnis der Kanalkapazitäten die einzig echte konkurrenzfähige Alternative zur Übertragung mittels IN. Diese scheidet in diesen Szenarien ergo aus.

Kanäle gleicher Kapazitäten

Werden Kanäle gleicher Güte verwendet, so hängt die Wahl des Verteilalgorithmus von der Ausgangssituation ab. Wenn sichergestellt werden kann, dass auch immer zwei gleich gute Kanäle zur Verfügung stehen, sollte klassisches Loadsharing Verwendung finden.

Da dies im Normalfall aber nicht gegeben ist, kann das IN auch in diesem Szenario punkten. Die zwar sehr gute Übertragung mittels Standard-SCTP kommt als Alternative nicht wirklich in Frage, da durch die Nichtnutzung des zweiten Kanals, der lediglich für die Ausfallsicherung herangezogen wird, die Übertragung zwingend doppelt so viel Zeit benötigt wie eine optimale gleichverteilte Übertragung. Falls nicht andere Verfahren zur Verfügung stehen, die anhand der aktuellen Netzsituation die Pfadwahl aktiv begleiten, kann durch die Verwendung des IN der maximal mögliche Traffic pro Zeiteinheit deutlich erhöht werden.

Kanäle unterschiedlicher Kapazitäten

Es konnte gezeigt werden, dass unter der Voraussetzung, dass zwei unterschiedliche Kanäle zur Übertragung verwendet werden, das IN auch dann eine echte Alternative darstellt, wenn die Güte der Pfade im Vorfeld bekannt ist. Die Übertragungen mittels IN verliefen deutlich besser als bei Pfadwahl unter Verwendung von a-priori Wissen.

Liegen sehr große Differenzen zwischen den Kapazitäten der Kanäle vor, tritt ein interessantes Phänomen auf. Dadurch, dass der kleine Kanal derartig wenig zum Gesamtdurchsatz beisteuert, kann die optimale Übertragung über Standard-SCTP mit einem Pfad bessere Ergebnisse liefern als die Verteilung der Daten auf beide Kanäle. Aber auch hier konnte anhand der Messergebnisse gezeigt werden, dass die Verteilung mittels IN einen zumindest gleichwertigen Durchsatz ermöglicht.

Das IN erweitert das Transportprotokoll um eine aktive Komponente, die in der Lage ist, autonom aufgrund von vorher gelerntem Wissen die Pfadwahl zu treffen und eine optimale Übertragung zu ermöglichen. Die durchgeführten Experimente belegen, dass unabhängig von dem gewählten Szenario sich die Pfadwahl mittels IN bewährt hat.

Teil III

Sicherheitsaspekte bei der Übertragung von Medizindaten

10 SCTP – Erweiterungen des Standards

10.1 Erweiterungen von SCTP

Bevor auf spezielle Erweiterungen für den sicheren Datentransfer eingegangen wird, werden in diesem Abschnitt grundsätzliche Erweiterungen von SCTP thematisiert. Die meisten in Teil I der vorliegenden Arbeit vorgestellten Medizin-Szenarien lassen sich nicht nur auf Grundlage des Standard-SCTP-Protokolls realisieren. So ist beispielsweise für die Übertragung von Videostreams das PR-SCTP notwendig, um auch bei Fehlern eine unterbrechungsfreie Übertragung zu realisieren.

Beim Einsatz in einem Krankenwagen, dem nicht eine stationäre IP-Adresse zugewiesen werden kann, werden Erweiterungen benötigt, die speziell auf mobile Endgeräte zugeschnitten sind. Somit stellt sich nicht nur die Frage nach der Absicherung der SCTP-Verbindung, sondern es ist immer zu prüfen, ob auch die notwendigen Erweiterungen durch einen bestimmten Sicherungsmechanismus abgedeckt sind. Im weiteren Verlauf der Arbeit wird im Rahmen der Sicherheitsbetrachtung immer wieder auf die einzelnen Erweiterungen zurückgegriffen.

10.2 PR-SCTP

Grundsätzlich verwendet SCTP den zuverlässigen Versand der Daten, d.h. verloren gegangene Pakte werden neu übertragen, sodass sichergestellt ist, dass sämtliche Informationen beim Empfänger eingetroffen sind.

Dieses Vorgehen ist nicht immer gewollt bzw. sinnvoll. Werden beispielsweise Videodaten in Echtzeit gestreamt, wobei beim Empfänger keine Speicherung der Daten vorgesehen ist, sollte von der Neuübertragung abgesehen werden. Entweder steht die Übertragung, da auf ein fehlendes Paket gewartet wird oder die Wiedergabe läuft mit kleinen Fehlern weiter, sodass später eintreffende Pakete ungenutzt verworfen würden. Für solche Szenarien wird ein *teilgesicherter Transport* benötigt, wie er beispielsweise von *UDP* angeboten wird.

10.2.1 Grundsätzlicher Ablauf beim teilgesicherten Transport

Der teilgesicherte Transportmodus wird bei SCTP als Erweiterung angeboten, als *PR-SCTP*[1] bezeichnet und in [Stewart et al. 2004] spezifiziert.

Begriff der Zuverlässigkeit

Die einzelnen Protokolle haben ein unterschiedliches Verständnis vom Begriff *Zuverlässigkeit*, wie der folgende Vergleich zeigt. Im Angebot stehen neben SCTP die derzeit standardmäßig verwendeten Transportprotokolle *TCP* und *UDP*. Drei Fragestellungen stehen bei der Beantwortung der Frage der Zuverlässigkeit im Fokus: der *Paketverlust*, die *Ordnung* der Pakete beim Empfänger und die Frage nach den *Duplikaten*.

Verwendet man SCTP in der Standardvariante, so verhält es sich, zumindest was die Zuverlässigkeit anbetrifft, identisch mit TCP, d.h. es findet ein zuverlässiger Versand statt. Für verloren gegangene Pakete wird mittels Neuübertragung dafür gesorgt, dass sämtliche gesendeten Pakete auch beim Empfänger eingehen – es kommt zu keinem Paketverlust. Durch die Mechanismen der zuverlässigen Übertragung ist auch sichergestellt, dass keine Duplikate bei der Empfangsapplikation berücksichtigt werden müssen. Zudem ist eine Methode implementiert, die dafür sorgt, dass die Pakete in der „richtigen“ Reihenfolge bei der Empfängeranwendung ankommen, d.h. in der Reihenfolge, in der sie vom Sender verschickt wurden.

UDP ist als unzuverlässiges Protokoll konzipiert, sodass bei Problemen Datenpakete verloren gehen können. Auf der Empfängerseite werden die Daten „einfach“ durchgeschleust, d.h. es wird keine Methode angewendet, die bei Versandproblemen die ursprüngliche Paketreihenfolge wieder herstellt bzw. darauf achtet, dass keine doppelten Pakete an die Empfangsapplikation ausgeliefert werden. Somit treffen Datenpakete nicht zwingend geordnet beim Empfänger ein, zudem sind Duplikate nicht auszuschließen.

Kontrollierter Verlust

Bei PR-SCTP spricht man auch vom *kontrollierten Verlust* (engl.: controlled loss) oder auch von *zeitlich begrenzter Zuverlässigkeit* (engl.: timed reliability), da der Verlust von Paketen, die nach einer bestimmten Zeit nicht beim Empfänger eingegangen sind, in Kauf genommen wird.

Anhand von Abbildung 10.1 wird die Problematik deutlich. Eine zeitkritische Anwendung wartet auf die Chunks mit den TSNs $9 - 13$, die auch bereits beim Empfänger eingetroffen sind. Diese können aber noch nicht ausgeliefert werden, da der Chunk mit der TSN 8 noch aussteht. In einem solchen Fall ist die Option, den fehlenden Chunk

[1] engl.: Partially Reliable SCTP

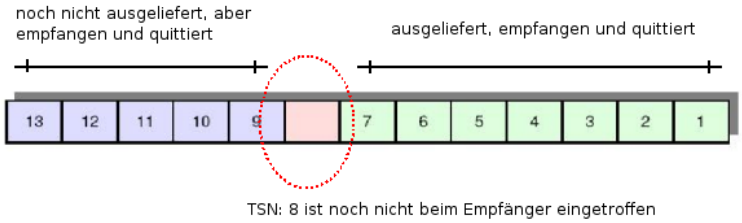


Abbildung 10.1: Eine zeitkritische Anwendung wartet auf Daten, die bereits eingetroffen sind.

nochmals zu übertragen, nicht sinnvoll, da beim Eintreffen des wiederholt gesendeten Chunks die zeitkritische Endanwendung die verspätet eintreffenden Chunks nicht mehr benötigt. Daher greift hier die PR-Erweiterung.

Die PR-Erweiterung legt fest, wann auf die Auslieferung eines verloren gegangenen Daten-Chunks verzichtet werden kann. Die Steuerung geht dabei vom Sender aus, d.h. der Sender entscheidet, wann ein Daten-Chunk „aufgegeben“, auch als *abandonment* bezeichnet, wird. Hierfür steht dem Sender ein neuer Chunk-Typ *Forward-TSN-Chunk*() zur Verfügung, mit dem er den Empfänger davon in Kenntnis setzt, dass die Auslieferung an die Zielanwendung ohne die im Forward-TSN-Chunk angegebenen Chunks fortgesetzt werden soll.

Um eine solche Entscheidung treffen zu können, greift der Sender auf die eingehenden SACK-Quittungen zurück. Die Generierung von *Forward-TSN-Chunks* erfolgt immer aufgrund der Informationen, die ein SACK-Chunk liefert. Dies hat einen positiven Nebeneffekt, da so direkt auf den Verlust eines bereits gesendeten *Forward-TSN-Chunks* reagiert werden kann. Es muss somit kein zusätzlicher Mechanismus implementiert werden, der überprüft, ob das *abandonment* auch angekommen ist.

In den folgenden Teilabschnitten wird der hier beschriebene grundsätzliche Ablauf in die SCTP-Terminologie eingebunden.

10.2.2 Terminologie von PR-SCTP

Da es sich bei PR-SCTP um eine Erweiterung des Standards handelt, müssen alle Endpunkte dieses zusätzliche Feature unterstützen.

Wird die PR-Erweiterung von den Endpunkten unterstützt?

Um sicherzustellen, dass die Endpunkte über die PR-Erweiterung verfügen, wird ein zusätzlicher optionaler Parameter für den INIT- bzw. den INIT-Ack-Chunk eingefügt. Mit dem *Forward-TSN-Supported*-Parameter kann der Sender wie auch der Empfänger angeben, ob er in der Lage ist, einen Forward-TSN-Chunk zu bearbeiten. Falls ein End-

punkt dazu nicht in der Lage ist, darf auch kein Forward-TSN-Chunk verwendet werden, d.h. bei Verwendung wird mit einer Fehlermeldung in Form eines Error-Chunks reagiert.

Verfall eines Daten-Chunks

Grundlage für den Verfall eines Daten-Chunks ist immer ein abgelaufener Timer bzw. das Ende der sogenannten *Lebenszeit* (engl.: lifetime) der Nachricht. Wie lange eine Nachricht gültig ist, wird demnach über ihre Lebenszeit geregelt, die wiederum von der ausführenden Applikation festgelegt wird.

Im Standard [Stewart 2007] wird neben den im Abschnitt 6 beschriebenen SCTP-Interna auch die funktionale Charakteristik einer Schnittstelle zwischen Anwendung (ULP) und SCTP-Kern beschrieben. Der Abschnitt 10.1 *ULP-to-SCTP* der Spezifikation beschreibt die Schnittstelle lediglich formal, sie muss entsprechend der für die Implementierung zu verwendenden Hochsprache umgesetzt werden.

In der Schnittstelle findet sich der Parameter für die Lebenszeit einer Nachricht wieder. Da als Basis für die praktische Auswertung die sctplib verwendet wurde, wird direkt auf Basis der dort zur Verfügung gestellten Funktionalität argumentiert.

Für die Übergabe von Daten in Form von Nachrichten stellt die *sctplib* die Funktion *SCTP_send* in der folgenden Struktur zur Verfügung:

```
int   SCTP_send(
          unsigned int associationID, unsigned short streamID,
          unsigned char* buffer, unsigned int length,
          unsigned int protocolId, short path_id,
          void*  context,   unsigned int lifetime,
          int unorderedDelivery,  int dontBundle);
```

Man erkennt, dass die Sender-Applikation gezielt über die *streamID* einen bestimmten Stream ansprechen und über den Parameter *lifetime* Einfluss auf die Lebensdauer der Nachricht, die über den *buffer* an SCTP übergeben wird, nehmen kann. Dieser Lebenszeit-Begriff wird im Folgenden als Basis für den Verfall einer Nachricht verwendet. Dies führt zum Begriff des *Abandoned-Chunk*:

Definition 10.2.1 (Abandoned-Chunk) *Ein Daten-Chunk wird als* Abandoned-Chunk *bezeichnet, wenn er vom Sender aufgrund*

1. *einer abgelaufenen Lebenszeit (lifetime) ohne eingegangene SACK-Quittung oder*
2. *einer expliziten Entscheidung der Sender-Applikation*

als „aufgegeben“ gekennzeichnet wurde.

Da sich die Lebenszeit auf eine Nachricht bezieht und nicht auf den Daten-Chunk heruntergebrochen wird, verfallen alle Fragmente einer fragmentierten Nachricht und somit alle zugehörigen Daten-Chunks. Welche Auswirkungen hat die Kennzeichnung eines Daten-Chunks auf den weiteren Verlauf des Transports?

Ein einmal gekennzeichneter Daten-Chunk wird so behandelt, als wäre er bereits beim Empfänger eingegangen und quittiert worden. Er wird ab diesem Zeitpunkt nicht mehr neu übertragen. Solange die Lebenszeit der Nachricht noch nicht abgelaufen ist, greifen die Standard-Mechanismen von SCTP, d.h. neben der Fluss- und Staukontrolle kann auch durch Neuübertragung von nicht rechtzeitig bestätigten Daten-Chunks ein zeitlich begrenzter zuverlässiger Versand erreicht werden. In welchen Abständen eine Neuübertragung möglich ist, wird über den im Abschnitt 6.3.5 eingeführten Retransmission-Timer (T3-RTX) geregelt. Somit kann über die Lebenszeit der Nachricht und der Größe des Retransmission-Timers gezielt das Verhalten der Übertragung gesteuert werden.

Die Staukontrolle wurde bereits erwähnt. Ein als aufgegeben markierter Daten-Chunk vergrößert nicht das zugehörige Sendefenster *cwnd*, d.h. die Berechnung des cwnd, wie es in Abschnitt 6.3.4 beschrieben ist, greift nicht. Dies entspricht der Logik, die hinter der Staukontrolle, wie sie in Abschnitt 6.4 beschrieben ist, steht, da ein aufgegebener Chunk nicht wirklich als Indiz für eine „freie Leitung“ angesehen werden kann.

Bisher hat nur der Sender Kenntnis vom Verfall der Nachricht, d.h. als Nächstes muss der Empfänger informiert werden. Dies erfolgt über den bereits angesprochenen neuen Chunk-Typen *Forward-TSN-Chunk*.

Den Empfänger von der Aufgabe einzelner Chunks in Kenntnis setzen

Der *Forward-TSN-Chunk* wird nur vom Sender verwendet und ist in Abbildung 10.2 abgebildet. Das Kernelement ist der *New Cummulative-TSN-Ack*, mit dem der Sender die Möglichkeit erhält, den realen Cum-Ack ohne Neuübertragung der Daten zu verschieben.

	← 32 Bit →			
Bits	**0 - 7**	**8 – 15**	**16 - 23**	**24 - 31**
0	Chunk-Typ 0x81	Flags = 0	Chunk-Länge – variabel	
32	New Cummulative TSN ACK			
64	Stream 1		Stream-Sequence-1	
...	...			
...	Stream n		Stream-Sequence-n	

Abbildung 10.2: Der Forward-TSN-Chunk zur Unterrichtung des Empfängers über zu vernachlässigende Chunks

Der Sender verwendet einen Hilfsparameter *Adv.Ack.Pt*, mit dem er die TSN und die aufgegebenen Daten-Chunks in Relation setzt. Der Adv.Ack.Pt enthält demnach den Wert der höchsten TSN der Abandoned-Chunks oder den Wert des Cum-Acks, falls dieser

TSN	SID	SSN	Status
…	…	…	
102	1	54	
103	1	55	abandoned
104	2	32	abandoned
105	1	56	abandoned
106	2	33	
…	…	…	

Sack-Chunk	
Cum-Ack	102

(a) Situation vor Eintreffen des SACK-Chunks

	TSN	SID	SSN	Status
	…	…	…	
Adv.-Ack.-Pt (1)	102	1	19	*Acked*
	103	1	20	abandoned
	104	2	77	abandoned
Adv.-Ack.-Pt (2)	105	1	21	abandoned
	106	2	78	gesendet
	…	…	…	

(b) Setzen des Adv.-Ack-Pt.-Parameters

	← 32 Bit →			
Bits	**0 - 7**	**8 – 15**	**16 - 23**	**24 - 31**
0	0x81	Flags = 0	Länge = 128	
32	New Cummulative TSN ACK = **105**			
64	Stream 1		**21**	
96	Stream 2		**77**	

(c) Der daraus resultierenden Forward-TSN-Chunk

Abbildung 10.3: Beispiel für die Verwendung des Forward-TSN-Chunks aus Sicht des Senders.

größer ist. Der Adv.Ack-Pt wird immer beim Eintreffen eines SACK-Chunks überprüft bzw. festgelegt. Zu diesem Zeitpunkt kann anhand des Cum-Ack-Wertes im SACK-Chunk die Sicht des Empfängers nachgestellt werden. Ein Beispiel soll dies verdeutlichen:

Beispiel 10.2.1 (Aktualisieren des Adv.Ack.Pt-Parameters) *Begleitend zum Beispiel ist die Sicht des Senders auf die Daten in Abbildung 10.3 dargestellt.*

Seien $103, 104$ *und* 105 *drei TSNs von Abandoned-Daten-Chunks,* 106 *die TSN eines noch nicht markierten aber bereits gesendeten Daten-Chunk und* $t_{sack} = 102$ *die vom SACK-Chunk gelieferte aktuelle Cum-Ack des Empfängers. Dies entspricht der Situation in Abbildung 10.3a.*

Der Sender legt als Erstes seinen Cum-Ack-Parameter fest, indem er den Wert vom SACK-Chunk übernimmt. Dieser Wert wird als initialer Wert für den Adv.Ack-Pt-Parameter verwendet. Somit wird im Beispiel der Adv.Ack.Pt auf 102 *gesetzt. Dies entspricht der Adv.-Ack-Pt.-Position (1) in Abbildung 10.3b.*

Im nächsten Schritt wird der Adv.Ack.Pt auf die höchstmögliche TSN eines Abandoned-Daten-Chunks gesetzt. Dies wäre im Beispiel die 105*, in Abbildung 10.3b ist diese Position als Adv.-Ack–Pt. (2) gekennzeichnet, da der darauf folgende Chunk nicht markiert*

und auch noch nicht quittiert wurde.

Gäbe es in diesem Szenario einen Abandoned-Daten-Chunk mit der TSN 108*, würde trotzdem der Adv.Ack.Pt auf* 105 *gesetzt, da die Daten-Chunks* 106 *und* 107 *noch über die normalen SCTP-Mechanismen, wie beispielsweise der Neuübertragung, abgewickelt werden müssen.*

Als Abschluss ist in Abbildung 10.3c der Forward-TSN-Chunk mit dem neuen Cum-Ack dargestellt. Die in der Abbildung verwendeten Stream-Informationen werden im folgenden Teilabschnitt erläutert und sind im Beispiel der Vollständigkeit halber aufgeführt.

Aus dem Beispiel ergibt sich bereits die Regel für den Versand eines Forward-TSN-Chunks. Wenn der Adv.Ack-Pt-Parameter einen größeren Wert enthält als der über den SACK gelieferten Cum-Ack-Wert, wird ein Forward-TSN-Chunk gesendet, wobei der Adv.Ack.Pt als neuer anzunehmender Cum-Ack an den Empfänger übermittelt wird. Dieser setzt seinerseits, nachdem er den Forward-TSN-Chunk entgegengenommen hat, seinen Cum-Ack-Parameter auf den übermittelten neuen Wert und verwirft die Abandoned-Daten-Chunks.

Nachdem der neue Wert für den Cum-Ack auf Empfängerseite feststeht, kann dieser alle bereits quittierten Daten-Chunks bis zur nächsten Lücke an die Zielanwendung ausliefern. Der sich so neu ergebende Wert für den Cum-Ack wird per SACK-Chunk an den Sender übermittelt, der somit eine Bestätigung seines Forward-TSN-Chunks erhält. Im Beispiel 10.2.1 kann so der bereits quittierte Daten-Chunk mit der $TSN = 106$ an die Ziel-Anwendung ausgeliefert werden.

Ein Sonderfall liegt vor, wenn die Lebenszeit einer Nachricht mit Null vorgegeben wird. In einem solchen Fall wird eine Nachricht immer nur einmal versendet. Es kann zu keiner Neuübertragung der zugehörigen Daten-Chunks kommen. Die Zeit, die für die Übertragung vorgesehen ist, bevor die Daten-Chunks aussortiert werden, ist durch den Retransmission-Timer (T3-rtx) festgelegt. Nach Ablauf des Retransmission-Timers werden die Daten-Chunks „abandoned“.

10.2.3 Vorteile von PR-SCTP

Im Vergleich zu anderen Protokollen, die einen teilgesicherten Datentransfer anbieten, kann SCTP punkten. Da auf dem Basis-SCTP-Protokoll aufgesetzt wird, stehen sämtliche Mechanismen zur Fluss- und Staukontrolle auch für den PR-Modus zur Verfügung. Anders als beispielsweise UDP zählt PR-SCTP zu den „fairen“ Protokollen, wie sie in Abschnitt 6.4 beschrieben sind, wodurch ein Congestion-Kollaps nach Definition 6.4.1 vermieden werden kann.

Zudem können die Mechanismen zur Neuübertragung des zuverlässigen Versands genutzt

werden, um so das Optimum an Fehlertoleranz zu erreichen. Zudem ist es möglich, einen geordneten teilgesicherten Versand zu realisieren, da auch hier auf die Mechanismen des verbindungorientierten Basis-SCTP-Protokolls zurückgegriffen werden kann.

Ein weiterer Vorteil ergibt sich, wenn die Multi-Streaming-Eigenschaft von SCTP konsequent mit der PR-SCTP-Erweiterung umgesetzt wird. Dies wird im folgenden Teilabschnitt gesondert erläutert.

10.2.4 Auswirkungen der PR-Erweiterung auf das Multi-Streaming

Der Vorteil des Multi-Streamings besteht u.a. darin, dass innerhalb einer einzelnen Assoziation Übertragungen unterschiedlicher Ausprägungen möglich sind. So kann eine Assoziation geordnete und ungeordnete Streams für die Übertragung anbieten, sodass verschiedene Anforderungen an die Übertragung mit nur einer Verbindung realisiert werden können. Diesem Konzept bleibt die Umsetzung von PR-SCTP treu, indem es möglich ist, in einer Assoziation Streams unterschiedlicher Zuverlässigkeit zur Verfügung zu stellen.

Neben den Informationen zum neu zu setzenden Cum-Ack können im Forward-TSN-Chunk Informationen über die verwendeten Streams übermittelt werden. Diese Informationen können grundsätzlich zur einfacheren Abarbeitung der Forward-TSN-Chunk-Informationen beim Empfänger genutzt werden. In [Jungmaier und Rathgeb 2005] wird aufgezeigt, wie diese Information für die Verwendung von Streams unterschiedlicher Zuverlässigkeitsklassen genutzt werden kann.

Um die Problematik beim Multi-Streaming zu verdeutlichen, wird auf das Beispiel 10.2.1 und die dazugehörige Abbildung 10.3 zurückgegriffen.

Beispiel 10.2.2 (Problem bei Streams unterschiedlicher Zuverlässigkeit)
Ändert man das Beispiele 10.2.1 ab, indem nur noch der Stream mit der Stream-ID= 2 im teilgesicherten Modus verwendet wird und der Stream mit der Stream-ID= 1 für den zuverlässigen Versand konfiguriert wurde.

Diese Voraussetzungen haben Auswirkungen auf die Festlegung des Adv.-Ack.-Pt. und damit auch auf den neuen Cum-Ack, der über den Forward-TSN-Chunk an den Empfänger übermittelt wird. Dadurch, dass der Chunk mit der $TSN = 103$ noch nicht quittiert wurde und im veränderten Beispiel nicht als aufgegeben gekennzeichnet werden darf, kommt als neuer Cum-Ack nur die 102 in Frage, da sonst der Empfänger auch die 103 als abgearbeitet betrachten würde.

Dies wiederum führt zu dem Problem, dass ein einzelner Daten-Chunk eines zuverlässigen Streams die Auslieferung der unzuverlässigen Streams verhindern würde.

Um diesem Problem zu begegnen, kann die zusätzliche Information im Forward-TSN-Chunk genutzt werden. Zu jedem ausgehandelten Stream wird im Forward-TSN-Chunk

die größte Sream-Sequence-Number der bereits quittierten oder aussortierten Daten-Chunks übermittelt. Im Beispiel 10.2.1 werden die Stream-Sequence-Nummern 21 und 77 an den Empfänger weitergereicht. Im abgewandelten Beispiel 10.2.2 würde für Stream Nummer zwei weiterhin die 77 übertragen, während für den Stream mit der $SID = 1$ nur die $SSN = 19$ als definitiv vollständig übertragen betrachtet werden kann.

Mit dieser zusätzlichen Information kann der Empfänger beide Streams unabhängig voneinander bewerten und könnte die Daten-Chunks mit der $TSN = 106$ und der $SID = 78$ an die Zielapplikation ausliefern. Es kommt zu keinem „Stau“ auf dem teilgesicherten Stream, obwohl der zuverlässige Versand auf dem Stream mit der $SID = 1$ ebenfalls gesichert ist.

Gerade bei der Echtzeitübertragung von Videodaten kann diese Koexistenz von teilgesicherter und zuverlässiger Datenübertragung effektiv genutzt werden.

11 Sicherheitslösungen für das SCTP-Protokoll

In Abschnitt 12 wird untersucht, inwieweit sich der Einsatz von Machine-Learning und adaptiver Kryptographie lohnt. Es wird ein Maß vorgestellt, mit dem verschiedene Kryptographische Systeme miteinander verglichen werden können, um so festzustellen, welcher Vorteil durch den Einsatz des IN im Bereich der Verschlüsselung erzielt werden kann.

Grundsätzlich verfügt SCTP über keine Mechanismen zur kryptographischen Absicherung einer Assoziation. Soll durch adaptive Algorithmen Einfluss auf die verwendeten Sicherheitsmerkmale Verschlüsselungsalgorithmus und Schlüssellänge genommen und aktiv eingegriffen werden, so müssen die Auswahl und Anwendung der Merkmale in der Transportschicht erfolgen. Für die durchgeführten Tests wurde daher auf *Secure-SCTP* zurückgegriffen, das die kryptographischen Algorithmen direkt in der Transportschicht verwendet. Somit kann eine intelligente Komponente in Abhängigkeit der von SCTP bereitgestellten Parameter, wie sie bereits für den Multipfad-Transport im Teil II dieser Arbeit erarbeitet wurden, aufgebaut werden.

Neben Secure-SCTP stehen noch weitere Sicherheitslösungen zur Verfügung, die entweder direkt oder in angepasster Form mit SCTP verwendet werden können. Um die Absicherung von SCTP in ihrer Gesamtheit betrachten und bewerten zu können, wird im Abschnitt 11.4 eine Zusammenfassung der Sicherheitslösungen gegeben, die neben Secure-SCTP von Bedeutung sind. Insbesondere wird auf die Unterschiede bzw. Vor- und Nachteile in Bezug auf Secure-SCTP eingegangen, anhand derer für jede Vorgabe und Anforderung die beste Sicherheitslösung ausgewählt werden kann.

11.1 SecureSCTP – Verschlüsselung und Authentifizierung auf Daten-Chunk-Ebene

Im Vergleich zu anderen Sicherheitslösungen ist Secure-SCTP speziell auf SCTP abgestimmt, sodass die Verschlüsselung bzw. Authentifizierung nicht nur auf Paketebene, sondern sogar bis auf Chunkebene heruntergebrochen werden kann. In diesem Abschnitt werden die wesentlichen Merkmale von Secure-SCTP kurz vorgestellt, wobei ein besonderes Augenmerk auf die Möglichkeit der Erweiterung im Zusammenhang mit dem IN gelegt wird.

11.2 Grundlagen

Als Erweiterung von SCTP setzt Secure-SCTP auf der Basisimplementierung von SCTP auf. Entsprechend der Vorgehensweise im vorherigen Abschnitt 10.1 wird die neue Funktionalität durch den Austausch von neuen Chunk-Typen ermöglicht. Um die Erweiterung nutzen zu können, müssen alle Endpunkte diese auch unterstützen. Auch hierfür setzt Secure-SCTP auf das Standardvorgehen, indem beim initialen Aufbau der Assoziation die notwendigen Parameter ausgetauscht bzw. ausgehandelt werden. Im Rahmen der Initialisierung kann die aufrufende Applikation auf einen möglicherweise fehlenden Sicherheitsbaustein reagieren.

Der Vorteil dieses Vorgehens liegt in der *Abwärtskompatibilität*. Falls die kryptographische Behandlung der Nachrichten nicht durch die Anwendung zwingend vorgeschrieben ist, kann die Datenübertragung über die Basisfunktionalität von SCTP durchgeführt werden.

11.2.1 Die sichere Session

Kern der Erweiterung ist die Definition der *Secure-Session*, die, nachdem sie etabliert wurde, den Rahmen für die sichere Datenübertragung bereitstellt.

Definition 11.2.1 (Secure-Session) *Eine* Secure-Session *erweitert den Transmission-Control-Block nach Definition 6.2.3 um die notwendigen Parameter für den sicheren Datentransfer und stellt den Endpunkten die zusätzliche Funktionalität hierfür zur Verfügung. Nach Aufbau einer Secure-Session sind die Endpunkte in der Lage, sichere und unsichere Übertragung von Daten in einem gemischten Szenario zu kombinieren.*

Die *Secure-Session* kann auch im laufenden Betrieb, d.h. bei Bestehen einer etablierten SCTP-Assoziation, aufgebaut werden. Über die zusätzlichen Chunks können in einem solchen Fall die notwendigen Parameter ausgetauscht werden.

Durch den Begriff der Secure-Session ist Secure-SCTP in der Lage, Daten-Chunks auf unterschiedliche Art kryptographisch zu behandeln. Es kann nicht nur zwischen kryptographischer Absicherung und unsicherem Versand unterschieden werden, sondern es kann aus dem gesamten Spektrum der möglichen Sicherheitseinstellungen gewählt werden. Welche Sicherheitseinstellungen von Secure-SCTP bereitgestellt werden, wird durch verschiedene *Security-Level* bestimmt.

11.2.2 Sicherheitseinstellungen und Security-Level

Basis-SCTP stellt keine kryptographische Funktionalität zur Verfügung. Lediglich zur Verhinderung von speziellen Angriffen beim Verbindungsaufbau wird die in Definition 6.2.5 eingeführte Hash-Funktion verwendet. Das im Abschnitt 6.2.1 näher beschriebene Verfahren dient aber nicht zur Bereitstellung von Sicherheit im kryptographischen

Sinn. Welche Anforderungen sind von einem Protokoll zu erfüllen, damit von einem *sichereren Datentransfer* gesprochen werden kann.

In [Schneier 1996] werden die Anforderungen *Geheimhaltung*, *Authentifizierung*, *Integrität* und *Verbindlichkeit* als wesentlich herausgestellt. Hinzu kommt häufig noch die Frage nach dem Schlüsselaustausch bzw. dem *Schlüsselmanagement*. Die verwendeten Schlüssel spielen bei der Betrachtung der Sicherheit eine große Rolle, da bei modernen Verfahren nicht etwa das eigentliche Verfahren geheimgehalten wird, sondern die Sicherheit des Gesamtverfahrens auf die Sicherheit des einzelnen Schlüssels reduziert werden kann.

Definition 11.2.2 (Authentifikation) *Durch* Authentifikation *ist die Herkunft einer Nachricht eindeutig zuzuordnen. Man unterscheidet zwischen* Benutzerauthentifikation *und der* Geräteauthentifikation*. Bei der Geräteauthentifikation wird lediglich geprüft, ob die Nachricht von einem Gerät stammt, welches sich vorher beim Empfänger angemeldet hat. Die Benutzerauthentifikation geht einen Schritt weiter, hier wird zusätzlich gefordert, die Person, die die Übertragung durchführt, eindeutig zu verifizieren.*

Klassische Sicherheitsprotokolle im Bereich der Netzsicherheit fokussieren immer auf die Geräteauthentifikation. Für die Benutzerauthentifikation werden digitale Zertifikate benötigt, mit dem der Empfänger den Sender eindeutig identifizieren kann. Die Verwendung von Zertifikaten ist im Secure-SCTP-Standard nicht vorgesehen. Für den weiteren Verlauf der vorliegenden Arbeit wird der Begriff Authentifikation immer im Zusammenhang mit der Geräteauthentifikation gebraucht, die vom Secure-SCTP unterstützt wird.

Über die Authentifikation wird häufig auch bereits die Forderung nach *Integrität* erfüllt.

Definition 11.2.3 (Integrität) *Durch Sicherstellung der* Integrität *einer Nachricht kann sichergestellt werden, dass die Nachricht während des Transports nicht verändert wurde, also auch tatsächlich der Nachricht entspricht, die der Sender in das Netz eingespeist hat.*

Aufgrund der fehlenden Benutzerauthentifikation kann die Forderung nach Verbindlichkeit von Secure-SCTP nicht erfüllt werden.

Definition 11.2.4 (Verbindlichkeit) *Durch die Forderung der* Verbindlichkeit *einer Nachricht kann eine nicht zu leugnende Abhängigkeit von Sender und Nachricht hergestellt werden.*

Die wichtigste aller Forderungen ist die Forderung nach der Geheimhaltung, also nach dem verschlüsselten Versand der Nachrichten.

Definition 11.2.5 (Geheimhaltung) *Durch die Forderung der* Geheimhaltung *wird sichergestellt, dass nur der Sender und der Empfänger Informationen über den Inhalt der Nachricht erhält, ein Außenstehender erhält dagegen keine Information über die gesendete Nachricht.*

Für die Geheimhaltung gibt es in der Literatur weitere Differenzierungen, je nachdem in welchem Zusammenhang das Verschlüsselungsprotokoll eingesetzt wird. Wird die Verschlüsselung beispielsweise als eingebettetes Subprotokoll verwendet, so ist es möglicherweise nicht ausreichend, den Begriff der Information, den ein möglicher Angreifer erhält, auf den eigentlichen Inhalt der Nachricht zu beziehen. Als Beispiel sei ein Wahl-Protokoll angeführt, welches eine geheime Wahl über das Internet ermöglichen soll. Da die möglichen „Ergebnisse“ zahlenmäßig sehr eingeschränkt sind, da nur eine begrenzte Anzahl an Parteien zur Wahl stehen, darf das Verschlüsselungsverfahren selber keine zusätzlichen eindeutig reproduzierbaren Merkmale bei der Verschlüsselung generieren. In einem solchen Fall wird häufig die *semantische Sicherheit* des Verfahrens gefordert. Eine Übersicht über semantische Verschlüsselungsverfahren im Allgemeinen und zur Konstruktion von sicheren Protokollen ist in [Kamphenkel und Kamphenkel 2002] dargestellt. Von Secure-SCTP wird lediglich die Geheimhaltung des Nachrichteninhalts im klassischen Sinne gefordert.

Secure-SCTP bietet einen flexiblen Einsatz der Sicherheitsanforderungen. Über sogenannte Sicherheits-Levels kann der Grad der gewünschten Sicherheit gezielt eingestellt werden. Wird lediglich die Authentifizierung der Informationen gefordert, sodass auf den Verschlüsselungspart verzichtet werden kann. Häufig hat dies Performance-Gründe, da eine zusätzliche Verschlüsselung möglicherweise die Datenmenge vergrößert, die über das Netz übertragen werden muss. Weiterhin wird für den Ablauf der Verschlüsselungs- und Entschlüsselungsalgorihtmen Zeit benötigt, die in einem zeitkritischen Ablauf ggf. nicht in Kauf genommen werden können. Eine gezielte Betrachtung der Performance-Probleme erfolgt im Abschnitt 12.

Definition 11.2.6 (Security-Level) *Der* Security-Level *oder die* Sicherheitsstufe *entspricht einer Sicherheitsklassifikation für eine Assoziation und damit für die SCTP-Pakete bzw. für die zu schützenden Chunks.*

Der Security-Level legt eine Sicherheitsklassifikation fest, damit wird aber noch nicht auf die verwendeten Verfahren oder gar auf Schlüssellängen bzw. das zugehörige Schlüsselmanagement abgestellt, sondern grundsätzlich festgelegt, wie die Daten einer Assoziation nach Aufbau einer Secure-Session behandelt werden sollen. Secure-SCTP unterscheidet vier verschiedene Security-Level, wobei der *Security-Level 0* keine Sicherheit bietet und nur die Kompatibilität zum Basis-SCTP herstellt.

Den umfassendsten Schutz, aber auch die geringste Flexibilität bietet der *Security-Level 3*. Der Security-Level 3 bezieht sich immer auf eine gesamte Assoziation, somit auch auf alle ausgehandelten Streams. Es wird die gesamte Assoziation vollständig geschützt, d.h. es wird die Authentifikation sämtlicher SCTP-Patkete und die Verschlüsselung sämtlicher Daten- und Controllchunks durchgeführt. Bei diesem Security-Level werden demnach alle Daten gleichwertig behandelt, eine gezielte flexible Behandlung einzelner Chunks, Streams oder Pakete ist nicht vorgesehen. Der Security-Level 3 eignet sich für

Anwendungen, bei denen die Sicherheit im Mittelpunkt steht und zwingend sämtliche Daten auf gleich hohem Niveau behandelt werden müssen. Zusätzlich sollten die Kapazitäten der einzelnen Endpunkte bzw. des Netzes ausreichend sein, um alle Daten performant zu übertragen, da im Grundsatz eine einmalig gewählte Sicherheitsstufe bis zum Abbau der Verbindung beibehalten wird. Es werden für sämtliche zu übertragenden Daten die Anforderungen *Authentifikation*, *Geheimhaltung* und *Integrität* erfüllt.

Der *Security-Level 1* sieht nur die *Authentifikation* und die *Integrität* der Daten auf SCTP-Paketebene vor. Daher eignet sich dieser Security-Level nur bedingt für die sichere Übertragung, da alle Daten einer Assoziation unverschlüsselt übertragen werden. Falls eine Anwendung mit öffentlichen Daten operiert, die von Außenstehenden ohne Weiteres gelesen und eingesehen werden dürfen, aber sichergestellt sein soll, dass ein bestimmter Sender die Übertragung durchgeführt hat, kann diese extrem schwache Form der kryptographischen Behandlung der Informationen gewählt werden. Der Einsatz eines solchen Security-Levels ist aus Sicht der Sicherheit sehr fragwürdig, da durch den Aufbau einer Secure-Session dem Anwender eine gesicherte Übertragung suggeriert wird, die aber de-facto nicht vorliegt. Eine solche Sicherheitsstufe sollte nur in Ausnahmefällen und nur dann angewendet werden, wenn technische Gründe wie beispielsweise mangelnde Performance oder fehlende Netzkapazität den Verzicht auf die Geheimhaltung notwendig machen. Für eine flexible und sichere Datenübertragung ist die Sicherheitsstufe 2 vorzuziehen.

Der *Security-Level 2* macht die Stärke von Secure-SCTP aus, da neben der Integrität und Authentifikation der Daten eine gezielte Verschlüsselung durch die Sender-Applikation möglich ist. Es werden nicht grundsätzlich alle Daten-Chunks gleich behandelt, sonder es erfolgt eine gezielte kryptographische Behandlung einzelner Daten-Chunks. Der Vorteil hierbei liegt auf der Hand. Sicherheitsrelevante Daten können so auf einen speziell abgesicherten Stream übertragen werden, während unkritische Daten über einen unverschlüsselten Stream übermittelt werden. Die „gemischte“ Übertragung von verschlüsselten und nicht verschlüsselten Daten innerhalb einer einzigen Assoziation bietet somit eine elegante und effektive Art der flexiblen kryptographischen Behandlung der zu übertragenden Daten. Im Abschnitt 12 wird dieses Konzept dahingehend erweitert, dass nicht nur über den Security-Level die Applikation die Möglichkeit erhält, Einfluss auf die Verschlüsselung einzelner Daten-Chunks zu nehmen, sondern auch der SCTP-Kern in Form des IN kann aktiv in den Prozess der Verschlüsselung eingreifen. Ein solches Vorgehen eröffnet neue und effektive Szenarien bei Datenübertragungen, die besondere Anforderungen nicht nur für die kryptographische Behandlung der Daten, sondern auch besondere Anforderungen für die Art und Weise bzw. die Rahmenbedingungen der Übertragung fordern.

Etablierung einer Secure-Session

Auf eine ausführliche Beschreibung der Initialisierungsprozedur sowie der Schlüsselgenerierung soll an dieser Stelle verzichtet werden, die genaue Vorgehensweise kann u.a.

in [Unurkhaan 2005] nachgelesen werden. Die verwendeten kryptographischen Verfahren sowie die Einbindung in die SCTPLib wird in Abschnitt 12.1 im Zusammenhang mit der verwendeten Testumgebung nachgereicht.

11.3 Sichere Datenübertragung

In diesem Unterabschnitt steht die sichere Datenübertragung mittels Secure-SCTP im Mittelpunkt. Neben neuen Chunks, die für das Aushandeln der Secure-Session Verwendung finden, werden auch für den Datentransfer neue Chunktypen definiert.

(a) Der Authentifikations-Chunk

(b) Der Encrypted-Data-Chunk

Abbildung 11.1: Chunks für den sicheren Datentransfer

11.3.1 Neue Chunktypen für die sichere Übertragung

Sämtliche zusätzliche Informationen werden über Chunks übermittelt, so auch die verschlüsselten Daten-Chunks und der für die Authentifikation benötigte elektronische Fingerabdruck. Für die Authentifikation wird ein HMAC, wie er bereits in Definition 6.2.11 im Zusammenhang mit dem Verbindungsaufbau von SCTP eingeführt wurde, verwendet. Für beide sicherheitsrelevanten Aktionen steht jeweils ein Chunk-Typ zur Verfügung, wie sie in Abbildung 11.1 dargestellt sind.

Der Authentifikations-Chunk

Aus Gründen der Performance wird nicht jeder einzelne Chunk authentifiziert, sondern ein komplettes SCTP-Paket. Dieses Vorgehen hat zudem den Vorteil, dass nicht nur die Chunks, sondern fast alle Felder des SCTP-Pakets vor Veränderung durch einen Angreifer geschützt sind.

Der HMAC wird über einen Authentifikations-Chunk (0x11) übermittelt. Hierbei ist der Zeitpunkt der Generierung des HMACs wichtig, da die Checksumme für das gesamte Paket gelten muss und somit den HMAC mit einschließt. Daher ist die Checksumme der einzige Wert, der nicht von der Authentifizierung erfasst wird. Damit ist die grundsätzliche Vorgehensweise vorgegeben.

Nachdem das SCTP-Paket ohne den Authentifikations-Chunk zusammengestellt und der Kopfbereich hinzugefügt wurde, kann der HMAC für das so weit erstellte Paket berechnet werden. Der berechnete HMAC wird in den vorbereiteten Authentifikations-Chunk (vgl. Abbildung 11.1a) eingetragen. Jetzt kann der Authentifikations-Chunk in das Paket integriert und die Checksumme berechnet werden. Dieser Vorgang kann invers auf Seiten des Empfängers nachvollzogen werden.

Lassen wir den Security-Level 0 außen vor. Da hier Standard-SCTP Verwendung findet, wird in allen „echten“ Secure-Sessions der Authentifikations-Chunk verwendet. Dabei wird der Authentifikations-Chunk immer als letzter Chunk angehängt, sodass dahinter kein weiterer Daten-Chunk folgen kann.

Der Encrypted-Data-Chunk

Kommen wir zum interessantesten neuen Chunk-Typ, nämlich dem Encrypted-Data-Chunk (0x10), der eine verschlüsselte Nachricht aufnehmen kann. Wird Secure-SCTP im Secure-Level 3 gefahren, so kann ein SCTP-Paket nur verschlüsselte Daten-Chunks enthalten, sodass die Assoziation keine klassischen Daten-Chunks mehr verwenden kann.

Anders sieht es beim flexiblen Security-Level 2 aus, bei dem der gesicherte und ungesicherte Datentransfer innerhalb einer Assoziaation möglich ist. Hierbei ist anzumerken, dass immer erst die verschlüsselten Daten-Chunks in dem SCTP-Paket untergebracht werden, gefolgt von den nicht verschlüsselten. Den Abschluss bildet, wie bereits gesagt, der Authentifikations-Chunk.

Da, wie im Folgenden ausführlicher beschrieben, die Verschlüsselug mit unterschiedlichen Verfahren durchgeführt werden kann, die auch auf Blöcken unterschiedlicher Länge agieren, muss durch einen zusätzlichen Mechanismus sichergestellt werden, dass das Paket eine korrekte Größe aufweist. Dies wird durch einen Hilfs-Chunk erreicht, der lediglich zum Auffüllen des verschlüsselten Datenbereichs benötigt wird. Dieser Hilfs-Chunk wird Padding-Chunk (0x12) genannt.

Mit diesen neuen Chunktypen kann eine gesicherte Übertragung erreicht werden. Die softwaretechnische Einbindung in den SCTP-Kern wird in Abschnitt 12.1 im Zusammenhang mit der für die praktischen Versuche verwendeten Implementierung auf Basis der SCTPlib besprochen. In diesem Abschnitt wird auch die Beschreibung der möglichen kryptographischen Algorithmen nachgeholt, da diese eng mit der Implementierung verknüpft sind.

An dieser Stelle sollen die noch nicht erläuterten Einträge in den neuen Chunks kurz beschrieben werden. Alle Sitzungsschlüssel werden ähnlich wie beim SSL-Protokoll von einem *Pre-Master-Secret* abgeleitet. Das Pre-Master-Secret kann nach der Etablierung der Secure-Session aus den ausgetauschten Schlüsselinformationen abgeleitet werden und steht Sender und Empfänger gleichermaßen zur Verfügung. Die Besprechung des SSL-

Protokolls führt in diesem Zusammenhang zu weit, für eine genaue Beschreibung wird daher auf [Schäfer 2003] verwiesen.

Um bei längeren Sitzungen die Sicherheit zu erhöhen, kann durch das sogenannte *Re-Keying* ein neuer Wert für das Pre-Master-Secret ausgehandelt werden. Damit Sender und Empfänger ihre Schlüssel synchron halten können, wird immer eine Referenz auf das verwendete Pre-Master-Secret übermittelt. So können in einer Übergangsphase auch noch auf alter Basis verschlüsselte Daten-Chunks vom Empfänger entschlüsselt werden.

Neben dem Verweis auf das Master-Secret enthält der Encrypted-Data-Chunk eine Zufallszahl. Diese wird zur Generierung eines Initialisierungs-Vektors benötigt, der seinerseits für das Verketten von verschlüsselten Blöcken, dem sogenannten *Cipher-Block-Chaining*, benötigt wird.

Bei der Verwendung von blockorientierten Verschlüsselungsverfahren, der sogenannten *Blockchiffre*, wird die zu verschlüsselnde Nachricht in Blöcke einer für das verwendete Verfahren vorgeschriebenen Größe zerlegt. Jeder so entstandene Block wird unter Verwendung einer Verschlüsselungsroutine verschlüsselt.

Definition 11.3.1 (ECB-Modus) *Wird jeder Block einer Blockchiffre unabhängig voneinander verschlüsselt, so spricht man von der Verschlüsselung in der Betriebsart des* ECB-Modus *oder auch* Electronic-Codebook-Modus.

Es hat sich gezeigt, dass die Verwendung des ECB-Modus Angriffspunkte für die Kryptoanalyse bietet. In [Wätjen 2003] sind hierfür Beispiele ausgeführt. Um diese Probleme zu beseitigen, kann das bereits angedeutete *Cipher-Block-Chaining* verwendet werden.

Definition 11.3.2 (CBC-Modus) *Werden alle Blöcke einer zu verschlüsselnden Nachricht einer Blockchiffre in Abhängigkeit voneinander verschlüsselt, so spricht man von der Verschlüsselung in der Betriebsart des* CBS-Modus *oder auch* Cipher-Block-Chaining-Modus.

Beim CBC-Modus geht der Chiffretext des Vorgängerblocks in die Berechnung des Chiffretextes des aktuellen Blocks ein. Da der erste Block auf keinen Vorgänger zurückgreifen kann, wird ein *Initialisierungs-Vektor* anstelle des Vorgängerblocks verwendet. Dieser wird wie bereits erwähnt über die im Encrypted-Data-Chunk mitgelieferte Zufallszahl bestimmt.

11.3.2 Das Konzept von Secure-SCTP

Im folgenden Unterabschnitt sollen die Besonderheiten von Secure-SCTP insbesondere im Vergleich zu bestehenden Lösungsansätzen zur sicheren Datenübertragung erläutert werden. Da Secure-SCTP speziell auf das Basis-Protokoll zugeschnitten ist, ist man in der Lage, sehr gezielt und flexibel auf die Sicherheitsansprüche der Applikationen einzugehen.

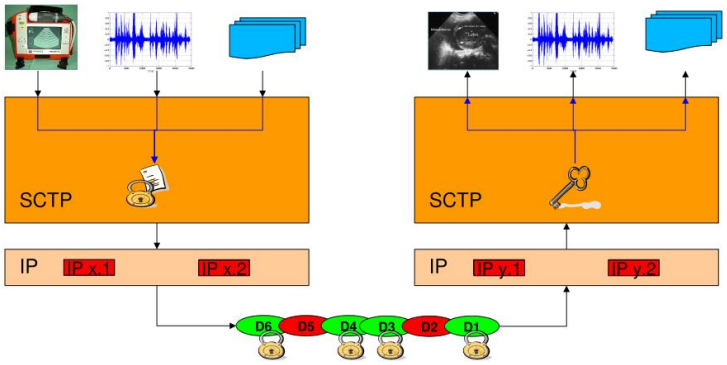


Abbildung 11.2: Konzept von Secure-SCTP

In Abbildung 11.2 ist das grundsätzliche Konzept von Secure-SCTP schematisch dargestellt. Ausgangspunkt ist das im ersten Teil der Arbeit ausführlich beschriebene erweiterte Notfallszenario mit den verschiedenen zu übertragenden Daten. Es sollen Sonographie-, Sprach- und Textdaten übermittelt werden. Jede Datenart soll in einem eigenen SCTP-Stream übertragen werden.

Die Streams einer SCTP-Assoziation können unabhängig voneinander betrachtet werden, sodass jeder Datentyp oder besser gesagt jeder Stream eine eigne kryptographische Behandlung der Daten festlegen kann. Im Beispiel könnte beispielsweise die Sprachinformation ohne zusätzliche Verschlüsselung erfolgen, wohingegen die sensiblen Daten der Sonographie und der Patientenakte sicher übertragen werden sollen.

Der gesicherte und nicht sichere Datentransfer kann in einer einzelnen Assoziation innerhalb einer einzigen Secure-Session erfolgen. In der Abbildung sind beispielsweise die grünen Daten-Chunks gesichert, werden also in Form von Encrypted-Daten-Chunks übermittelt, während die roten Daten-Chunks als Standard-SCTP-Daten-Chunks übertragen werden.

11.4 Alternative Ansätze

Secure-SCTP ist nicht der einzige mögliche Ansatz, um einen sicheren Datentransfer zu erreichen. Da die Sicherheit erst in den letzten Jahren in das Interesse der Anwendungen gerückt ist, sind auch altbewährte Protokolle wie TCP oder UDP von Hause aus nicht in der Lage, sicher Daten zu übertragen. Aus diesem Grund wurden zu einem späteren Zeitpunkt spezielle Protokolle für die Realisierung der Sicherheit entwickelt, die aber nicht in die entsprechenden Protokolle integriert werden, sondern als eigenständige Protokolle den Datentransfer absichern.

Hierbei haben sich zwei Ansätze etabliert, die mit Einschränkungen auch für SCTP ge-

nutzt werden können. Mit Einschränkungen, da die zusätzlichen Features von SCTP, namentlich Multi-Homing und Multi-Streaming, beim Design in der Form noch nicht existierten und daher auch nicht berücksichtigt werden konnten.

Netzwerkprotokolle werden üblicherweise einer Schicht eines Schichtenmodells zugeordnet, wobei jede Schicht einen anderen Aspekt der Kommunikation abdeckt. In [Stevens 1998] wird beispielsweise ein vereinfachtes Modell basierend auf vier Schichten, nämlich der Anwendungsschicht, der Transportschicht, der Netzwerkschicht und der „Link"-Schicht, vorgestellt. SCTP selber wird in der Transportschicht geführt, während das IP-Protokoll der Netzwerkschicht zugeordnet wird. Der wesentliche Unterschied der „aufgesetzten" Sichherheitslösungen liegt in der Schicht, die für die Sicherheit zuständig ist. Mit Secure-SCTP wird die Sicherheit direkt im SCTP-Kern abgebildet, sodass hier die Transport-Schicht für die Sicherheit verantwortlich ist.

11.4.1 IPSec – Sicherheit in der Netzwerkschicht

Ein möglicher alternativer Ansatz ist durch IPSec gegeben, wobei die Sicherheit in der Netzwerkschicht realisert wird, indem einzelne IP-Pakete geschützt werden. Eine mögliche Definition wird in [Doraswamy und D. 2000] gegeben.

Definition 11.4.1 (IPSec) *Bei* Internet-Protocol-Security *oder kurz* IPSec *handelt es sich um einen Sicherheitsstandard für das Internet, das Intranet und virtuelle private Netzwerke. Die Basisarchitektur ist in [Kent und Atkinson 1998] spezifiziert.*

Durch [Kent und Atkinson 1998] wird lediglich eine Basisarchitektur vorgestellt, auf der alle konkreten Implementierungen aufsetzen. Im Wesentlichen stellt IPSec Protokolle zur Authentifizierung und Geheimhaltung bereit. Mit dem Protokoll *Authentication Header-Protokoll* (AH) können IP-Pakete authentifiziert werden, mit dem Protokoll *Encapsulation Security-Payload* (ESP) wird die Geheimhaltung sichergestellt.

Ein durch IPSec geschütztes Datagrammm ist grundsätzlich eine andere Art von IP-Paket, sodass es möglich ist, verschiedene Sicherheitsdienste zu verschachteln. Der Vorteil liegt darin, dass zusätzlich zum Schutz der Pakete, über öffentliche Netze, auch gezielt der Transfer im privaten Netzwerk geschützt werden kann. Das klassische Beispiel ist eine mit ESP geschützte Verbindung zwischen den Endpunkten des öffentlichen Netzes und einer Authentifizierung des einzelnen Rechners im privaten Netz durch eine zusätzlich eingeschachtelte AH-Absicherung bzw. eine weitere ESP-Absicherung.

IPSec wird hauptsächlich dann eingesetzt, wenn es gilt, die maximal mögliche Sicherheit für alle übertragenen Daten inklusive der verwendeten Kontrolldaten zu gewährleisten. Die größte Akzeptanz erfährt IPSec bei der Konstruktion von sogenannten *virtuellen privaten Netzwerken*. Hier hat IPSec sich als Standard herauskristallisiert.

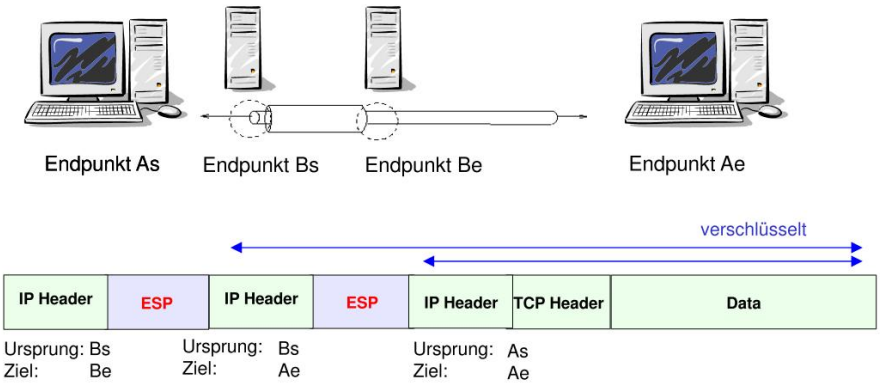


Abbildung 11.3: Aufbau eines virtuellen privaten Netzwerks mit IPSec

Definition 11.4.2 (VPN) *Durch ein* virtuelles privates Netzwerk *(VPN) wird eine sichere Verbindung zwischen sicheren lokalen Netzen über ein unsicheres Netz, normalerweise das Internet, hergestellt.* Privat *heißt, dass die Verbindung zwischen den Netzen als genauso sicher anzusehen ist wie in einem privaten und somit sicheren Netz. Unter* virtuell *versteht man die Betrachtung von räumlich voneinander getrennten ggf. öffentlichen Netzen als virtuelles Gesamtnetz.*

Durch ein VPN können die Kommunikationspartner auf dem Weg durch das unsichere Netz verborgen werden. Für den Zugriff von „außen“ in ein besonders gesichertes Firmennetzwerk, sicher eine grundlegende Forderung, die durch einfache Verschlüsselung, wie sie von Secure-SCTP angeboten wird nicht erfüllt werden kann. Die Forderung nach maximaler Sicherheit und Abbildung von virtuellen privaten Netzwerken birgt aber auch Probleme. Dies wird deutlich, wenn man sich ansieht, wie ein VPN praktisch realisiert wird.

Definition 11.4.3 (Tunneln) *Unter* Tunneln *versteht man ein Verfahren, bei dem Datenpakete eines Protokolls mit Hilfe eines anderen Protokolls übertragen werden.*

Für einen solchen *Tunnel* werden IP-Pakete in IP-Paketen gekapselt. In Abbildung 11.3 ist die Basisstruktur eines VPNs abgebildet. Man erkennt, dass sowohl der innere Tunnel als auch der äußere Tunnel verschlüsselt und authentifiziert wird. Zudem werden drei IP-Pakete und zwei IPSec-Protokoll-Pakete zu einem IP-Paket gebundelt. Somit wird ein sehr großer Overhead erzeugt, der zu einem erheblich höheren Traffic auf der Leitung führt. Zudem ist durch die mehrfache Verschlüsselung mit einem zeitlichen Mehraufwand zu rechnen, insbesondere da der innere Tunnel nicht bei den Endpunkten, im Beispiel der Sender As und der Empfänger Ae, sondern auf Seiten der inneren Endpunkte Be und Bs erfolgt.

11.4.2 SCTP und IPSec

Nachdem die grundsätzliche Struktur von IPSec dargestellt wurde, soll jetzt geprüft werden, inwieweit anstelle von TCP alternative Protokoll SCTP zum Aufbau eines VPNs genutzt werden kann. Mit [Bellovin et al. 2003] liegt eine ausführliche Aufbereitung dieser Thematik vor.

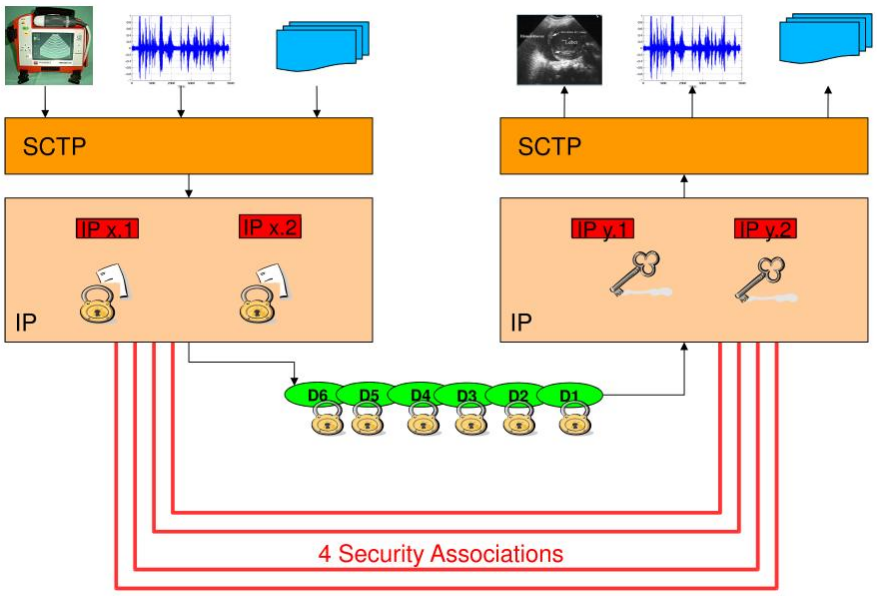


Abbildung 11.4: SCTP über IPSec

In Abbildung 11.4 ist die grundsätzliche Struktur einer mit IPSec geschützten SCTP-Verbindung dargestellt. Man erkennt, dass IPSec bei Verzicht der Multi-Homing-Eigenschaft von SCTP direkt ohne Änderungen angewendet werden kann. Die Vergrößerung des Traffics aufgrund der zusätzlichen Paket-Strukturelemente ist IPSec-inhärent und somit auch bei der Verwendung von SCTP anstelle von TCP hinzunehmen.

Um die Probleme, die durch das Multi-Homing von SCTP bei der Verwendung von IPSec entstehen, zu verdeutlichen, wird die Definition der Security-Assoziation benötigt:

Definition 11.4.4 (SA) *Unter einer* Security-Assoziation *(SA) versteht man einen Vertrag zwischen den kommunizierenden Endpunkten, der sämtliche relevanten Sicherheitsfeatures festlegt.*

Ein grundsätzliches Sicherheitsfeature im Sinne der Definition 11.4.4 ist das zu verwendende Protokoll, bei IPSec stehen derzeit das ESP und das AH zur Auswahl, sowie die zu verwendenden Schlüssel etc. Eine SA ist nicht bidirektional, sodass jeweils für die Hin- und Rückrichtung eine SA aufgebaut werden muss. Da eine IPSec-SA basierend auf einer

Adresse angelegt wird, kann die Möglichkeit, einem Endpunkt mehrere Adressen zuzuordnen, nicht direkt umgesetzt werden. Als Alternative wird in [Bellovin et al. 2003] vorgeschlagen, für jede mögliche Kombination eine eigenständige SA aufzubauen. Schon bei relativ einfachen Modellen wird die Anzahl der notwendigen SAs sehr groß, wodurch auch übermäßig viel Speicher zur Hinterlegung der Daten alloziert werden muss, von der Zeit, die für den Verbindungsaufbau benötigt wird, ganz zu schweigen.

Zusammenfassend kann festgehalten werden, dass die Kombination von SCTP und IPSec nur dann sinnvoll eingesetzt werden kann, wenn auf das Multi-Homing verzichtet wird sowie eine maximale Absicherung der Übertragung gewährleistet werden muss.

11.4.3 TLS und Datagramm-TLS

Die Absicherung einer Verbindung unter Verwendung von *TLS (Transport Layer Security)* bzw. der ursprünglichen Variante *SSL (Secure Socket Layer)* wurde primär zur Absicherung von HTTP-Sitzungen entworfen, da aufgrund der zunehmend kommerziellen Nutzung des Internets sich ein entsprechender Bedarf herauskristallisiert hat. Das TLS-Protokoll wird der Transportschicht zugeordnet, liegt aber oberhalb des verwendeten Transportprotokolls und damit auch oberhalb von Secure-SCTP. Die Verwendung von TLS in Verbindung mit SCTP wurde in [Jungmaier et al. 2002] spezifiziert.

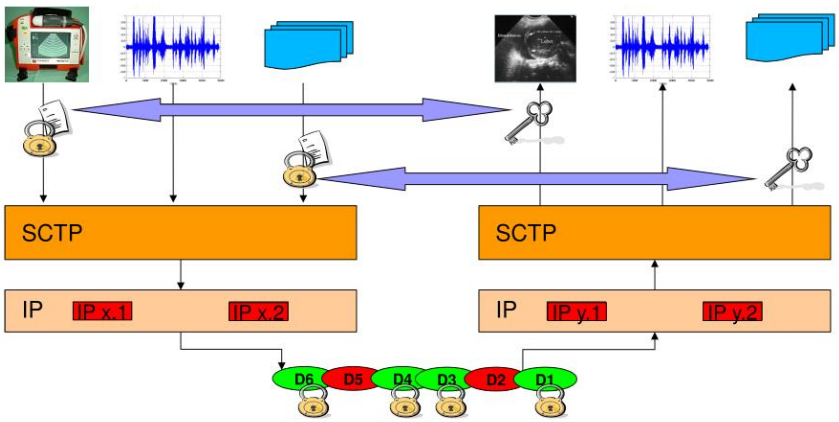


Abbildung 11.5: Absicherung einzelner Streams durch TLS

Die grundsätzliche Absicherung einer SCTP-Assoziation durch TLS ist in Abbildung 11.5 skizziert. Da TLS als Schnittstelle zu einer Anwendung auf Basis von TCP entwickelt wurde, müssen die einzelnen Anwendungen für die Verwendung von SCTP erst angepasst werden. Ist eine solche Anpassung erstmal erfolgt, kann – zumindest ein zuverlässiger Transfer – sehr elegant abgesichert werden.

TLS selber setzt sich aus verschiedenen Teilprotokollen zusammen, die spezielle Aufgaben der Absicherung übernehmen. Der zu verschlüsselnde Datenblock wird als *Record* bezeichnet.

Definition 11.4.5 (Records) *Die Benutzerdaten werden in Blöcke zerlegt, die im Sprachgebrauch von TLS als* Records *bezeichnet werden, um eine Blockchiffre optimal anwenden zu können.*

Für den Aufbau einer sicheren Verbindung sind neben dem *Record-Layer-Protokoll*, das die eigentliche Verschlüsselung, Authentifizierung und Fragmentierung bzw. Defragmentierung durchführt, auch das *Change-Cipher-Protokoll* für die Erneuerung der Sitzungsschlüssel und das *Handshake-Protokoll* zum Aufbau einer Verbindung einschließlich der Aushandlung der Sicherheitsparameter von Interesse. Beim Aufbau der Verbindung werden bereits die notwendigen Informationen, wie die zu verwendenden kryptographischen Verfahren und die zu verwendenden Schlüssel, ausgehandelt. Nach erfolgreichem Verbindungsaufbau steht eine *sichere Session* (Secure-Session) ähnlich wie bei Secure-SCTP zur Verfügung.

Der Vorteil von TLS über SCTP liegt darin, dass eine sichere Session auf Streamebene aufgebaut werden kann. Das Beispiel in Abbildung 11.5 veranschaulicht die Situation nach Aufbau einer Assoziation mit drei Teilanwendungen, die jeweils einen eigenen SCTP-Stream für die Übertragung nutzen. Das Beispiel orientiert sich wieder an dem Beispiel des erweiterten Notfallszenarios. Im Beispiel wird für die Übertragung der Sonographiedaten und für die Textdaten eine sichere TLS-Session aufgebaut, während die Sprachdaten ohne kryptographische Behandlung übertragen werden sollen. Ähnlich wie beim Secure-SCTP ist es möglich, in einem einzelnen IP-Paket verschlüsselte und unverschlüsselte Daten-Chunks zu transportieren. TLS lässt allerdings nur einen geordneten Datenstrom zu, da das Fehlen eines Chunks als Sicherheitsproblem erkannt wird. Somit können mit TLS über SCTP nur geordnete Streams zur Anwendung kommen.

Dieses Vorgehen weist zwei grundsätzliche Probleme auf. Zum einen setzt TLS eine zuverlässige Verbindung voraus, zum anderen kann es bei der Verwendung von sehr vielen Streams zu Performanceproblemen kommen. Das gewählte Beispiel in Abbildung 11.5 ist von daher nicht ganz passend, da die Sonodaten, als Videodaten im Normalfall „unzuverlässig“ übertragen werden. Für dieses Problem werden derzeit Lösungen erarbeitet, insbesondere durch die Spezifikation von DTLS, wie es im folgenden Unterabschnitt behandelt wird.

Bei Verwendung von vielen Streams zur gezielten Behandlung der Daten einer Anwendung bzw. einer Teilanwendung hat Secure-SCTP deutliche Vorteile, da hier die Aushandlung der Sicherheitsparameter nur einmal erfolgen muss, da nur eine einzige Secure-Session ausreicht, um die einzelnen Streams abzusichern. Da innerhalb einer Secure-Session von Secure-SCTP die gemischte Übertragung von verschlüsselten und

unverschlüsselten Daten möglich ist, braucht man nicht auf die notwendige Flexibilität zu verzichten. Eine ausführliche Analyse der hier beschriebenen Security-Lösungen für SCTP findet man beispielsweise in [Nordhoff 2006], wobei neben einem ausführlichen Performancevergleich der einzelnen Verfahren und Protokolle auch die Betrachtung der Sicherheit nicht zu kurz kommt.

11.4.4 Datagramm TLS (DTLS)

Aufgrund der weiten Verbreitung und der einfachen Handhabung hat sich TLS als Quasistandard zur Absicherung von Client-Server-Verbindungen, wie sie beim Surfen im Internet mit einen Browser verwendet werden, herauskristallisiert. Allerdings nimmt die Anzahl der multimedialen Daten zu, die häufig unzuverlässig übertragen werden sollen. Ein Protokoll, welches versucht, sich möglichst nah am TLS-Protokoll zu orientieren, aber auch die Möglichkeit bietet, einen unzuverlässigen Datentransfer abzusichern, ist durch *Datagramm-TLS* (DTLS) gegeben.

Datagramm-TLS, wie es in [Rescorla and Modadugu 2006] spezifiziert ist, ist aber nicht direkt für die Verwendung im Zusammenspiel mit SCTP geeignet. Eine mögliche Lösung, SCTP über DTLS abzusichern, wird im Internet-Draft [Tuexen and Seggelmann 2008] gegeben.

Bevor auf eine mögliche Kombination von SCTP mit DTLS eingegangen wird, werden kurz die zwei wesentlichen Unterschiede zwischen DTLS und TLS aufgeführt. So muss bei Verwendung von DTLS die bestehende Abhängigkeit zwischen aufeinanderfolgende Records aufgehoben werden, da es beim unzuverlässigen Versand zum Verlust von „Vorgängerrecords“ kommen kann. Werden mit einer DTLS-Nachricht Kontrolldaten übertragen, so müssen diese auch im unzuverlässigen Versand neu übertragen werden, da diese für die korrekte Abarbeitung der Protokolle benötigt werden.

Die einfachste Variante zur Absicherung einer PR-SCTP-Assoziation besteht in der direkten Anwendung von DTLS auf SCTP. Dies führt aber zu Sicherheitsproblemen, da die Authentifikation der SCTP-Kontrollchunks nicht sichergestellt werden kann. Um dies zu erreichen, wird eine zusätzliche Erweiterung von SCTP benötigt, die die Authentifikation von einzelnen Chunks erlaubt. Mit dem *SCTP-AUTH*-Protokoll liegt eine solche Erweiterung von. Die Verwendung vom klassischen DTLS und der SCTP-AUTH-Erweiterung wird nach [Hohendorf et al. 2006] auch als *Hybrid-Szenario* bezeichnet.

Mit dem Hybrid-Szenario wird aber immer noch nicht der volle Sicherheitsumfang von Secure-SCTP erreicht. Insbesondere erlaubt Secure-SCTP auch die geheime, also verschlüsselte Übertragung von SCTP-Kontrollchunks. Mit Secure-SCTP können sichere und unsichere Streams innerhalb einer Assoziation gemeinsam verwendet werden, im Hybrid-Szenario ist dies nicht möglich. Um dieser Problematik Abhilfe zu schaffen, wurde in [Hohendorf et al. 2006] eine Ereiterung des Hypbrid-Szenarios dargestellt, die als *SCTP aware DTLS* bezeichnet wird.

11.4.5 Zusammenfassung und Bewertung

Grundsätzlich liegt mit Secure-SCTP eine vollständige und performate Lösung zur vollständigen und flexiblen Absicherung einer SCTP-Assoziation vor. Da Secure-SCTP den Standardisierungsprozess noch nicht vollständig durchlaufen hat, kann Secure-SCTP nur bedingt als Lösung für den Echtbetrieb empfohlen werden.

Als einziges Manko an Secure-SCTP kann die fehlende Benutzerauthentifikation angesehen werden, da diese von der Logik her in der Anwenderschicht auf Basis von Zertifikaten angesiedelt werden muss. Muss auch die Benutzerauthentifikation bei einem Projekt berücksichtigt werden, stehen mit TLS bzw. den Varianten von DTLS Alternativen zur Verfügung. Eine vollständige Umsetzung von DTLS für SCTP im Hybrid-Szenario wie auch im *SCTP aware DTLS* stand zum Zeitpunkt der Erstellung der vorliegenden Arbeit leider nicht zur Verfügung.

In der Diplomarbeit [Roll 2008] wurde die Leisstungsfähigkeit der verschiedenen Sicherheitslösungen im praktischen Einsatz untersucht. Für diese Arbeit wurde die bestehende Implementierung von OpenSSL so erweitert, dass zumindest die Grundeigenschaften von DTLS für SCTP angewendet werden konnten. Es konnte gezeigt werden, dass sowohl mit Secure-SCTP als auch mit DTLS unter Verwendung von PR-SCTP eine effiziente Übertragung von Videodaten möglich ist.

Mit IPSec über SCTP sieht es ähnlich aus, allerdings ist bei Verwendung von mehreren „Tunneln“ mit einem großen Overhead zu rechnen, sodass hier die Echtzeitfähigkeit nicht zwingend gegeben ist. Für die vollständige Absicherung beispielsweise eines Firmennetzwerks ist IPSec anderen Lösungen allerdings aufgrund der Möglichkeit zur Erstellung von VPNs überlegen. In [Roll 2008] konnte gezeigt werden, dass SCTP als Ersatz für TCP für den zuverlässigen Versand über IPSec geeignet ist.

Im weiteren Verlauf der vorliegenden Arbeit wird auf Secure-SCTP aufgesetzt, da hier die Kontrolle über die kryptographische Behandlung der Chunks vollstängig in den Händen von SCTP liegt. Dies ist eine grundsätzliche Forderung, wenn das Transportprotokoll, hier SCTP, auf die kryptoraphische Behandlung aktiv Einfluss nehmen soll. Genau dies soll durch den Einsatz des IN speziell für den Bereich Sicherheit erreicht werden.

12 Adaptive Verschlüsselung

12.1 Die SCTPLib als Referenzimplementierung

Für die Untersuchung des Zeitverhaltens der verschiedenen Sicherheits-Suiten wurde auf das bereits in Abschnitt 8.3.1 beschriebene *Testtool* zurückgegriffen. Weiterhin stand eine spezielle Variante der SCTPLib zur Verfügung, die, wie im Abschnitt 5.2 beschrieben, anstelle der Originalversion in die Teststellung integriert wurde.

Mit dieser speziellen Variante der SCTPLib wurden zwar nicht sämtliche Funktionalitäten von Secure-SCTP abgedeckt, die wesentlichen Erweiterungen hinsichtlich Verschlüsselung und Authentifizierung konnten aber verwendet werden. Dabei wurde auf eine fest eingestellte Sicherheits-Suite aufgesetzt, da die Funktionalität zum Aufbau einer sicheren Session während der Initialisierungsphase nur rudimentär umgesetzt ist.

Welche Funktionalität zur Verfügung stand und wie diese in die Struktur der SCTPLib integriert wurde, wird zur Vervollständigung der Beschreibung der Testumgebung aus Abschnitt 5.2 kurz zusammengestellt.

12.1.1 Sicherheits-Suiten

Der Begriff Sicherheits-Suite legt die Rahmenbedingungen für die kryptographische Behandlung der Verbindung fest.

Definition 12.1.1 (Sicherheits-Suite) *Eine* Sicherheits-Suite *oder* Cipher-Suite *definiert mögliche Kombinationen der notwendigen Algorithmen zur Absicherung der Verbindung. Dies sind im Einzelnen*

- *ein Schlüsselaustauschalgorithmus,*
- *ein Authentifikationsalgorithmus,*
- *ein Verschlüsselungsalgorithmus mit Angabe der zu verwendenden Schlüssellänge*
- *und eine Hashfunktion.*

Vor Etablierung einer sicheren Session müssen diese Parameter über Kontroll-Chunks zwischen den Endpunkten ausgetauscht werden. Secure-SCTP stellt hierfür neue Chunktypen zur Verfügung. Da die vorliegende Implementierung diesen Bereich nur unvollständig abgedeckt hat, wird für eine präzise Definition und Beschreibung der einzelnen

neuen Chunk-Typen und deren konkrete Verwendung auf die Arbeit von Unurkhaan [Unurkhaan 2005] verwiesen. Mit der SSOpReq-Meldung wird vom Sender bei der Initialisierung der sicheren Session eine Liste aller Sicherheits-Suiten, die ihm zur Verfügung stehen, an den Empfänger übermittelt. Der Empfänger seinerseits antwortet mit einer SSoReq_ACK-Meldung, mit der er die von ihm selektierte und für die Session zu verwendende Sicherheits-Suite dem Sender mitteilt.

Da die verwendete Implementierung sich die OpenSSL zu Nutze macht, stehen alle Suiten zur Verfügung, die von der OpenSSL-Implementierung angeboten werden. In [Unurkhaan 2005] werden folgende Algorithmen empfohlen.

Für den Schlüsselaustausch bzw. die Schlüsselgenerierung werden Public-Key-Verfahren eingesetzt, da hier ein Austausch über unsichere Leitungen erfolgen soll. Nachdem der Schlüssel generiert wurde, erfolgt die eigentliche Verschlüsselung mit symmetrischen Verfahren auf Basis des erzeugten Schlüsselmaterials. Neben dem klassischen *Diffie-Hellman-Schlüsselaustausch*, der speziell für die Schlüsselgenerierung entworfen wurde, kann auch die Übertragung mittels *RSA* erfolgen.

Für die eigentliche Verschlüsselung wird neben dem nicht mehr als sicher anzusehenden *DES* auch der *Tripel-DES* als Alternative angeboten. In der Zeit von 1976 bis 2000 wurde der DES offizieller Standard für die US-Regierung geführt, was zu einer weltweiten Verbreitung geführt hat. Da der DES lediglich mit einer Schlüssellänge von 56-Bit arbeitet und die Rechnerleistung Ende des letzten Jahrhunderts deutlich angestiegen ist, konnte der DES durch sogenannte *Brute-Force-Angriffe*, also das Ausprobieren sämtlicher möglichen Varianten, gebrochen werden. Eine ausführliche Historie der erfolgreichen Angriffe auf den DES ist u.a. in [Schneier 1996] im Kapitel 12.7 „Wie sicher ist DES heutzutage?“ nachzulesen.

Als Folge dieser Entwicklung wurde versucht, die Schlüssellänge durch Mehrfachanwendung des DES beim Verschlüsselungsvorgang zu erhöhen. Als Ergebnis dieser Bemühungen ist der Tripel-DES hervorgegangen. Beim Tripel-DES werden zwei 56-Bit Schlüssel in drei DES-Durchläufen verwendet. So kommt man auf eine Schlüssellänge von 112-Bit.

Die Mehrfachanwendung des DES stellt aber keine zukunftssichere Methode dar, sodass das *NIST* (National Institute of Standards and Technology) zu einem Wettbewerb zum Kreieren eines neuen Sicherheitsstandards aufgerufen hat. In diesem Wettbewerb stellten namhafte Firmen und Institute ihre Entwicklungen vor, wobei nach ausgiebigen Prüfungen und Sicherheitschecks, der Rijndael Algorithmus, zum neuen Standard unter dem Namen *Advanced Encryption Standard* oder kurz *AES* erkoren wurde. Die Geschichte und die Implementierungs-Details des von Joan Daemen und Vincent Rijmen aus Belgien entworfenen Algorithmus kann u.a. in [Daemen und Rijmen 2002] nachgelesen werden. Der Auswahlprozess wurde bereits im Januar 1997 ausgerufen, die eigentliche Festlegung erfolgte im Oktober 2000. Man erkennt an der langen Testphase, dass hier eine Entscheidung für einen Algorithmus getroffen werden sollte, der für einen längeren

Zeitraum als Standard fungieren kann.

Der AES kann mit Schlüsseln verschiedener Länge betrieben werden. Für den Einsatz im Secure-SCTP sind Schlüssellängen von 128, 192 und 256-Bit vorgesehen. Da der AES sich mittlerweile auch in der Praxis als Standard für die symmetrische Verschlüsselung herauskristallisiert hat, wird in den praktischen Testreihen des folgenden Abschnitts 12.2 nur der AES verwendet. Der DES-Algorithmus in den verschiedenen Varianten ist lediglich aus Gründen der Abwärtskompatibilität in die Spezifikation von Secure-SCTP aufgenommen wurden.

Als Basis für den zu verwendenden HMAC, wie er bereits in Definition 6.2.11 festgelegt wurde, wird ein Hashverfahren als Basis benötigt. Hierfür stehen die bereits im Abschnitt 6.2.1 beschriebenen Hashfunktionen SHA-1 und MD5 zur Auswahl. In der Spezifikation von SecureSCTP steht für jede mögliche Suite eine Konstante zur Verfügung, so wird beispielsweise durch den Wert DH_with_AES_128_CBC_SHA-1 eine Verschlüsselung unter Verwendung eines 128-Bitlangen Schlüssels mit dem AES beschrieben, wobei der Schlüssel über das Diffie-Hellmann-Verfahren generiert wird und SHA-1 als Hashfunktion zum Einsatz kommt.

12.1.2 Erweiterung der SCTP-Architektur

Betrachtet man die in in Abschnitt 6 beschriebene Architektur von SCTP, so kann man grob zwei funktionale Blöcke zur Beschreibung von SCTP erkennen. Dies ist zum einen ein Steuerblock mit dem SCTP-Controller sowie dem Bereich des Pfad-Managements und zum anderen der Block des Datenpfades mit den Funktionalitäten des Bundlings und der Stau- und Flusskontrolle. Beide funktionalen Blöcke werden von Secure-SCTP erweitert.

Als neue Steuerkomponente führt Secure-SCTP einen Crypto-Controller ein. In der SCTPLib-Implementierung von Secure-SCTP werden im Modul *cryptoctrl.c* sämtliche kryptographisch relevanten Funktionen zusammengefasst. So wird über die Funktionen *decrypt_chunk* und *encrypt_chunk* ein einzelner Chunk verschlüsselt bzw. entschlüsselt. Enstsprechende Funktionen stehen für die Authentifikation zur Verfügung. Der Zugriff auf die OpenSSL-Bibliotheken erfolgt daher geschlossen über den Crypto-Controller.

Den Aufruf der Funktionalität des Crypto-Controlers wird in den Block des Daten-Pfads ausgelagert. Das Modul für die Flusskontrolle *flowcontrol.c* prüft über eine spezielle Funktion – *fc_check_for_txmit* – ob die aktuelle Netzsituation den Versand des anliegenden Daten-Chunks erlaubt, ist dies der Fall, wird das *Bundling* angestoßen, welches für die Generierung der für den Versand benötigten SCTP-Pakete sorgt. Hier setzt Secure-SCTP an, indem das zugehörige Modul *sbundling.c* erweitert wird.

Beim eigentlichen Bundling wird jeder Chunk entsprechend der vorher ausgehandelten Sicherheitsstufe verschlüsselt. Die Authentifikation kann erst erfolgen, wenn das SCTP-

Paket vollständig generiert wurde, da die Authentifikation nicht nur für einen einzelnen Chunk durchgeführt wird. Beim Empfänger erfolgt der gegenteilige Prozess, indem erst die Authentifikation der Chunks überprüft wird und beim Unbundling durch die Funktion *rbu_rcvDatagram* wieder entschlüsselt wird. Da ein Daten-Chunk, falls er verschlüsselt übertragen wird, als Encryted-DATA-Chunk gekennzeichnet ist, kann er ohne zusätzlichen Aufwand als verschlüsselter Daten-Chunk erkannt werden.

Durch diese wenigen Eingriffe in die Architektur von SCTP ist es bereits möglich, alle geforderten Sicherheitsanforderungen abzudecken.

12.2 Verzögerung und Durchsatz-Manko durch Verschlüsselung

Betrachtet man den Durchsatz bei verschlüsselter Übertragung, so wird dieser lediglich durch die zusätzlichen Daten negativ beeinflusst. Dies sind insbesondere der Authentifikations-Chunk und das zusätzliche Padding. Es kann aber auch zu einer erhöhten Datenlast durch zusätzliches Fragmentieren der Daten kommen. In diesem Abschnitt werden die Ergebnisse aus [Kamphenkel et al. 2008] zusammengestellt, in der ein Modell vorgestellt wird, mit dem man diesen Durchsatz-Manko (engl. throughput penalty) in Beziehung zur verwendeten Sicherheits-Suite stellen kann.

Theoretischer Mehraufwand einer sicheren Session

Im Vergleich zu anderen Sicherheitslösungen für SCTP ist bei Secure-SCTP die Sicherheit durch einen geringen Mehraufwand zu erreichen. Da der Begriff der sicheren Session, wie er in Abschnitt 11.2.1 eingeführt wurde, sich immer auf eine Assoziation bezieht, können die ausgetauschten Parameter streamübergreifend genutzt werden. Innerhalb der Assoziation kann die Verwaltung und Anwendung der einzelnen Sicherheitsmaßnahmen gezielt und die Ressourcen schonend durchgeführt werden.

Der Aufbau einer sicheren Session kann mit geringen Aufwand durchgeführt werden. Es werden maximal acht Nachrichten für den Aufbau einer sicheren Session ausgetauscht. Im Vergleich dazu benötigt ein TLS-Handshake dreizehn Nachrichten, wobei hier die Session auf Streamebene aufgebaut wird. Dies hat zur Folge, dass die Verwendung von mehreren Streams zu einem drastischen Anstieg der auszutauschenden Nachrichten führt. Noch größer ist der Aufwand, falls in einem solchen Szenario IPSec zur Absicherung der Assoziation genutzt werden soll. Die Probleme von SCTP in Verbindung mit IPSec wurden bereits in Abschnitt 11.4.1 thematisiert.

Neben der reinen Initialisierung einer sicheren Session wird aus Sicherheitsgründen das zur Verfügung stehende Schlüsselmaterial ausgetauscht bzw. erneuert. Auch hier ist Secure-SCTP so angelegt, dass möglichst wenig zusätzliche Last auf die Leitung gelegt wird, insbesondere, da die gesamte Verwaltung der Schlüssel in einer Assoziation erfolgen kann.

Der zusätzliche Mehraufwand für die Initialisierung bzw. die Schlüsselerneuerung ist bei Betrachtung des Throughput zu vernachlässigen, da hier nur einmal bzw. selten in das System eingegriffen werden muss. Als Manko (engl.: penalty) in Bezug auf den Throughput müssen die zusätzlichen Daten betrachtet werden, die im laufenden Betrieb zusätzlich im Vergleich zum Standard-SCTP übertragen werden müssen.

Neben den zusätzlichen Daten, die auf den zusätzlichen Authentifikations-Chunk zurückzuführen sind, ist das Padding, also das Auffüllen von Datenbereichen auf eine feste Blocklänge, wesentlich für die Berechnung des möglichen Throughputs. Da jedes Verschlüsselungsverfahren mit anderen Blocklängen agieren kann, kann es auch unterschiedlich große Datenbereiche für das Padding geben. Für die Verfahren AES und DES soll die Größe eines möglichen Paddings beipspielhaft dargestellt werden.

Beispiel 12.2.1 (Padding aufgrund der Blockgröße) *Das DES-Verfahren arbeitet intern mit einer Blockgröße von* 64*-Bit. Sollen Daten der Größe* 70*-Bit verschlüsselt werden, wird der Datenbereich in zwei Datenblöcke von* 64*-Bit bzw.* 6*-Bit aufgeteilt. Um den DES auch für den zweiten Block anwenden zu können, wird dieser auf* 64*-Bit aufgefüllt, d.h. in diesem Fall ergibt sich ein Mehraufwand von* 58*-Bit.*

Der AES verwendet unabhängig von der verwendeten Schlüssellänge Blöcke in der Größe von 128*-Bit. Eine Erhöhung der Sicherheit durch die Verwendung von längeren Schlüsseln führt daher zu keiner zusätzlichen Datenlast auf der Leitung. Werden nur sehr kleine Chunks verschlüsselt, hat die Blocklänge den Nachteil, dass ein großer Mehraufwand durch Padding entstehen kann. Wendet man das Zahlenbeispiel des obigen DES-Beispiels für die Verschlüsselung mit dem AES an, so ergibt sich ein einzelner Block von* 70*-Bit mit einem Padding von* 58*-Bit.*

An Beispiel 12.2.1 kann man erkennen, dass der reale Mehraufwand von den zu versendenden Daten abhängig ist und somit eine generelle Beschreibung des zu erwartenden Mehraufwands durch zusaätzliche Verschlüsselung nicht gegeben werden kann. Für bestehende Anwendungen mit kalkulierbaren Längen der zu übertragenden Nachrichten kann der Mehraufwand theoretisch abgeschätzt werden.

Host-Performance als Flaschenhals

Grundsätzlich spielt die CPU-Zeit, die für die Verschlüsselung bzw. Entschlüsselung eines Encrypted-Daten-Chunks verbraucht wird, eine untergeordnete Rolle für den Throughput. Unter der Voraussetzung, dass ein sehr großer Durchsatz aufgrund von Technik möglich ist, hat diese Aussage nicht mehr Bestand. In [Hohendorf et al. 2006] wurde ein spezielles Szenario vorgestellt, das auf 1 Gbps-Ethernet-Links basiert, die durch einen hohen möglichen Datendurchsatz charakterisiert sind, bei dem ein Duchsatz-Manko aufgrund der Laufzeiten der verwendeten kryptographischen Algorithmen nachgewiesen werden konnte.

In einem solchen Szenario stellt die Host-Performance den Flaschenhals dar. Dies hat Auswirkungen auf die Anwendung der kryptographischen Funktionen, die insbesondere bei der Verschlüsselung ein signifikantes Durchsatz-Manko mit sich bringt. In einem solchen Fall ist es von Vorteil, wenn nicht sämtliche Daten einer Assoziation verschlüsselt bzw. mit unterschiedlichen Sicherheistanforderungen verschlüsselt werden können. Dies ist bei Secure-SCTP gegeben, da hier für jeden einzelnen Stream gesondert die Verschlüsselung angegeben werden kann.

Worauf ist dieses Manko zurückzuführen? Wenn die Send-Queue dieses Prozesses voll ist, blockiert der Send-Aufruf und wartet, bis neue Pakete versendet werden können. Während dieser Zeit läuft der Prozess leer. In dem Fall, wenn Bundling nicht mehr genutzt werden kann oder wenn lange Nachrichten fragmentiert werden müssen, fällt der Durchsatz ab.

An dieser Stelle kann das Intelligente-Netz genutzt werden, um dieses Manko auszugleichen. Ein auf Lernverfahren beruhender Algorithmus kann in einem vorgegebenen Rahmen Einfluss auf die Verschlüsselung einzelner Chunks nehmen. Dies geht über die grundsätzliche Möglichkeit von Secure-SCTP hinaus, da jetzt sogar innerhalb eines Streams unterschiedliche Verschlüsselungsalgorithmen bzw. Schlülssellängen verwendet werden können. Somit ist man in der Lage, die freien Kapazitäten des Netzwerks zu nutzen und einen höheren Datendurchsatz zu erreichen.

Welcher Benefit ist von einem solchen Vorgehen zu erwarten? Oder anders ausgedrückt, sind die Unterschiede der Algorithmenlaufzeiten signifikant, sodass die Verwendung einer anderen Schlüssellänge tatsächlich einen höheren Durchsatz ermöglicht? Die Verwendung von kürzeren Schlüsseln birgt immer die Gefahr, dass der Performancegewinn auf Kosten der Sicherheit erzielt wird. Die hier vorgestellten Ergebnisse sind aus [Kamphenkel et al. 2008] entnommen und bilden die Grundlage eines möglichen adaptiven Eingreifens in die Verschlüsselung.

Um einen Eindruck über die Kosten der Verschlüsselung bei Änderung der Schlüssellänge zu erhalten, wurden die realen Zeiten für den Verschlüsselungsvorgang unter verschiedenen Konstellationen ermittelt. In Abbildung 12.1 sind die Werte unter Verwendung eines Standard-PCs aufgetragen.

Für den Vergleich wurde nicht zwischen verschiedenen Verfahren gewechselt, da jedes Verfahren aufgrund des zugrunde liegenden Verschlüsselungsalgorithmus unterschiedliche Laufzeiten erwarten lässt. Beim derzeitigen Standardverfahren AES, welches bereits in Abschnitt sec:suite eingeführt wurde, wurden bereits beim Entwurf neben der Sicherheit Aspekte der Laufzeit und des Speicherbedarfs berücksichtigt. Speziell die Möglichkeit einer Implementierung auf Smartcards, auf denen Speicherressourcen stark begrenzt sind, wurde bei der Evaluierung des Algorithmus einbezogen. Aus diesem Grund wurde durch Wahl verschiedener Schlüssellängen nur der Grad der Sicherheit „variiert“.

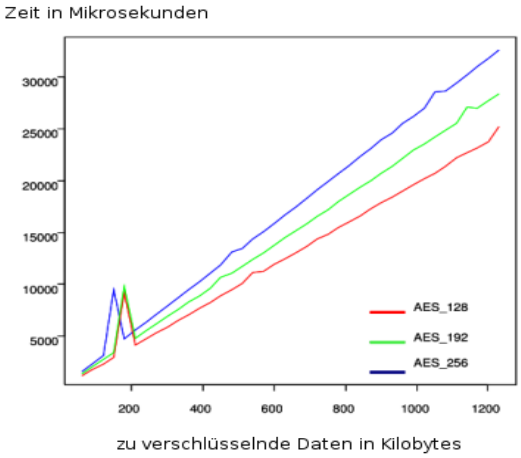


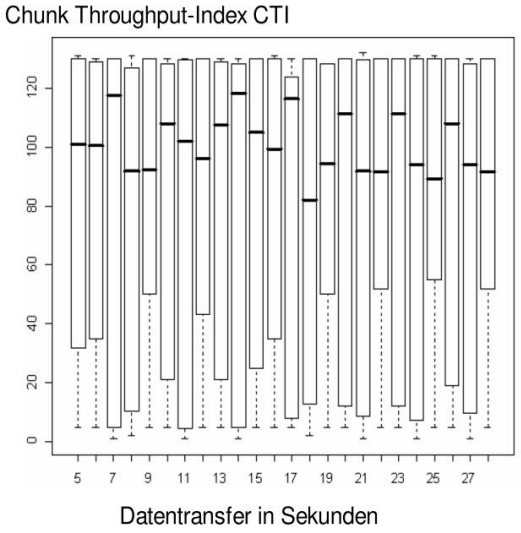
Abbildung 12.1: Penalty durch Verschlüsselung

Bezieht man Abbildung 12.1 in die Betrachtung mit ein, so erkennt man, dass der Aufwand, der für die Ver- bzw. Entschlüsselung benötigt wird, bei höheren Sicherheitsanforderungen deutlich ansteigt. Wenn die zu übertragenden Daten nicht den höchsten Sicherheitsstandard erfordern, kann dieses Potenzial zwischen den verschiedenen Schlüssellängen genutzt werden, um den erkannten Flaschenhals zu entlasten. Hierfür ist es notwendig, dass das IN in der Lage ist, aufgrund der aktuellen Netzsituation zu erkennen, dass mögliche Probleme bei der Übertragung aufgrund der mangelnden Host-Performance, seien sie beim Sender oder beim Empfänger begründet, zurückzuführen sind. Die gewählten adaptiven Verfahren müssen demnach Rückschlüsse auf den Grund des Engpasses auf der Leitung ermöglichen. Die so gewonnene Information kann vom IN genutzt werden, um gezielt in die Datenübertragung einzugreifen, indem die Schlüssellänge bzw. das Verschlüsselungsverfahren auf ein ressourcenschonenderes Verfahren umgestellt wird.

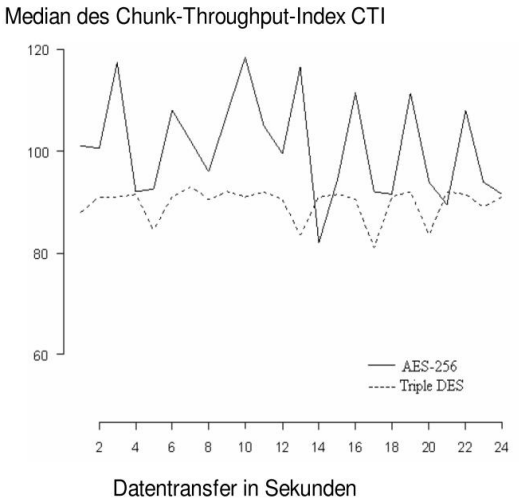
Vergleich verschiedener Verschlüsselungsverfahren und deren Schlüssellänge

Um die verschiedenen Verfahren und Schlüssellängen vergleichen zu können, wird ein einheitliches Maß benötigt. Es wird daher ein *Chunk-Throughput-Index*, kurz CTI, definiert, der für jede Permutation von Verfahren und Schlüssellänge unter gleichen Bedingungen ermittelt wird und als Vergleichsmaß herangezogen werden kann. In Abbildung 12.2a ist der CTI für das AES-Verfahren mit einer Schlüssellänge von 256 Bit exemplarisch in grafischer Form abgebildet.

Für die Messung wurde das Testtool aus Abschnitt 8.3.1 verwendet. Das Testtool übernimmt die Rolle des ULP und übergibt eine feste Menge an Daten pro Sekunde an SCTP. Aufgrund der Verwendung der SCTPLib werden die Daten unter Verwendung der SCTP_SEND-Funktion in das Netz eingespeist. Der Empfänger protokolliert in einem festen Zeitintervall die eingehenden Daten-Chunks bzw. die Größe der übertragenen



(a) CTI des AES-256



(b) Vergleich des Tripel-DES mit AES-256 auf Basis des CTI

Abbildung 12.2: Anwendung des CTI

Daten. Für den CTI in Abbildung 12.2a wurde ein Intervall im Hundertstel-Sekunden-Bereich gewählt. Jeder übertragene Chunk hatte eine feste Größe von 400 Byte, wobei 500 Chunks pro Sekunde an SCTP übergeben wurden.

Als CTI wird die Anzahl der eingehenden Chunks definiert, die innerhalb eines Intervalls beim Empfänger eingehen. Je größer der CTI-Wert, desto größer ist auch der aktuelle Durchsatz durch das Netzwerk. Der CTI-Wert eines einzelnen Intervalls kann keine Aussage liefern, da die Übertragung allein durch die Stau- und Flusskontrolle nicht zwingend gradlinig verläuft. Daher wurde die Messung über eine feste Zeit durchgeführt und ein Mittelwert über die gemessenen CTI-Werte gebildet. Als Mittelwertfunktion wurde der Median gem. Definition 8.5.2 angewendet. Trägt man die Messergebnisse in Form von Boxplots pro Zeiteinheit auf, so ergibt sich die Darstellung in Abbildung 12.2a, wobei die Linie innerhalb der Box den Median darstellt.

Führt man die Messung für alle zur Verfügung stehenden Verfahren aus, kann das theoretische Durchsatz-Manko auf Basis des CTI ermittelt werden. Ein Vergleich zwischen dem Tripel-DES und dem AES-Verfahren mit einer Schlüssellänge von 256-Bit ist in Abbildung 12.2b aufgetragen. Der CTI des AES-256 bewegt sich im Bereich der 100, wogegen der Vergleichswert des Tripel-DES sich im Bereich der 87 bewegt. Die Differenz der Werte kann als Basis für eine adaptive Auswahl des Verschlüsselungsverfahrens durch die IN-Komponente herangezogen werden.

Wann der Wechsel des Verfahrens sinnvoll und notwendig ist, muss anhand der aktuellen

Netzsituation entschieden werden. Hierfür sind entsprechend der Auswertung der Netzsituation für die Pfadwahl im Teil II der vorliegenden Arbeit Untersuchungen über die vorhandenen Parameter und deren Aussage über die HOST-Performance als Flaschenhals notwendig. Diese Untersuchungen sind nicht mehr durchgeführt worden und geben die Möglichkeit zu weiteren wissenschaftlichen Untersuchungen.

Teil IV

Schlussbetrachtung

13 Schlussbetrachtung

13.1 Zusammenfassung

Die Arbeit behandelt die drei wesentlichen Aspekte moderner Telemedizin-Szenarien – Sicherheit, Flexibilität und Echtzeit. Neben den technischen Fragestellungen zur sicheren bzw. performanten Datenübertragung wird explizit auf medizinische Fragestellungen eingegangen.

Im Mittelpunkt der Untersuchungen des ersten Teilbereichs der Arbeit steht die Praxisrelevanz der Telemedizin. Untersucht wurde hier, inwieweit Telemedizin-Szenarien aus medizinischer Sicht benötigt bzw. akzeptiert werden. Insbesondere ist bezüglich der Akzeptanz noch ein großes Verbesserungspotential identifiziert worden. Nicht jedem Arzt bzw. dem medizinischen Personal ist der direkte Nutzen der Telemedizin bewusst. Im Rahmen der Untersuchung wurde die größte Akzeptanz im Rahmen des Notfallszenarios festgestellt, da hier der Zeitgewinn bei der Behandlung von polytraumatisierten Patienten ein entscheidendes Kriterium für die Erhöhung der Überlebenswahrscheinlichkeit des Patienten darstellt.

So konnte gezeigt werden, dass durch die konsequente Nutzung von präklinischer Sonographie in Verbindung mit der Möglichkeit der Expertenkonsultation die Notfallmedizin um einen nicht unwesentlichen Faktor bereichert wird. Darüber hinaus wurde der Einsatz der präklinischen Sonographie in die bestehenden Abläufe bei der Erstversorgung am Unfallort eingebunden.

Ein weiterer Aspekt stellt die sichere Datenübertragung der sensiblen personenbezogenen Daten dar. Hier wurde im dritten Teilbereich der Arbeit untersucht, inwieweit die Verschlüsselung Einfluss auf den Datendurchsatz hat. Es wurde gezeigt, wie durch adaptive Verschlüsselung die Übertragung beeinflusst werden kann.

Technischer Kern der Arbeit ist das im zweiten Teilbereich vorgestellte Intelligente Netzwerk(IN) zur adaptiven Pfadwahl bei Multipfad-Übertragungen. Es wurde ein innovativer Ansatz für die Multipfad-Übertragung als Erweiterung und Ergänzung der konkurrierenden Übertragung mittels CMT-SCTP vorgestellt.

Als Standardmethode zur Pfadwahl in Multipfad-Szenarien wurde bisher nur das klassische Loadsharing beschrieben. Es wurde bisher keine Methode beschrieben bei der die Möglichkeit bestand, aufgrund der Netzparameter Rückschlüsse auf die aktuelle Situation im Netz zu ziehen. Das Intelligente Netz beschriebt eine Methode, wie die von SCTP

bereits im Standard zur Verfügung stehenden Parameter genutzt werden können, um eine Voraussage über die Leistungsfähigkeit einzelner Pfade treffen zu können. Basierend auf diesen Informationen kann eine einfache Auswahlmethode verwendet werden, um die Daten immer bestmöglich auf sämtliche zur Verfügung stehende Pfade zu verteilen.

Zur praktischen Nutzung des IN kann festgehalten werden, dass unter realen Bedingungen – sprich realen umweltabhängigen Kanalkapazitäten– das IN jeweils den besten Datendurchsatz bereitstellt. Hierzu ist kein A-priori Wissen über die Kanalkapazitäten erforderlich, so dass auf unterschiedliche Übertragungsmedien zurückgegriffen werden kann. Bei der Verwendung von konstanten Kanalkapazitäten werden kaum Defizite festgestellt.

13.2 Fazit und Ausblick

Als Fazit der Arbeit kann festgehalten werden, dass die praktische Umsetzung von zeitkritischen und komplexen Telemedizin-Szenarien in naher Zukunft realisiert werden kann. Hierzu hat die Arbeit einen kleinen Beitrag geleistet, indem gezeigt wurde, dass eine bessere Auslastung der bestehenden Netzkapazitäten nicht den Umstieg auf vollständig neue Technologien notwendig macht. Intelligente und innovative Lösungsansätze können genutzt werden, um bestehende Protokolle so zu erweitern, dass die vorhandenen Kapazitäten deutlich effizienter ausgenutzt werden.

Zwar wurde die Untersuchung grundsätzlich in Bezug auf die Nutzung der präklinischen Sonographie durchgeführt, die Ergebnisse lassen sich aber derart verallgemeinern, dass auch weitere Medizin-Szenarien davon profitieren. Sichere Datenübertragung ist ein wesentlicher Baustein moderner Netzwerke. Daher sind die Überlegungen zur sicheren Datenübertragung aus Abschnitt 12 universell.

Die Überarbeitung des medizinischen Workflows für das Notfallszenario in Kapitel 4 ist sehr speziell, zeigt aber einen grundsätzlichen methodischen Ansatz. Bevor in der Medizin ein neues Telemedizin-Szenario eingesetzt werden kann, müssen die Ärzte und das medizinische Personal geschult und angeleitet werden. Dies betrifft nicht nur die technische Abwicklung, sondern gerade auch die medizinisch-algorithmischen Abläufe. Mit der Integration der technischen Abläufe in die medizinischen Algorithmen ist somit der Grundstein gelegt, auch andere Telemedizin-Szenarien in den Fokus der ausführenden Ärzte zu rücken.

Die Ergebnisse der Arbeit sind geeignet, auf andere Telemedizin-Szenarien übertragen zu werden. Ein kleiner Einblick hierzu wurde bereits in Abschnitt 2.1 gegeben, in dem bereits das Potential der vorgestellten Ansätze für die nicht minder anspruchsvollen Szenarien, wie Ferndiagnostik, Telekonsultation, Teleradiologie sowie Telechirurgie angedeutet wurden.

Gerade für Szenarien mit zwingend hohem Datenvolumen ist das Konzept des Intelligenten Netzwerks, wie es in Abschnitt II ausführlich untersucht wurde, von großem Interesse. Durch die Möglichkeit bestehende Netztechnik weiter zu nutzen, wird die Möglichkeit eröffnet, die beschriebenen Szenarien ohne kostenintensive und zeitraubende Neuinstallation im laufenden Betrieb einzubinden.

Index

Literaturverzeichnis

[drf] *Präklinische Sonographie in der Luftrettung*. http://www.drf.de/freiburg.html.

[Allman et al. 1999] Allman, M., V. Paxson und W. Stevens (1999). *TCP Congestion Control*. RFC 2581 (Proposed Standard). Updated by RFC 3390.

[Alpaydin 2008] Alpaydin, Ethern (2008). *Maschinelles Lernen*. Oldenbourg.

[Beck et al. 2002] Beck, A., F. Gebhard und L. Kinzl (2002). *Notärztliche Versorgung des Traumapatienten*. Notfall & Rettungsmedizin, 5(1):57–71.

[Bellovin et al. 2003] Bellovin, S., J. Ioannidis, A. Keromytis und R. Stewart (2003). *On the Use of Stream Control Transmission Protocol (SCTP) with IPsec*. RFC 3554 (Proposed Standard).

[Berger 1998] Berger, R. & Partner (1998). *Telematik im Gesundheitswesen. Studie im Auftrag des Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie und des Bundesministeriums für Gesundheit*.

[Braden 1989] Braden, R. (1989). *Requirements for Internet Hosts - Communication Layers*. (1122). Updated by RFCs 1349, 4379.

[Breiman et al. 1984] Breiman, L., J. Friedman, R. Olshen und C. Stone (1984). *Classification and Regression Trees*. Wadsworth and Brooks.

[Buchholz et al. 1994] Buchholz, S., D. NAst-Kolb, C. Waydhas und P. Betz (1994). *Frühletalität beim Polytrauma. Eine kritische Analyse vermeidbarer Fehler*. Unfallchirurg, 97:285–291.

[Busch 2006] Busch, M. (2006). *Portable ultrasound in pre-hospital emergencies: a feasibility study*. Acta Anaesthesiologica Scandinavica, 6:754–758.

[Clarke et al. 2002] Clarke, JR, S. Trooskin, P. Doshi, L. Greenwald und C. Mode (2002). *Time to Laparotomy for Intra-abdominal Bleeding from Trauma Does Affect Survival for Delays Up to 90 Minutes*. Journal of Trauma-Injury Infection & Critical Care, 52:420–425.

[Cowley 1976] Cowley, RA (1976). *The resuscitation and stabilization of major multiple trauma patients in a trauma center environment*. Clin Med, 83:14–22.

[Culemann et al. 2003] Culemann, U., A. Seekamp, U. Riedel, U. Lehmann, A. Pizanis und T. Pohlemann (2003). *Interdisziplinäres Polytraumamanagement*. Notfall & Rettungsmedizin, 6(8):573–579.

[Daemen und Rijmen 2002] Daemen, J. und V. Rijmen (2002). *The Design of Rijndael AES — The Advanced Encryption Standard*. Springer.

[Donabedian 1966] Donabedian, A. (1966). *Evaluating the quality of medical care. part 2*. Milbank.

[Doraswamy und D. 2000] Doraswamy, N. und H. D. (2000). *IPSec – Der neue Sicherheitsstandard für das Internet, Intranets und virtuelle private Netze*. Addison-Wesley.

[eEurope 2002] eEurope (2002). http://ec.europa.eu/information_society/newsroom/library/referencedoc/eEurope_de.pdf.

[Fayyad et al. 1996] Fayyad, U., G. Piatetsky-Shapiro und P. Smyth (1996). *From data mining to knowledge discovery in databases*. Ai Magazine, 17:37–54.

[Field 1996] Field, M.J. (1996). *Telemedicine: A Guide to Assessing Telecommunications in Health Care*. National Academy Press, Washington, DC.

[Frezza et al. 1999] Frezza, E., R. Solis, R. Silich, R. Spence und M. Martin (1999). *Competency-based instruction to improve the surgical technique and accuracy of the trauma ultrasound*. Am Surg., 65:884–888.

[Gaspari et al. 2005] Gaspari, J. R., J. C. Fox und P. R. Sierzenski (2005). *Emergency Ultrasound: Principles and Practice*. A Mosby Title.

[GMG 2003] GMG (2003). *Gesetz zur Modernisierung der gesetzlichen Krankenversicherungen (GKV-Modernisierungsgesetz – GMG)*. Bundesgesetzblatt Jahrgang 2003 Nr. 55 http://www.bgblportal.de/BGBL/bgbl1f/bgbl103s2190.pdf.

[Grob 1999] Grob, H. L. und Bensberg, F (1999). *Das Data-Mining-Konzept*. Arbeitsbericht Nr. 8. Münster.

[Haas 1997] Haas, N. P. (1997). *Empfehlungen zur Struktur, Organisation und Ausstattung der präklinischen und klinischen Patientenversorgung an Unfallchirurgischen Abteilungen in Krankenhäusern der Bundesrepublik Deutschland*. Der Unfallchirurg, 100(1):2–7.

[Hoffmann et al. 2002] Hoffmann, R, M. Nerlich, M. Muggia-Sullam, T. Pohlemann, B. Wippermann, G. Regel und H. Tscherne (2002). *Blunt abdominal trauma in cases of multiple trauma evaluated by ultrasonography: a prospective analysis of 291 patients*. Journal of Trauma-Injury Infection & Critical Care, 52(3):420–425.

[Hohendorf et al. 2006] Hohendorf, Carsten, E. P. Rathgeb, E. Unurkhaan und M. Tüxen (2006). *Secure End-to-End Transport over SCTP.*. In: Müller, Günter, Hrsg.: *ETRICS*, Bd. 3995 d. Reihe *Lecture Notes in Computer Science*, S. 381–395. Springer.

[Horsch und Handels 2005] Horsch, A und H. Handels (2005). *Telematik im Gesundheitswesen.*. Kapitel in: Lehmann T, Meyer zu Bexten E (Hrsg): Handbuch der Medizinischen Informatik. München, 1:673–712.

[Horsch] Horsch, Alexander. *Ergebnisbericht der Studie ANARAD - Analyse und Konzeption der Teleradiologie in Deutschland*. München.

[Iyengar et al. 2006] Iyengar, Janardhan R., P. D. Amer und R. Stewart (2006). *Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths*. IEEE/ACM Trans. Netw., 14(5):951–964.

[Jacobson 1988] Jacobson, Van (1988). *Congestion Avoidance and Control*. In: *ACM SIGCOMM '88*, S. 314–329, Stanford, CA.

[Jähn 2004] Jähn, Karl (2004). *E-Health*. Springer, Berlin.

[Jungmaier et al. 2002] Jungmaier, A., E. Rescorla und M. Tuexen (2002). *Transport Layer Security over Stream Control Transmission Protocol*. RFC 3436 (Proposed Standard).

[Jungmaier 2005] Jungmaier, Andreas (2005). *Das Transportprotokoll SCTP*. Doktorarbeit, Universität Duisburg-Essen.

[Jungmaier und Rathgeb 2005] Jungmaier, Andreas und E. P. Rathgeb (2005). *A Novel Method for SCTP Load Sharing.*. In: Boutaba, Raouf, K. C. Almeroth, R. Puigjaner, S. X. Shen und J. P. Black, Hrsg.: *NETWORKING*, Bd. 3462 d. Reihe *Lecture Notes in Computer Science*, S. 1453–1456. Springer.

[Jungmaier und Rathgeb 2006] Jungmaier, Andreas und E. P. Rathgeb (2006). *On SCTP multi-homing performance*. Telecommunication Systems, 31(2-3):141–161.

[Kamphenkel K. et al. 2007] Kamphenkel K., Bauer J., Blank M. und Carle G. (2007). *Secure Transmissionof Pre-clinical Ultrasound Video Data at the Scene of a Mass Casualty Incident*. In Proceedings ISCRAM 2007, S. 377–383.

[Kamphenkel et al. 2008] Kamphenkel, K., M. Blank, J. Bauer und G. Carle (2008). *Adaptive encryption for the realization of real-time transmission of sensitive medical video streams*. In: *Proc. International Symposium on a World of Wireless, Mobile and Multimedia Networks WoWMoM 2008*, S. 1–6.

[Kamphenkel und Kamphenkel 2002] Kamphenkel, K. und L. Kamphenkel (2002). *Sichere Multiparty-Berechnungen*. Diplomarbeit, TU-Braunschweig.

[Kamphenkel et al. 2009] Kamphenkel, K., S. Laumann, J. Bauer und G. Carle (2009). *Path Selection Techniques for SCTP Multihoming*. In: *First International Workshop on Medical Application Networking MAN 2009, Workshop of the International Conference on Communications, ICC 2009*.

[Kamphenkel] Kamphenkel, Kai. *Untersuchung über die Praxisrelevanz der Telemedizin anhand konkreter Arbeitsabläufe als Basis für eine Arbeit aus dem Themenkomplex Sichere Echtzeitübertragung in der Medizin-Telematik*.

[Kamphenkel et al. 2006] Kamphenkel, Kai, S. Laumann, J. Bauer und G. Carle. (2006). *SCTP zur sicheren Echtzeitübertragung in der präklinischen Sonographie*. 6. ITeG Workshop Mobiles Computing in der Medizin (MoCoMed 2006) Frankfurt.

[Kanz et al. 2002] Kanz, K. G., J. A. Sturm, W. Mutschler und u. A. N. d. DGU (2002). *Algorithmus für die präklinische Versorgung bei Polytrauma*. Der Unfallchirurg, 105(11):1007–1014.

[Kent und Atkinson 1998] Kent, S. und R. Atkinson (1998). *Security Architecture for the Internet Protocol*. RFC 2401 (Proposed Standard). Obsoleted by RFC 4301, updated by RFC 3168.

[Krawczyk et al. 1997] Krawczyk, H., M. Bellare und R. Canetti (1997). *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104 (Informational).

[Krüger und Reschke 2002] Krüger, G und D. Reschke, Hrsg. (2002). *Telematik: Netze – Dienste – Protokolle*. Fachbuchverlag Leipzig.

[Lackner und Kanz 1996] Lackner, C. und K. Kanz (1996). *Prophylaxe des posttraumatischen Organversagens durch Qualitätskontrolle*. Hefte Unfallheilkunde, 253:67–73.

[Lacroix et al. 2002] Lacroix, A., L. Lareng, D. Padeken, M. Nerlich, M. Bracale, Y. Ogushi, Y. Okada, O. Orlov, J. McGee, J. Sanders, C. Doarn, S. Preros und I. Mc Donald (2002). *International Concerted Action on Collaboration in Telemedicine: Recommendations of the G-8 Global Healthcare Applications Subproject-4*. Telemedicine Journal and e-Health, 8(2):149 – 157. LIDO-Berichtsjahr=2002,;.

[Ladha und Amer 2003a] Ladha, S. und P. Amer (2003a). *mproving Multiple File Transfers Using SCTP Multistreaming*. tech. report TR2003-06, Computer and Information Sciences Dept., Univ. of Delaware.

[Ladha und Amer 2003b] Ladha, Sourabh und P. D. Amer (2003b). *Improving Multiple File Transfers Using SCTP Multistreaming*.

[Lechleuthner et al. 1994] Lechleuthner, A, R. Lefering, B. Bouillon, E. Lentke, M. Vorweg und T. Tiling (1994). *Prehospital detection of uncontrolled haemorrhage in blunt trauma*. European Journal of Emergency Medicine, 1:13–18.

[Lehmenn et al. 2005] Lehmenn, T.M., J. Hiltner und H. Handels (2005). *Medizinische Bildverarbeitung*. Kapitel in: Lehmann T, Meyer zu Bexten E (Hrsg): Handbuch der Medizinischen Informatik. München, 1:361–423.

[Ligges 2006] Ligges, Uwe (2006). *Programmieren mit R (Statistik und ihre Anwendungen)*. Springer.

[Lindner und Studer] Lindner, Guido und R. Studer. *Forecasting the Fault Rate Behavior for Cars*.

[Ma et al. 1995] Ma, O. J., J. R. Mateer, M. Ogata, M. P. Kefer, D. Wittmann und C. Aprahamian (1995). *Prospective analysis of a rapid trauma ultrasound examination performed by emergency physicians.*. J Trauma, 38(6):879–885.

[McGahan et al. 2002] McGahan, J, J. Richards und M. Gillen (2002). *The focused abdominal sonography for trauma scan: pearls and pitfalls*. Journal of ultrasound in medicine, 21:789–800.

[Menezes et al. 1996] Menezes, Alfred J., P. C. van Oorschot und S. A. Vanstone (1996). *Handbook of Applied Cryptography*. CRC Press.

[Märkle 2002] Märkle, S. und Lemke, UH. (2002). *Die elektronische Patientenakte – Ist eine Standardisierung in Sicht?*. Tagungsband der Jahrestagung des VDE.

[Muss] Muss, Mike. *ttcp – Ein Tool für die Bestimmung des Durchsatz einer TCP/IP Verbindung*. ftp://ftp.arl.mil/pub/ttcp.

[Nagle 1984] Nagle, J. (1984). *Congestion control in IP/TCP internetworks*. RFC 896.

[Nast-Kolb et al. 1993] Nast-Kolb, D., C. Waydhas, S. Kastl, K. Duswald und L. Schweiberer (1993). *Stellenwert der Abdominalverletzung für den Verlauf des Polytraumatisierten*. Der Chirurg, 64:552–559.

[NIST 2003] NIST (2003). *NIST Net - A Linux-based Network Emulation Tool General paper on NIST Net*. Computer Communication Review.

[Nordhoff 2006] Nordhoff, M. (2006). *Security Evaluation of SCTP*. Diplomarbeit, University of Duisburg-Essen – Computer Networking Group –.

[Orebaugh, A. and Syngress Autorenteam 2004] Orebaugh, A. and Syngress Autorenteam (2004). *Ethereal Protokollanalyse*. mitp.

[Paland and Riepe 2005] Paland, N. and C. Riepe (2005). *Politische Aspekte und Ziele der Gesundheitstelematik*. Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz, 48(6):623–628.

[Peters and Runggaldier 2005] Peters, O. and K. Runggaldier (2005). *Algorihmen im Rettungsdienst*. Urban & Fisher.

[Pfllügmeyer 2001] Pfllügmeyer, M. (2001). *Informations- und Kommunikationstechnologien zur Qualitätsverbesserung im Kramkenhaus – Computerbasierte Terminologien als semantische Basis für medizinische Informationssysteme*. PhD thesis, Sozial- und Wirtschaftswissenschaftlichen Fakultät der Johannes Kepler Universität Linz.

[Rescorla and Modadugu 2006] Rescorla, E. and N. Modadugu (2006). *Datagram Transport Layer Security*. RFC 4347 (Proposed Standard).

[Richards et al. 2002] Richards, John R, N. H. Schleper, B. D. Woo, P. A. Bohnen, and J. P. McGahan (2002). *Sonographic assessment of blunt abdominal trauma: a 4-year prospective study.*. J Clin Ultrasound, 30(2):59–67.

[Roll 2008] Roll, Oliver (2008). *Sicheres Echtzeitstreaming von Videosequenzen*. Master's thesis, Universität Tübingen – Lehrstuhl für Rechnernetze und Internet.

[Schäfer 2003] Schäfer, Günter (2003). *Netzsicherheit*. dpunkt.

[Schneier 1996] Schneier, B. (1996). *Angewandte Kryptographie:*. Addison Wesley.

[Schug and Redders 2005] Schug, S. H. and M. Redders (2005). *Gesundheitstelematik-Projekte in Deutschland aus Ländersicht*. Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz, 48(6):649–656.

[Sloan 2001] Sloan, Joseph D. (2001). *Network Troubleshooting Tools*. O'Reilly.

[Stevens 1998] Stevens, W. Richard (1998). *Unix network programming (2 vols)*. Prentice Hall, 2 ed.

[Stevens 2007] Stevens, W. Richard (2007). *TCP/IP*. Hüthig Verlag.

[Stewart 2007] Stewart, R. (2007). *Stream Control Transmission Protocol*. RFC 4960 (Proposed Standard).

[Stewart et al. 2004] Stewart, R., M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad (2004). *Stream Control Transmission Protocol (SCTP) Partial Reliability Extension*. RFC 3758 (Proposed Standard).

[Stewart et al. 2000] Stewart, R., Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson (2000). *Stream Control Transmission Protocol*. (2960). Obsoleted by RFC 4960, updated by RFC 3309.

[Stewart et al. 2007] Stewart, R., Q. Xie, M. Tuexen, S. Maruyama, and M. Kozuka (2007). *Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration*. RFC 5061 (Proposed Standard).

[Stewart and Metz 2001] Stewart, Randall and C. Metz (2001). *Sctp: New transport protocol for tcp/ip*. IEEE Internet Computing, 05(6):64–69.

[Stewart and Xie 2001] Stewart, Randall R. and Q. Xie (2001). *Stream Control Transmission Prorocol (SCTP)*. Addison Wesley.

[Tayal et al. 2004] Tayal, Vivek S, M. A. Beatty, J. A. Marx, C. A. Tomaszewski, and M. H. Thomason (2004). *Fast (focused assessment with sonography in trauma) accurate for cardiac and intraperitoneal injury in penetrating anterior chest trauma.*. J Ultrasound Med, 23(4):467–472.

[Tuexen and Seggelmann 2008] Tuexen, M. and R. Seggelmann (2008). *Datagram transport layer security for stream control transmission protocol*. Network Working Group Internet-Draft.

[Tüxen] Tüxen, Michael. *Sctplib - an sctp implementation from university of essen/siemens ag*. http://www.sctp.de/sctp.html.

[Unurkhaan 2005] Unurkhaan, Esbold (2005). *Secure End-to-End Transport over SCTP*. PhD thesis, Universität Duisburg-Essen.

[Unurkhaan et al. 2004] Unurkhaan, Esbold, E. P. Rathgeb, and A. Jungmaier (2004). *Secure sctp - a versatile secure transport protocol*. Telecommunication Systems, 27(2-4):273–296.

[Vanit-Anunchai 2008] Vanit-Anunchai, Somsak (2008). *Towards formal modelling and analysis of sctp connection management*. In *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*.

[Walcher 2003] Walcher, F. (2003). *Präklinische Sonographie*.

[Walcher et al. 2002] Walcher, F, S. Kortüm, T. Kirschning und I. Weihgold, N.and Marzi (2002). *Optimierung des Traumamanagements durch präklinische Sonographie*.

[Walcher et al.] Walcher, F., M. Weinlich, G. Conrad, U. Schweigkofler, R. Breitkreutz, T. Kirschning und I. Marzi. *Prehospital ultrasound imaging improves management of abdominal trauma*. British Journal of Surgery, 93(2):238–242.

[Wang und Yu 2005] Wang, Xiaoyun und H. Yu (2005). *How to Break MD5 and Other Hash Functions*. In: *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, S. 19–35.

[Waydhas et al. 1997] Waydhas, C., K. G. Kanz, S. Ruchholtz und D. Nast-Kolb (1997). *Algorithms in the early management of severely injured patients*. Der Unfallchirurg, 100(11):913–921.

[WHO 97] WHO (97). *World Health Organization – A health telematics policy*. In: *Geneva: Report of the WHO Group Consultation on Health Telematics*.

[Witten 2001] Witten, Ian H. Eibe, Frank (2001). *Data Mining*. Hanser.

[Wootton et al. 2006] Wootton, Richard, J. Craig und V. Patterson (2006). *Introduction to Telemedicine*. Royal Society of Medicine Press Ltd.

[Wätjen 2003] Wätjen, Dietmar (2003). *Kryptographie: Grundlagen, Algorithmen, Protokolle*. Spektrum Akademischer Verlag.

[Ärzte Zeitung 2008] Zeitung Ärzte (2008). *E-Card: Meinungen prallen aufeinander – E-Card: Meinungen prallen aufeinander*. Ärzte Zeitung.

[Ziegenfuss 1998] Ziegenfuss, T. (1998). *Polytrauma Präklinische Erstversorgung und Schocktraumamanagement*. Der Anaesthesist, 47(5):415–431.