Optimierung von Übertragungsprotokollen zur Verbesserung der Dienstgüte in Telemedizinischen Echtzeitanwendungen

Von der Fakultät für Ingenieurwissenschaften
Abteilung Elektrotechnik und Informationstechnik
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Pascal Aljoscha Klein
aus
Duisburg

Gutachter: Prof. Dr.-Ing. Axel Hunger
Gutachter: Prof. Dr.-Ing. Torben Weis
Tag der mündlichen Prüfung: 25.10.2017

Vorwort und Danksagung

Als ich im Jahr 2011 als wissenschaftlicher Mitarbeiter in der Technischen Informatik der Universität Duisburg-Essen begann, wusste ich noch nicht, worauf ich mich einließ. Jetzt, wo es vollbracht ist, will ich mit Freuden den zahlreichen Leuten, ohne die das alles gar nicht möglich gewesen wäre, dafür danken, dass sie mich in dieser Zeit unterstützt und stets an mich geglaubt haben.

Zunächst danke ich Herrn Prof. Dr.-Ing. Axel Hunger, der mich durch diese Zeit gebracht hat, mich stets mit den nötigen Ressourcen versorgte, mir die nötige Freiheit gab mein Thema zu finden und zu bearbeiten. Er hat mir so viel beigebracht, dass der Platz hier nicht ausreichend ist es aufzuzählen. Mein Dank gebührt außerdem Herrn Prof. Dr.-Ing. Torben Weis für die Übernahme des Korreferats.

Eine Realisierung dieser Arbeit wäre ohne meine lieben Kollegen nicht möglich gewesen. Hier ein gewaltiges Dankeschön an meinen Kollegen und Freund, Dr.-Ing. Stefan Werner, der mir stets mit Rat und Tat zur Seite stand, und auch in dunklen Zeiten mit warmen Worten bei mir war. Dipl.-Ing. Joachim Zumbrägel, der mich immer mit der nötigen Hardware versorgen konnte, die ich gerade brauchte, und den ich stets mit Fragen löchern konnte. Dem Sekretärinnen-Team, das immer für mich da war: Sigrid Wolters, Elvira Laufenburg sowie Marion Bröckels. Dipl.-Ing. Uwe Dippel, der immer dafür offen war, meine Fragen zu beantworten und diese akribisch zu lösen versuchte. Dr. Renate Kärchner-Ober, die immer interessiert war und mir mit ihren Erfahrungen zur Seite stand, Dipl.-Inf. Ina Wentzlaff, die mit mir als Leidensgenossin ein Büro teilte, und mit der ein verbaler Austausch mir oftmals wie Balsam war. Hinzu kommen die anderen wissenschaftlichen Mitarbeiter, denen ich ebenfalls für ihre geistige Unterstützung danken möchte: Alexander Maxeiner, Torben Gebhardt, Vahid Sohani, Yue Wu, Yuwei Liang, Angela Hirlehei und Astha Franziskus Ekadiyanto.

Zu Danken habe ich den vielen Studierenden, die mit an diesem Projekt gearbeitet haben und ohne die all dies nicht möglich gewesen wäre: Afiq Mohamad Riandrayana, Alexander Siahaan, Christian Natanael, Chunyao Gao, David Schellenburg, Francis Nweke, Hemalatha Krishnamurty, Huan Wang, Imranullah Khan, Jiadai Guo, Jie Zhang, Juan Cai, Kevin Jaya Darfian, Maneli Khanshaghaghi, Maoyuan Feng, Marco Tesch, Martin Verbunt, Masoud Shoja, Maurits Nicodemus, Michael Püttmann, Nicolas Dammin, Özbil Yilmaz, Paniz Gorji, Prajnavira Taslim, Sadiq Rehman, Salman Adjie Wiratama, Timur Damin und Zhengkai Chen.

Zu großem Dank verpflichtet bin ich ebenfalls meinem Freund und Mentor Heinz Oerter und seinen hilfsbereiten Kollegen, meinem früheren Matheprofessor Prof. Dr. rer. nat. Johannes Gottschling, meinen Eltern und Geschwistern sowie allen meinen Freunden – vor allem aber Shenja Jeworek, Boris Mrsic, Robin Verhoolen und Nina Tenhaef, die mich die ganze Zeit ertragen haben.

Zuletzt geht der größte Dank an meine Allerallerliebste, ohne die ich keine Seite weit gekommen wäre: Vera – Ich liebe Dich!

Inhaltsverzeichnis

Abkürzungsverzeichnis

ACK	Acknowledge (TCP)
AOI	Area of Interest
API	Applicatio Programmable Interface
ARPA	Advanced Research Projects Agency
ARQ	Automatic Repeat reQuest
ARTP	Augmented Reality Telepointer
AS	Autonomes System
BER	Bit-Fehler-Rate (Bit Error Rate)
BGP	Border Gateway Protocol
CSCW	Computer Supported Cooperative Work
EEG	Elektroenzephalogramm
EKG	Elektrokardiogramm
FEC	Forward Error Correction
FIFO	First In First Out
FIN	Finish (TCP)
FQ	Fair Queueing
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IKT	Informations- und Kommunikationstechik (siehe auch IT)
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System Protocol
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Informations- und Telekommunikationstechnik
ITU	International Telecommunication Union
LAN	Local Area Network
LLS	Logical Link Control
MAC	Medium Access Control
MAN	Metropolitan Area Network
MPLS	Multi Protocol Label Switching
MPRTP	Multipath Real-Time Protocol
MPTCP	Multipath Transport Control Protocol
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
NSP	Netzwerk Service Provider
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OWD	One Way Delay (dt. Einweglaufzeit)
PD	Pfad-Diversität
PDU	Protocol Data Unit
PQ	Prioritiy Queueing
QoS	Quality of Service
RED	Random Early Detection
RFC	Request For Comments
RIP	Routing Information Protocol
rMPTCP	redundant Multipath Transport Control Protocol
RR	Round-Robin

RSVP	Resource reSerVation Protocol
RTP	Real Time Protocol
RTT	Round Trip Time (dt. Umlaufzeit)
SCTP	Stream Control Transmission Protocol
SDN	Software Defined Network
SLA	Service Level Agreement
SPoF	Single Point of Failure
SYN	Synchronize (TCP)
TCP	Transport Control Protocol
TE	Traffic Engineering
TTL	Time To Live
UDE	Universität Duisburg-Essen
UDP	User Datagram
UKM	Universiti Kebangsaan Malaysia
WAN	Wide Area Network
WFQ	Weighted Fair Queueing
WHO	World Health Organization
WYSIWIS	What You See Is What I See

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die Nutzung von Echtzeitanwendungen über das Internet hat in den letzten Jahren stetig zugenommen. Online-Spiele, Kleingerätesteuerungen und interaktive Applikationen des „Internet of Things“, telemedizinische Anwendungen sowie Video- und Audioanwendungen sorgen für ein zunehmend höheres Datenaufkommen. Dabei wurde das Internet in seinen Anfängen weniger für eine echtzeitige Kommunikation entworfen, sondern eher, um Massendaten zu übertragen, die weniger zeitkritischen Anforderungen unterliegen. Die Zunahme von Echtzeitanwendungen schlägt sich in gewandelten Anforderungen an die Internetarchitektur nieder:

- bestimmte Anbieter verlangen, die Netzneutralität aufzuheben, um vorrangigen Datentransfer für ihre spezialisierten Anwendungen zu erhalten;
- Netzversorger stellen ihre Infrastruktur auf besser kontrollierbarere Techniken um.

Die Anforderungen, die Applikationen an einen einwandfreien Betrieb stellen, bekommen einen zunehmenden Stellenwert bei der Entwicklung und Verbesserung der Netzwerkarchitektur. Für interaktive Echtzeitanwendungen sind vor allem geringe Verzögerungen bei einem mäßigen Bedarf an Datenübertragung wichtig. In diese Anwendungsklasse fallen verschiedene Kommunikationsarten und Anwendungen des computergestützten kooperativen Arbeitens. Ein Beispiel für Anwendungen in diesem Bereich sind Übertragungen für die Telemedizin, die sich in einer rasanten Weiterentwicklung befindet.

Telemedizinische Anwendungen bieten vielversprechende Aussichten, um dem zunehmenden Ärztemangel in ländlichen Gebieten entgegenzuwirken, aber auch, um in ärmeren Ländern eine bessere medizinische Versorgung zu ermöglichen. Das Internet bildet das Rückgrat für die Verbreitung dieser Anwendungen, da es zu einer vielseitigen Erreichbarkeit angeschlossener Geräte beiträgt.

Bei der Datenübertragung unterliegen telemedizinische Anwendungen besonderen Anforderungen. Sie werden nicht nur zur Übertragung von schützenswerten persönlichen Daten verwendet, sondern auch, um lebenskritische Notfälle und Operationen in Echtzeit zu überwachen, zu begleiten und Hilfestellungen zu ermöglichen. Dies bedingt besondere Anforderungen an die Dienstgüte bei der Übertragung von relevanten Daten.

Die heterogene Struktur des Internets bringt Einschränkungen für diese Art von Anwendungen mit sich. Umstellungen in verbesserte Architekturen dauern lange oder sind meist nur in kleinem Umfang möglich. Viele Bestrebungen im Bereich der Forschung und Entwicklung zielen auf ein zuverlässigeres Internet, das eine bessere Basis für solche Anwendungen bieten soll.

1.1 Motivation

Zwischen der Universität Duisburg-Essen (UDE) und der Universiti Kebangsaan Malaysia (UKM) existiert eine langjährige Partnerschaft [Hunger et al., 2013]. Neben mehreren Projekten im akademischen Bildungsbereich werden zunehmend auch gemeinsame Forschungsprojekte durchgeführt. Eine wichtige Zusammenarbeit bei der gemeinsamen Forschung besteht im Bereich der Telemedizin. Hier wird an einem Werkzeug, einem ferngesteuerten „Augmented-Reality-Telepointer“, gearbeitet, welches in diversen telemedizinischen Szenarien eingesetzt werden soll [Karim et al., 2013]. Diese Szenarien umfassen Anwendungen in der Tele-Diagnostik, der Tele-Konsultation und der Tele-Chirurgie, aber auch Anwendungen für ein Distance-Learning-Szenario, der Tele-Ausbildung.

Von den Forschungen im Bereich der Telemedizin profitieren beide Länder. In Deutschland wird nach effizienzverbessernden Maßnahmen geforscht, um die hohen Gesundheitskosten zu verringern und dem strukturellen Wandel in der medizinischen Versorgung entgegenzuwirken. Eine verbesserte medizinische Versorgung und eine erweiterte Mobilität von Ärzten sind auch Forschungsschwerpunkte in Malaysia. Im Rahmen der Kooperation soll ein Austausch von Expertisen zwischen den beiden Ländern stattfinden. Gemeinsame Forschungsansätze werden integriert, um ein kooperatives Forschungsnetzwerk zu etablieren. Eine Ausdehnung der Forschungspartnerschaft wird derzeit mit Myanmar im Bereich des öffentlichen Gesundheitswesens erwogen.

Zur Erforschung verschiedener Aspekte im Bereich telemedizinischer Anwendungen wurde ein Basisszenario entworfen, bei dem ein Experte einen medizinischen Notfall aus der Ferne begleitet und Maßnahmen dirigiert. Der Patientenstandort ist hierfür mit mehreren informationstechnischen Gerätschaften ausgestattet, die den räumlich entfernten Experten mit den nötigen Daten versorgen. Dazu gehören Datenströme zur Überwachung der Lebensfunktionen, Audio- und Videoübermittlung sowie die Steuerung des Telepointers, welche besondere Echtzeitanforderungen an eine Verbindung stellt.

Für die Zusammenarbeit zwischen UDE und UKM erfolgt die Kommunikation zwischen den beiden Standorten über das Internet. Diese Verbindung ist mit vielfältigen Problemen konfrontiert, bedingt durch die große Distanz von ca. 10.000 km (Luftlinie) sowie eine unterentwickelte Infrastruktur gegeben durch den Schwellenland-Status Malaysias. An der UKM bestehen mehrere nutzbare Verbindungen, die alle unter ähnlichen Einschränkungen leiden: Die lange Kommunikationsstrecke zwischen Deutschland und Malaysia führt zu hohen Datenverlusten und ist starken zeitlichen Schwankungen bei den Übertragungszeiten ausgesetzt.

Die Dienstgüte von Übertragungen hängt im Internet von den vorhandenen Kapazitäten ab, die mit allen anderen Benutzern zu teilen sind. Vor allem in Malaysia bildet das Netzwerk einen Flaschenhals, gegeben durch zu viele Anwendungen und Benutzer gegenüber den zur Verfügung gestellten Kapazitäten. Hieraus entstehen Latenzzeiten, die für das oben vorgeschlagene Szenario mit zeitsensibler Echtzeitübertragung unge-

nügend sind. Die praktizierten Anwendungen in der Telemedizin stellen besonders anspruchsvolle Anforderungen an die Dienstgüte.

Die im Internet existierenden Ansätze für Lösungen der benannten Probleme sind vielfältig. Dies hängt mit der heterogenen Struktur des Internets zusammen, die einen Zusammenschluss verschiedenster Netzwerke und Protokolle innerhalb des Architekturmodells zulässt. Vielversprechende Ansätze existieren auf Seiten der Netzwerkbetreiber, welche allerdings nur mit einem Umbau der Infrastruktur umzusetzen sind.

Im Bereich der Transportprotokolle liegt die Kontrolle des Sende- und Empfangsverhaltens beim Endbenutzer. Das Transport Control Protocol (TCP) ist wegen seines zuverlässigen Sendeschemas das meist verwendete Internet-Transportprotokoll, auch wenn es bereits seit den Anfängen des Internets existiert. Durch das zeitverzögerte erneute Senden von Segmenten können Verluste aufgefangen werden. Dabei entstehen allerdings hohe Latenzzeiten und zusätzlicher Datenverkehr. Das User Datagram Protocol (UDP) ist eine Alternative für Echtzeitanwendungen ohne besondere Ansprüche an die Zuverlässigkeit. Anwendungen, die beide Vorteile nutzen wollen, bedürfen zusätzlicher Erweiterungen der genannten Protokolle, die wiederum andere Nachteile mit sich bringen.

Das ursprünglich für mobile Anwendungen entwickelte Multipath-TCP [Barré et al., 2011] Transportprotokoll erlaubt es, mehrere Netzwerkverbindungen gleichzeitig zu nutzen, und so die Datenübertragungsrate zu steigern. Es ist kompatibel zum TCP-Protokoll und kann damit im Internet eingesetzt werden. Eine experimentelle Implementierung wurde innerhalb des Standards RFC 6824 [Ford et al., 2013] veröffentlicht. Eine mögliche Methode, um Datenverluste und Latenzschwankungen auf der Verbindungsstrecke UDE-UKM auszugleichen, ist die Nutzung zusätzlicher Redundanzen auf mehreren Verbindungen. Unter Verwendung von Multipath-TCP lässt sich eine Erweiterung erschaffen, die mithilfe zusätzlicher Redundanz und durch die Nutzung mehrerer Pfade diese Probleme ausgleichen kann.

1.2 Problemstellung und Zielsetzung der Arbeit

Die gemeinsamen Anstrengungen von UDE und UKM haben zum Ziel, das oben beschriebene telemedizinische Szenario zu realisieren. Dazu gehört primär die Verbesserung der Dienstgüte, also der Übertragungsqualität der versendeten Daten.

Im Rahmen dieser Arbeit soll ein alternatives Transportprotokoll entwickelt werden, das in der Lage ist, sich besser an die Gegebenheiten der Datenverbindung anzupassen sowie den hohen Anforderungen der telemedizinischen Anwendungen zu entsprechen. Erreicht werden soll dies über eine Verbesserung der Dienstgüte. Unter der Verwendung von Multipath-TCP soll das Protokoll mehrere Verbindungen nutzen können und mithilfe zusätzlicher Redundanz aufgetretene Latenzspitzen und Datenverluste ausgleichen.

Hierfür ist es zunächst erforderlich, eine Spezifizierung der Dienstgüteanforderungen der anfallenden Datenströme des oben angesprochenen Szenarios vorzunehmen. Es

sollen die Arbeitsgebiete der Telemedizin abgesteckt und die Dienstgüteanforderungen von telemedizinischen Dienstarten ermittelt werden.

Eine grundlegende Erörterung von Dienstgüte-Methodiken im Internet in Kombination mit dem derzeitigen Stand der Technik soll für den Prozess der Entwicklung des Protokolls nachvollziehbar dargestellt werden. Zu diesem Zweck werden die elementaren Funktionsweisen und Mechanismen im Internet, die die Dienstgüte bereitstellen, dargelegt.

Da es vor allem um die Verbesserung der Dienstgüte auf der Verbindungsstrecke zwischen UDE und UKM geht, bietet es sich an, die Übertragungsstrecke zu evaluieren und die aufgetretenen Probleme zu identifizieren. Eine Evaluierung der verschiedenen verfügbaren Verbindungen zwischen UDE und UKM soll der Ausleuchtung einer kombinierten Nutzungsweise dienen.

Das Ziel dieser Arbeit ist es schließlich, eine Infrastruktur aufzubauen, die die erhebliche Distanz zwischen UDE und UKM mit verbesserter Dienstgüte überbrückt und verschiedene Kleingeräte miteinander verbindet.

Eine Analyse des zu entwickelnden Zustands ergibt die folgende Infrastruktur zwischen UDE und UKM:

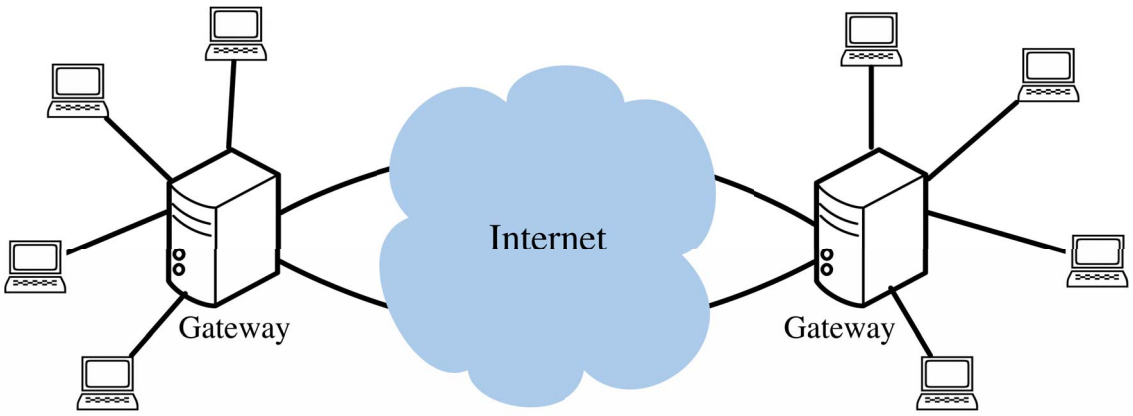

Abbildung 1.1: Infrastruktur zwischen UDE und UKM

Einzelne Entitäten sollen mittels Relais-Rechnern verbunden werden und mit der geplanten Erweiterung des Multipath Transport Control Protocol (MPTCP) eine verbesserte Verbindung der Strecke erlauben.

Überlegungen zu sicherheitsrelevanten Maßnahmen gegenüber möglichen Angreifern der hier entwickelten Protokollarchitektur werden innerhalb dieser Arbeit ausgeklammert und die Standardprozeduren von TCP und Multipath-TCP verwendet.

1.3 Phasen der Entwicklung und Aufbau der Arbeit

In Kapitel zwei geht es um die telemedizinischen Aspekte dieser Arbeit sowie um die Anforderungen von Echtzeitanwendungen in der Telemedizin. Hierfür wird eine Klassifizierung der verschiedenen Dienstarten vorgenommen und auf angrenzende Themenbereiche eingegangen. Dazu gehören eine kurze Abhandlung der Normen und

Standards sowie eine kurze Einführung in die Nutzung von Telerobotik und Telepointern in der Telemedizin. Zudem wird die Dienstgüte von in der Telemedizin genutzten Anwendungen thematisiert. Die Erkenntnisse hieraus werden an das oben beschriebene Szenario angeknüpft.

In Kapitel drei geht es um die Dienstgütegrundlagen im Internet. Als das in dieser Arbeit ausgewählte zugrundeliegende Architekturmodell wird zunächst das ISO/OSI-Referenzmodell betrachtet und die elementaren für die in dieser Arbeit wichtigen Schichten dargestellt. Es folgt eine kurze Abhandlung über die Grundsätze des Routings. Darin werden grundlegende Funktionsweisen darüber vorgestellt, wie Datenpakete im Internet vermittelt werden. Danach wird auf die Dienstgütemechanismen im Internet eingegangen, die eine Regulierung und Behandlung des Datenverkehrs beschreiben. Es wird der aktuelle Stand der Technik erläutert und eine erste Problemformulierung hinsichtlich dieser im Internet tragenden Mechanismen formuliert.

Kapitel vier beschäftigt sich mit der Kommunikationsverbindung zwischen UDE und UKM. In diesem Kontext werden Elemente der Partnerschaft zwischen UDE und UKM und die involvierten identifizierten Netzwerke beschrieben. Aussagen werden zu Routenbestimmungen und Messungen der verschiedenen UKM-Verbindungen getroffen. Unterschiedliche Dienstgüteparameter werden erfasst und evaluiert. Es wird eine mögliche Kombination verschiedener Verbindungen diskutiert und eine weiterführende erste Problemformulierung bezüglich der Verbindung und der Telemedizin geboten.

In Kapitel fünf werden Möglichkeiten einer Dienstgüteverbesserung erörtert. Darin wird speziell auf die ermittelten Probleme der Datenübertragung eingegangen und aktuelle Techniken werden hierzu diskutiert. Es wird die in dieser Arbeit entwickelte Lösung dargestellt und einer alternativen Parallelentwicklung gegenübergestellt.

Kapitel sechs beschäftigt sich mit der Modellierung und Entwicklung der modifizierten Protokollvariante. Es werden die grundsätzlichen mathematischen Zusammenhänge diskutiert und eine Einführung in die Funktionalitäten des Protokolls gegeben.

In Kapitel sieben wird die Implementierung des Protokolls und die Entwicklung von verschiedenen zusätzlichen Bestandteilen wie dem Gateway-Rechner beschrieben. Es werden Hilfserweiterungen für das Protokoll und Entwicklungen zur Auswertung und Evaluierung thematisiert.

Kapitel acht beschäftigt sich mit der Evaluierung des Protokolls und seinen Bestandteilen. Verschiedene Aspekte wie die Justierung der Parameter werden diskutiert.

Die Arbeit wird mit Kapitel neun, dem Fazit, abgeschlossen. Die Ergebnisse der Arbeit werden diskutiert und weiterführende Arbeiten besprochen. Es wird ein Ausblick auf zukünftig zu entwickelnde Funktionen gegeben.

2 Echtzeitanwendungen in der Telemedizin

Anwendungen in der Telemedizin gehören zu den sicherheitskritischsten Applikationen im professionellen Bereich. Die Bandbreite der Telemedizin reicht von einem einfachen Telefonat mit einem Arzt über die digitale Verwaltung von Gesundheitsdaten bis hin zu komplexen Fernoperationen mithilfe eines Roboters. Das Internet bietet nur begrenzte Eigenschaften, die Anwendungen der Telemedizin zu unterstützen.

In diesem Kapitel soll ein Überblick über den für diese Arbeit relevanten Bereich der Telemedizin gegeben und eine Problemdefinition hinsichtlich der Telemedizin entwickelt werden. Das Kapitel beginnt mit einer Einführung in die Telemedizin, bei der grundsätzliche Definitionen und Anwendungen erläutert werden. Telemedizinische Roboteranwendungen sowie eine Darstellung der in dieser Arbeit verwendeten Telepointertechnologie werden nähergehend betrachtet. Das Kapitel endet mit einer Problemformulierung hinsichtlich der Dienstgüte von telemedizinischen Anwendungen.

2.1 Begrifflichkeiten und Definitionen

Der Ausdruck *Telematik (franz.: telematique)* wurde im Jahr 1978 entwickelt und setzt sich aus den Begriffen **Tele**kommunikation und Infor**matik** zusammen [Nora & Minc, 1978]. Die Telematik beschreibt alle getrennten oder gemeinsamen Anwendungen von Telekommunikationstechnik und Informatik [Europäische Kommission, 1994]. Telematikanwendungen sind „einrichtungsübergreifende und ortsunabhängige vernetzte Anwendungen zur Überbrückung von Raum und Zeit, um damit betriebliche oder überbetriebliche Geschäftsprozesse jeglicher Art zwischen Unternehmen oder diesen und ihren Kunden abzuwickeln und/oder ganz oder teilweise zu automatisieren“ [Haas, 2006]. Abbildung 2.1 zeigt eine Überschneidung der genannten Begriffe.

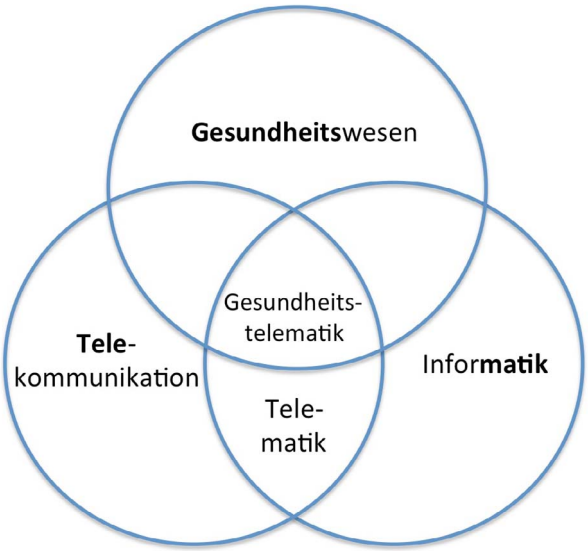

Abbildung 2.1: Einordung des Begriffs Gesundheitstelematik[Haas, 2006]

Die Telematik im Gesundheitswesen bzw. die *Gesundheitstelematik (health telematics)* ist auf Technologien für den medizinischen Sektor fokussiert. Laut der World Health Organization (WHO) wird die Gesundheitstelematik wie folgt definiert [World Health Organization, 1998]:

„Die Gesundheitstelematik ist ein Sammelbegriff für gesundheitsbezogene Aktivitäten, Dienste und Systeme, die über eine Entfernung hinweg mit Mitteln der Informations- und Kommunikationstechnologie ausgeführt werden: zum Zweck globaler Gesundheitsförderung, Krankheitskontrolle und Krankenversorgung sowie für Ausbildung, Management und Forschung für das Gesundheitswesen."

Gesundheitstelematik umfasst vielfältige medizinische Anwendungen, die mithilfe von Informations- und Kommunikationstechniken (IKT) ausgeführt werden. Es lassen sich folgende Funktionsbereiche ausmachen [Gärtner, 2006] [Horsch & Handels, 2002]:

- Patientenversorgung
- Ausbildung und Lehre
- medizinische Forschung
- Gesundheitsmanagement

Hiervon lässt sich der Begriff der *Telemedizin* abgrenzen. Die Telemedizin wird durch die WHO definiert als [World Health Organization, 1998]:

„[..] die Einbringung von Gesundheitsdienstleistungen durch Gesundheitsberufstätige unter Verwendung von Informations- und Kommunikationstechnologie zum Austausch gültiger Informationen für Diagnose, Therapie und Prävention von Krankheiten und Verletzungen, für Forschung und Bewertung, sowie für die kontinuierliche Ausbildung von Gesundheitsdienstleistern im Interesse der Förderung der Gesundheit von Individuen und ihren Gemeinwesen, wenn dabei die räumliche Entfernung einen kritischen Faktor darstellt."

Während die Definition für Gesundheitstelematik durch die WHO nur von einer undefinierten Entfernung spricht, wird für die Telemedizin die Distanz als kritischer Faktor angesehen. [Gärtner, 2006] betrachtet die Telemedizin als Unterkategorie der Gesundheitstelematik und stellvertretend für die von der WHO veröffentlichte Definition für den Funktionsbereich der Patientenversorgung. Für [Haas, 2006] steht der Begriff der Telemedizin für eine veraltete Begrifflichkeit, mit der nur noch ganz bestimmte „[...] eng medizinisch orientierte Zweitmeinungs- und Konsultations-Anwendungen wie Verfahren der Telepathologie, Teleradiologie, Telechirurgie usw. [...]" gemeint sind. [Palmer et al., 2009] legen in einem Report der Europäischen Kommission aus dem Jahr 2008 die Vorteile der Telemedizin für die Europäische Union dar und geben die folgende Definition:

„Telemedicine comprises ICT (Information and Communication Technology) - enabled healthcare services that are provided to patients in situations where one or more health care professionals and the patient are not in the same location. It involves

secure transmission of medical data and information, through text, sound, images or other forms needed for prevention, diagnosis, treatment and follow-up of patients."

Eine Studie aus dem Jahr 2007 hat von 104 verschiedenen wissenschaftlichen Beiträgen die Definitionen von Telemedizin näher betrachtet [Sood et al., 2007]. Sie gibt folgende wesentliche Aspekte an:

- breit gefächerte Komplexität (z.B. E-Mail-basiert, fernsteuerbare Roboter)
- Zustellung von Gesundheitspflege primär durch Medien oder Informationskanäle
- Überbrückung von Entfernungen mithilfe von Technologie
- im Wesen sich stetig verändernd in Anbetracht der sich wandelnden Informations- und Kommunikationstechnologie
- Versprechen von Verbesserungen (z.B. Kostenersparnis, Versorgungsqualität etc.)
- zentriert auf einen Patienten

Wenn es allgemein um Praktiken im Gesundheitssektor geht, bei denen IKT eingesetzt wird, hat sich mittlerweile international der Begriff *E-Health* etabliert. Im Jahr 2015 führte die WHO unter dem Namen „Third Global Survey on E-Health" eine globale Umfrage bei nationalen Gesundheitsministerien durch [World Health Organization, 2015]. Demnach ist der Begriff der Gesundheitstelematik mittlerweile komplett verschwunden. Ein Jahrzehnt zuvor hatte sich die Begriffsbestimmung für E-Health als eine Art gesundheitsbezogene Datenverarbeitung ausgebildet. Ein Artikel aus dem Journal of Medical Internet Research aus dem Jahr 2005 [Oh et al., 2005] beschreibt eine Sammlung der verschiedenen Definitionen von E-Health, die die folgenden Aspekte beinhalten (ebd.):

- technologische Verwendung im Gesundheitssektor
- Zusammenarbeiten zwischen Personen (z.B. Anbieter, Versorger, Patienten)
- Aktivitäten im Zusammenhang mit Gesundheit
- positive Entwicklungen innerhalb des Gesundheitssektors
- räumliche Unabhängigkeit
- Kosteneffizienz

Eine weitere Definition aus dem Jahr 2004 unterscheidet außerdem E-Health und Telemedizin durch die Art der Datenverwendung [Jäckel, 2004]: Während E-Health keinerlei personenbezogene Daten verwendet und somit datenschutzunkritisch ist, beschäftigt sich die Telemedizin direkt mit dem Patienten und seinen persönlichen Daten. Es kann also zwischen personenbezogenen Daten, medizinischem Wissen und Verwaltungsdaten im Gesundheitswesen unterschieden werden [Horsch & Handels, 2002].

In einer Studie aus dem Jahr 2015 grenzt die WHO den Begriff E-Health von weiteren Kategorien ab, die dabei helfen, Dienste in abgelegenen Besiedlungen und in unterversorgten Gemeinschaften bereitzustellen [World Health Organization, 2016]:

- *mHealth*, womit eine Nutzung von Gesundheitsdiensten durch ein mobiles Endgerät beschrieben wird
- *Telehealth*, als die Zustellung von Gesundheitsdienstleistungen, bei denen Patienten und Anbieter durch eine Distanz getrennt sind

Der Begriff *Telehealth* hat also bei der WHO den Begriff der Telemedizin, welcher in der Studie gar nicht mehr verwendet wird, weitestgehend ersetzt. In einem Papier der deutschen Gesellschaft für Telemedizin und der DGTelemed, das einen Überblick über den derzeitigen Stand der Telemedizin in Deutschland gibt, wird der Begriff der Telemedizin nach wie vor als Oberbegriff benutzt und sogar synonym zur Definition von E-Health der WHO verwendet [Brauns & Loos, 2015].

Auch auf der Internetseite des Bundesgesundheitsministeriums findet der Begriff Telemedizin weiterhin Verwendung: Eine dort artikulierte Definition beschreibt Telemedizin als ein Verfahren, um „[...] unter Einsatz audiovisueller Kommunikationstechnologien trotz räumlicher Trennung zum Beispiel Diagnostik, Konsultation und medizinische Notfalldienste anzubieten [...]“ [Bundesgesundheitsministerium (Hg.), 2016]. Die Formulierung lehnt sich eng an die Definition von [Bird, 1971] an, einem der ersten Telemedizin-Pioniere: „the practice of medicine without the usual physician-patient confrontation [...] via [an] interactive audio-video communications system.“ Obwohl hier lediglich die audiovisuelle Komponente in Betracht gezogen wird, weist die Definition bereits eindeutig in Richtung Echtzeitübertragung. Telemedizin kann als eine Summe von Kommunikationsmodalitäten angesehen werden, die die Übertragung von medizinischen Daten, Videobildern und Audiodaten zwischen Ärzten und anderen Leistungserbringern erlauben [Croteau & Vieru, 2002].

Die angeführten Beispiele zeigen, dass die vorgestellten Begriffe seit Ende des zwanzigsten Jahrhunderts ständiger Veränderung unterliegen. Dies liegt zu einem großen Teil daran, dass sich die zugrundeliegenden Technologien und Möglichkeiten mit dem technischen Fortschritt teilweise drastisch verändert haben. Zur Beschreibung des hierdurch manifestierten technologischen Wandels werden neue Begriffe geschaffen und alte auf bestimmte Aktivitäten beschränkt.

In dieser Arbeit wird der Begriff **E-Health** als Oberbegriff für die allgemeine Benutzung von IKT im Gesundheitssektor verwendet. In Abgrenzung hiervon und in Anlehnung an die weiterhin gebräuchliche deutsche Verwendung sowie unter Betonung des in dieser Arbeit verwendeten Fokus auf Echtzeitübertragung, soll nachfolgend der Begriff **Telemedizin** im Sinne einer örtlich getrennten und in Echtzeit stattfindenden Praktizierung von Medizin verwendet werden.

Ein telemedizinisches System umfasst die Kombination einer medizinischen Aktivität, die über die Ferne durchgeführt wird, und technischen Geräten, die hierzu benötigt werden. Nach der Definition der [Brockhaus Enzyklopädie Online, 2017] ist ein System ein aus mehreren Teilen zusammengesetztes, gegliedertes Ganzes.

Betrachtend nach dem Shannon'schen Informationsmodell sind die Teile des medizinischen Systems (ebd.):

1. der Anbieter einer Information, z.B. ein Arzt
2. der Dienstleistungsempfänger, z.B. ein Arzt, ein Patient
3. die Daten, die die Information ausmachen, z.B. medizinische Daten bzw. Steuerinformationen für medizinisches Equipment

Zusammenfassend lässt sich definieren, dass ein **telemedizinisches System** ein möglichst in sich geschlossenes technisches System ist, welches durch den Einsatz von IKT den Anbieter, den Abnehmer und die Übertragung von Daten, die dazu dienen, medizinische Hilfe zu erbringen, umfasst. Es setzt sich also aus den partizipierenden Akteuren, den notwendigen Daten und der dafür benötigten Informationstechnologie zusammen. Ein telemedizinisches System beinhaltet hierzu Technologien aus:

- der Informatik
- der Telekommunikation
- der Krankenhaustechnik
- der Maschinentechnik

2.2 Einführung in die Telemedizin

Die klassische Telemedizin gibt es bereits, seit es Möglichkeiten zur Übermittlung von Informationen gibt. Mit der Entwicklung der elektronischen Datenübermittlung begann auch eine Nutzung in medizinischer Hinsicht. Bereits die Erfindung des Telegrafen wurde für die Übermittlung von medizinischen Informationen verwendet [Zundel, 1996].

Ein sehr frühes Beispiel ereignete sich im Jahr 1876, als sich der Erfinder des Telefons, Alexander Graham Bell, mit Säure verletzte und das Telefon nutzte, um seinen Kollegen im Nebenraum darüber zu informieren [Deter & Markovski, 2011]. Eines der ersten telemedizinischen Systeme wurde 1967 zwischen dem Massachusetts General Hospital und der Krankenstation des Bostoner Flughafens eingerichtet [Bashshur et al., 2000]. Einen wissenschaftlichen Aspekt bekam die Telemedizin, als in den 1970er Jahren Konzepte für die Raumfahrt benötigt wurden, die es erlaubten, einen Spezialisten für medizinische Notfälle hinzuzuschalten und die Lebensfunktionen der Astronauten zu überwachen [Davis, 1974]. Hierbei wurden Herzfrequenz, Blutdruck, Atmungsfrequenz und Temperatur der Astronauten konstant auf den Flügen beobachtet [Bashshur et al., 2000]. Im Jahr 1985 wurde Telemedizin durch die National Aeronautics and Space Administration (NASA) zum ersten Mal zur Soforthilfe nach einem Erdbeben eingesetzt [Eren et al., 2007] [NASA (Hg.), 1985].

2.2.1 Ziele, Vorteile und Herausforderungen von Telemedizin

Mithilfe der Telemedizin kann eine bestimmte Distanz zwischen Patienten und Ärzten überbrückt werden. Dies ermöglicht den flexiblen Zugriff eines Arztes bzw. die Verfügbarkeit einer ärztlichen Expertise von theoretisch überall aus [Brauns & Loos, 2015].

Die Distanz kann allerdings nicht beliebig ausgedehnt werden, da bestimmte technische Grenzen bei der Übertragung von Informationen gelten.[1] Ein telemedizinisches System lässt eine weitgehende Flexibilität hinsichtlich der Umgebung und des Standorts zu, sodass die Partizipatoren nicht an bestimmte Umgebungen gebunden sind. Einsätze werden an entfernten Standorten möglich, die sonst nicht oder nicht rechtzeitig erreicht werden könnten. Dazu zählen:

- ländliche Umgebungen, in denen es nur unzureichende Versorgung gibt [Bundesgesundheitsministerium (Hg.), 2016]
- Krisengebiete [Eren et al., 2007] [Llewellyn, 1995]
- isolierte Umgebungen [Matusitz & Breen, 2007]

Telemedizin ermöglicht es, Operationen direkt vor Ort durchzuführen, auch wenn dort die entsprechende Expertise fehlt. Aufwändige oder riskante Patiententransporte lassen sich damit vermeiden [Mohr et al., 2004]. Hierdurch werden nicht nur Zeit [Croteau & Vieru, 2002] und Kosten [Sood et al., 2007] durch wegfallende Anreisen eingespart, in manchen Fällen wird eine medizinische Versorgung an bestimmten Orten überhaupt erst ermöglicht. Notoperationen, die mithilfe von zugeschalteten Spezialisten erfolgen, sind oftmals die letzte Chance für den Patienten [Brévart et al., 2011].

Bestimmte gesellschaftliche und medizinische Entwicklungen lassen den Einsatz von Telemedizin sinnvoll erscheinen. Dazu zählen die steigende Prävalenz chronischer Erkrankungen [Reiter et al., 2011], die zunehmende Alterung der Bevölkerung und eine daraus resultierende Kostensteigerung im Gesundheitswesen [Häcker et al., 2008]. Darum ist damit zu rechnen, dass die Bedeutung der Telemedizin in Zukunft noch größer werden wird.

Einen Beitrag könnte die Telemedizin auch angesichts des zunehmenden Ärztemangels auf dem Land leisten [Grönemeyer, 2015]. In vielen ländlichen Gebieten in Deutschland herrscht ein Mangel an Ärzten, während städtische Gegenden eher überversorgt sind [Albrecht et al., 2015]. Mithilfe der Telemedizin sind Ärzte aus den Städten in der Lage, Online-Sprechstunden bereitzustellen, die von der Bevölkerung auf dem Land genutzt werden können.

Eine andere Möglichkeit ist das Heranziehen von Expertenwissen, das auf dem Land nicht direkt verfügbar ist. Hiermit können Versorgungslücken in ländlichen Regionen geschlossen und eine stabile Patientenversorgung gewährleistet werden [Brauns & Loos, 2015]. Dies gilt auch für die bessere medizinische Versorgung in Entwicklungsländern, in denen ein Fachkräftemangel außerhalb großer Städte herrscht [Wootton, 2008].

Zusammenfassend kann die Telemedizin einen Nutzen für die beteiligten Akteure wie Patienten, Krankenhäuser, Ärzte, Kostenträger und Gesellschaft erzeugen (nach [Graf v.d. Schulenburg et al., 1995]):

- wirtschaftliche Effekte und Kostensenkungen

[1] Vgl. Kapitel 2.4

- Steigerung der Verteilungs- und Versorgungsgerechtigkeit [Brauns & Loos, 2015]
- Steigerung der Effektivität und Effizienz wissenschaftlicher Forschung und Ausbildung
- Verbesserung der gesundheitspolitischen Steuerung
- Verbesserung der Diagnose und Behandlung von Krankheiten sowie der Versorgungsqualität
- Verbesserung der organisatorischen Effektivität
- Verbesserung der Lebensqualität und Lebenserwartung [Häcker et al., 2008]

Eine stetige technische Verbesserung der telemedizinischen Anwendungen führt nicht zwingend zu einer Verbesserung der oben dargestellten Vorteile, da es viele Hemmnisse für den Gebrauch der technischen Anlagen gibt. Darunter fallen zunächst Akzeptanzprobleme bei vorwiegend älteren Menschen, die den Verlust des direkten persönlichen Arztkontaktes befürchten und darum neuartige Methoden nur bedingt getestet werden können [Dittmar et al., 2009]. Ebenfalls ist es schwierig, neue Methoden in bereits etablierte Vorgehensweisen bei anderen Nutzergruppen zu integrieren. Viele Verfahren sind zunächst mit einem erhöhten Arbeitsaufwand durch doppelte Buchführung sowie größerem Dokumentations- und Evaluierungsaufwand verbunden, was die Akzeptanz bei den Ärzten und Kostenträgern verringert [Klar & Pelikan, 2011]. Hinzu kommen die unterschiedlichen Systeme, Betriebsabläufe und Dokumentationsformen, die von den beteiligten Stellen eingesetzt werden (ebd.).

Rechtliche Unzulänglichkeiten hemmen die Einbindung von telemedizinischen Methoden in den medizinischen Alltag und deren weitere Entwicklung. Diese Probleme betreffen gesetzliche Änderungen im Hinblick auf Datenschutz oder den Unwillen der Behörden bei der Umsetzung gesetzlicher Aufträge [Brauns & Loos, 2015]. Die gesetzliche Nachweiserbringung des Nutzens der telemedizinischen Anwendung ist oftmals mit erheblichem Aufwand verbunden, der nicht einfach umsetzbar ist. Im Kontext dieser Arbeit ist vor allem ein extensiv ausgelegtes Fernbehandlungsverbot problematisch (ebd.). Demnach ist es nur erlaubt eine Fernbehandlung über kurze Distanzen, innerhalb eines Gebäudes, zu ermöglichen.

Die technischen Anforderungen für telemedizinische Anwendungen und Prozesse sind teilweise erheblich. Maßgebliche Barrieren bestehen im Bereich der Echtzeitanwendungen, die besondere Ansprüche an die genutzte Hardware und die Übertragung stellen [Malindi, 2011]. Die Diskussion um diese Anforderungen stellt einen Hauptfokus dieser Arbeit dar und wird in den nächsten Kapiteln tiefgehender betrachtet.

2.2.2 Anwendungsgebiete und Klassifizierung der telemedizinischen Dienstarten

Ein Einsatz von Telemedizin kann in mehreren medizinischen Gebieten erfolgen. [Dittmar et al., 2009] nehmen eine einfache Einteilung telemedizinischer Anwendungen in drei Kategorien vor:

1. Telediagnostik
2. Homecare
3. Spezialanwendungen

Hinzu kommen Verfahren, die eine Mischung aus medizinischen Anwendungsbereichen, nutzbar für die Telemedizin, sowie telemedizinischen Grundszenarien darstellen (ebd.).

Entsprechend der in Kapitel 2.1 angegebenen Definition und entscheidend für den Kontext dieser Arbeit lassen sich die folgenden **telemedizinischen Dienstarten** unterscheiden:

- Telekonsultation
- Telediagnostik
- Teletherapie
- Telemonitoring (Überwachung, Beobachtung)
- Notfalltelemedizin
- Teleausbildung
- Tele-Operation (z.B. Chirurgie)
- Teledokumentation

Ein auf die Funktionsweise bezogenes entscheidendes Kriterium der unterschiedlichen Dienstarten ist die Unterscheidung, ob die Übertragung bzw. Interaktion zwischen den Partnern zur selben Zeit (*synchron*) oder zu unterschiedlichen Zeiten (*asynchron*) stattfindet.

Bei der *Telekonsultation* wird eine Telekommunikationsverbindung verwendet, damit zwei oder mehrere Ärzte über das diagnostisch-therapeutische Vorgehen bei der Behandlung eines konkreten Krankheitsfalles beraten können [Feussner et al., 1998]. Ein Arzt zieht hierbei einen Experten hinzu, um Fragen in Spezialfällen zu klären, oder um eine zweite Meinung heranzuholen. Die Telekonsultation kann synchron wie auch asynchron ablaufen und es können unterschiedliche IKT-Werkzeuge von Chat über Videokonferenz bis hin zu einfacher Email- oder Foren-Konsultation genutzt werden [Delrobaee et al., 2006]. Die Telekonsultation ist einer der telemedizinischen Dienste, die in dieser Arbeit im Speziellen behandelt werden.

Die *Telediagnostik* erlaubt die Diagnose eines spezifischen Problems über die Ferne [Dugas & Schmidt, 2003]. Sie findet Anwendung, um Krankheitssymptome durch einen anderen, entfernten Arzt untersuchen zu lassen. Die Teleradiologie und auch die Telepathologie sind telemedizinische Anwendungen dieses Dienstes. Röntgenbilder oder Fotos bestimmter Symptome werden an einen Experten zur Befundung oder an den zu behan-

delnden Arzt übermittelt [Bundesgesundheitsministerium (Hg.), 2016]. Zu einer Diagnose zählen gleichfalls Untersuchungen und Diagnostizierung eines Patienten aus der Ferne.

Die *Teletherapie* ermöglicht die Durchführung therapeutischer Maßnahmen mithilfe von IKT zwischen Arzt und Patient [Bundesgesundheitsministerium (Hg.), 2016]. Hierbei ist die Standortunabhängigkeit von besonderer Bedeutung: Für den Patienten bietet dies den Vorteil, von einem beliebigen Standort aus eine Therapie erhalten zu können; der behandelnde Therapeut muss nicht mehr unmittelbar mit dem Patienten interagieren, sondern kann die Therapie z.B. von seiner Praxis aus durchführen. Denkbar sind sowohl synchrone als auch asynchrone Methoden.

Beim *Telemonitoring* geht es um die Überwachung eines Patienten ohne räumliche Verbindung [Dugas & Schmidt, 2003]. Hiermit können Vitalwerte wie Puls, Blutdruck und Körpertemperatur vom behandelnden Arzt aus der Ferne überwacht werden. Gegebenenfalls können Krankenhausaufenthalte vermieden und die Aufzeichnung von Daten unter alltäglichen Lebenssituationen durchgeführt werden. Synchrone Anwendungen sind z.B. eine Direktübertragung der Werte über ein Mobilfunknetz oder das automatische Absetzen eines Notrufs bei Verschlechterung bestimmter Werte. Asynchrone Anwendungen zeichnen die Vitalwerte auf und geben diese erst zu einem bestimmten Zeitpunkt an den behandelnden Arzt weiter, wenn z.B. eine Netzverbindung verfügbar ist oder wenn das Aufzeichnungsgerät in bestimmten Intervallen ausgewertet wird [Häcker et al., 2008]. Das Senden von Messdaten wird auch *Telemetrie* genannt (ebd.).

Die *Notfalltelemedizin* dient der zeitnahen Übermittlung von Daten während eines Notfalls [Anästh Intensivmed (Hg.), 2016]. Der wichtigste Aspekt bei dieser Anwendungsart ist die mobile Natur des Szenarios. Daten werden unmittelbar vor der Einlieferung an den behandelnden Arzt übertragen, damit der Patient sofort und ohne weitere Tests nach Eintreffen behandelt werden kann (ebd.). Eine Datenübertragung in umgekehrter Reihenfolge ist ebenfalls denkbar, sodass Patientendaten direkt zum Arzt übertragen werden, während er auf dem Weg zum Unfallort ist.

Die *Teleausbildung* ermöglicht es, Teilnehmer im Rahmen von Bildungsangeboten aus der Ferne zu unterrichten [Horsch & Handels, 2002]. Dabei werden synchrone Bildungsangebote, wie z.B. eine Videoübertragung des Lehrenden, oder asynchrone Bildungsangebote, wie z.B. eine Online-Dokumentation, über ein bestimmtes Verfahren geschaltet. Hierbei liegt das Haupteinsatzgebiet bei der medizinischen Lehre (ebd.).

Bei der *Tele-Operation* (hier abzugrenzen von *Teleoperation*[2]) geht es darum, operative Eingriffe aus der Ferne durchzuführen [Hanly & Broderick, 2005]. Die Telechirurgie ist hierfür ein mögliches Anwendungsgebiet. Bei der Tele-Operation kann ein behandelnder Arzt oder Experte einen chirurgischen Roboter steuern oder mithilfe der Telekonsultation beratend während einer Operation dazugeschaltet sein. Es hat sich gezeigt, dass eine die Tele-Operation begleitende Videokonferenz eine Operation per Roboter stark verbessern kann [Hanly & Broderick, 2005]. Die Tele-Operation ist nach

[2] Vgl. Kapitel 2.2.4

zeitlichen Kriterien in erster Linie synchroner Natur. Für die Tele-Operation gelten die strengsten Regeln zur Dienstgüte.[3] Die Dienstart der Tele-Operation ist ein Schwerpunkt in dieser Arbeit.

Die *Teledokumentation* beschreibt Anwendungen, „bei denen mehrere Akteure unabhängig von Raum und Zeit gemeinsam eine logisch zentrale Dokumentation nutzen und je nach Berechtigung einsehen, fortschreiben und löschen können“ [Haas, 2006]. Einsatz findet die Teledokumentation bei der kooperativen Dokumentation von Krankheitsfällen und beim Protokollieren von Symptomen beim Patienten. Die einfachste Möglichkeit der Teledokumentation ist die Bereitstellung eines digitalen Verzeichnisses oder einer Datei, die durch mehrere Benutzer beschrieben oder gelesen werden können (ebd.).

Alle hier genannten Dienstarten können unter Verwendung einer mobilen Datenverbindung unter die Definition von mHealth fallen. Insbesondere sind dies Telemonitoring und Notfalltelemedizin, da diese im Gegensatz zu den anderen Dienstarten jederzeit und von unterwegs aus durchgeführt werden können bzw. müssen. Die anderen Dienstarten eignen sich eher dazu, in vorhersehbaren Räumen eingesetzt zu werden, wenn es sich um Übertragungen in Echtzeit handelt.

Eine weitere Unterscheidung hinsichtlich einer synchronen oder asynchronen Übertragung lässt sich bezüglich der Nutzung telemedizinischer Daten treffen. Hierbei wird zwischen Anwendungen differenziert, die wie folgt deklariert werden können [Malindi, 2011]:

- *store-and-forward* (Speichern und Weiterleiten)
- *near real-time* (nahe Echtzeit)
- *real time* (Echtzeit)

Das *Speichern und Weiterleiten* wird verwendet, wenn es sich um keine akute Notsituation handelt (ebd.): Daten werden erhoben und mit einer möglichen Zwischenspeicherung von 24 bis 48 Stunden verzögert weitergeleitet. Im Falle von *Echtzeit*-Übertragungen werden die Daten sofort „live“ verwendet (ebd.). Einen Sonderfall bildet die *nahe Echtzeit*, bei der in Notsituationen erhobene Daten so schnell wie möglich weitergesendet werden, um der nachfolgenden Operation oder Konsultation im Vorfeld zu helfen (ebd.). Eine weitere Unterscheidung trifft [Hamann, 2002] zwischen

- Online-Modus und
- Offline-Modus.

Während eine *Online*-Interaktion dann gegeben ist, wenn die kooperierenden Personen zur selben Zeit aktiv in Beziehung zueinander handeln, ist die *Offline*-Interaktion dann gegeben, wenn die miteinander kooperierenden Personen nicht zur selben Zeit auf die Daten zugreifen (ebd.).

[3] Vgl. Kapitel 2.4.6.6

Unter Verwendung der von [Johansen, 1991] vorgestellten Zeit-Raum-Matrix kann eine grafische Einteilung der verschiedenen Dienstarten unter Erweiterung der oben vorgeschlagenen Unterscheidungen erfolgen. Die ursprüngliche Matrix unterscheidet zwischen zeitgleicher und nicht gleichzeitiger Interaktion, die im selben Raum oder in unterschiedlichen Räumen stattfinden kann.

Eine erweiterte Variante bietet [Grudin, 1994], der die Zeit-Raum-Matrix um die Gesichtspunkte „vorhersehbar“ und „nicht vorhersehbar“ erweitert. Die in dieser Arbeit betreffenden Anwendungen beschreiben Interaktionen, die nur in unterschiedlichen Räumen stattfinden. Hierfür soll eine Variante eingeführt werden, die keinen Bezug hinsichtlich des Raumes herstellt, aber die Gesichtspunkte „vorhersehbar“ sowie „nicht vorhersehbar“ unter dem Aspekt des Standortes beinhaltet.

Die Unterscheidung zwischen vorhersehbaren und nicht-vorhersehbaren Standorten bei der Telemedizin ist sinnvoll, da hiervon signifikant die Auslegung der benötigten Technik abhängt. Ein vorhersehbarer Standort kann z.B. über weitaus komplexere Technologien verfügen, als dies bei einem Standort von unterwegs aus möglich ist. Für nicht-vorhersehbare Standorte wird dementsprechende mobile Technologie benötigt, die die notwendigen Techniken von unterwegs aus anbieten kann. Entsprechende Beispiele hierfür sind das Telemonitoring und die Notfalltelemedizin, bei welchen davon auszugehen ist, dass der Ort des Patienten nicht bekannt ist. Aus diesem Grunde muss Technologie genutzt werden, die „mitbringbar“ bzw. portabel ist.

Abbildung 2.2 zeigt eine Darstellung der Zeit-Raum-Matrix nach [Johansen, 1991] und [Grudin, 1994] mit eigener Erweiterung um „Nahe Echtzeit“ und unter Einordung der telemedizinischen Dienstarten.

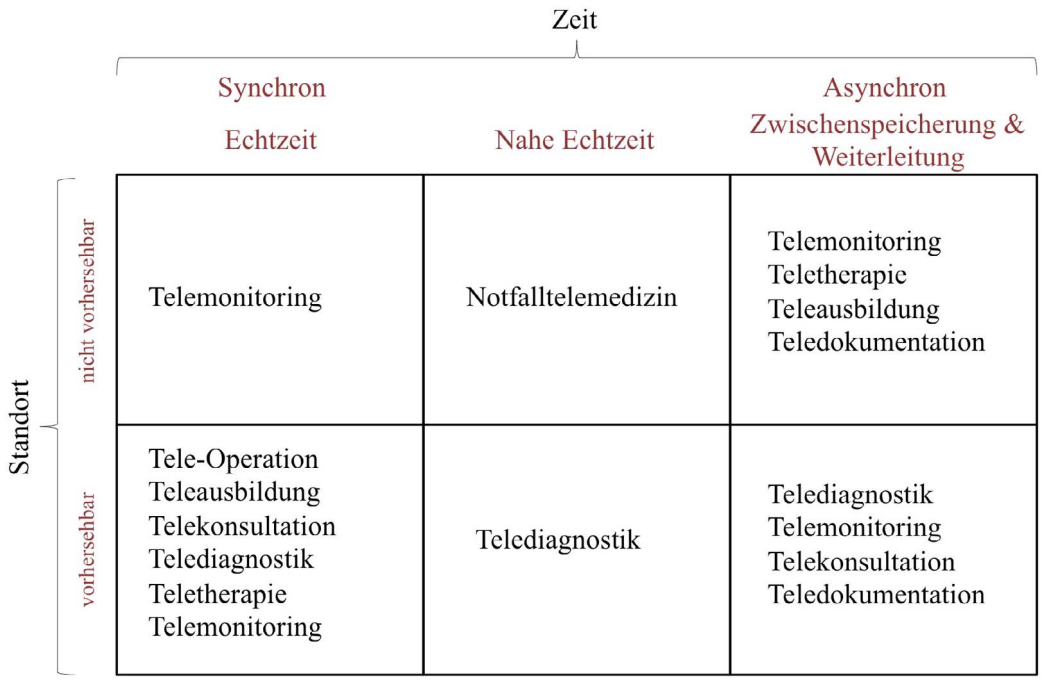

Abbildung 2.2: Zeit-Raum-Matrix nach [Johansen, 1991] und [Grudin, 1994] mit Erweiterungen

Andere, speziell asynchrone Technologien der Telemedizin können größtenteils unabhängig von einem vorhersehbaren Standort betrachtet werden bzw. erscheinen sie in der Regel weitaus unproblematischer. Hierfür werden Technologien genutzt, die bei Teilnehmern entweder schon „zu Hause“ vorhanden sind – ohne einen speziell medizinischen Fokus zu besitzen – oder die weniger komplex sind und über keine aufwendigen Übertragungsmechanismen verfügen. Hierzu zählen z.B. vorhandene Hardware wie Telefon, Computer mit Kamera und Mikrofon, Smartphones oder auch einfache Sensoren, die nur einmal innerhalb einer gewissen Zeitspanne ausgelesen werden müssen.

Des Weiteren kann zwischen verschiedenen Anwendungsbereichen unterschieden werden [Hensel et al., 2002] [Duftschmid et al., o.D.]

- Doc-to-Doc und
- Doc-to-Patient

Hierbei werden telemedizinische Dienstarten, die entweder für die Expertenberatung oder die Patientenbetreuung verwendet werden, unterschieden. Eine mögliche Einteilung der in dieser Arbeit getroffenen Auswahl verschiedener Dienstarten in der Telemedizin zeigt die nachfolgende Abbildung 2.3, die durch [Hensel et al., 2002] entwickelt wurde. Sie wurde für den Kontext dieser Arbeit mit den verwendeten telemedizinischen Dienstarten angepasst.

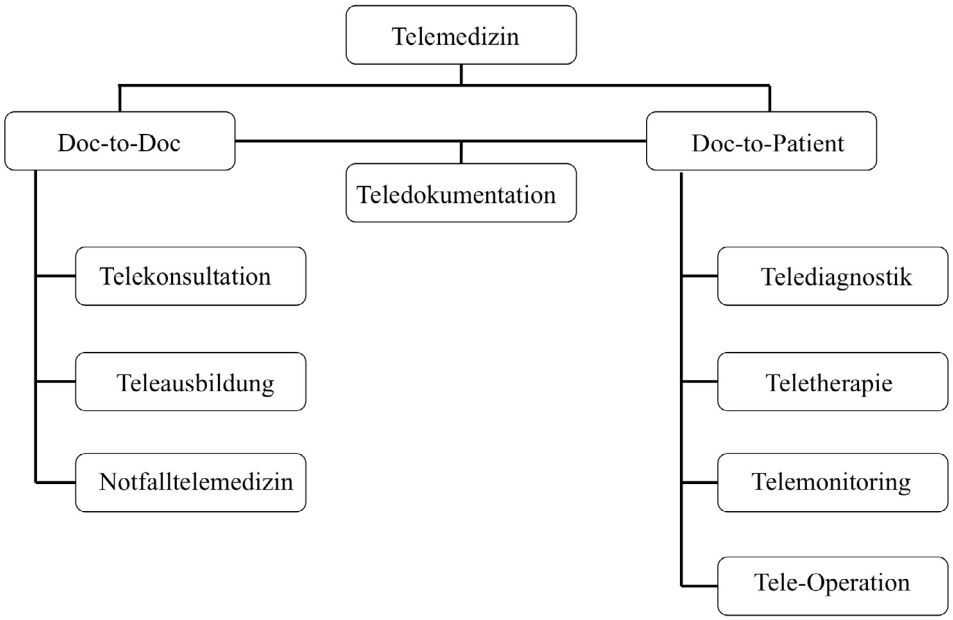

Abbildung 2.3: Gebiete der Telemedizin in Patient/Doktor-Beziehung nach [Hensel et al., 2002]

2.2.3 Normen und Standards in medizinischen Netzwerken

Netzwerke in telemedizinischen Systemen werden durch Anwendungen genutzt, deren Daten kritischen Maßstäben unterliegen. Medizinische Geräte werden in Netzwerke integriert und wandeln sich damit zu medizinischen Netzwerken [Gärtner, 2011]. Die DIN EN 80001-1 [Deutsches Institut für Normung (Hg.), 2010] bezieht sich auf die Nutzung von medizinischen Geräten und das Management ihrer Risiken in IT-

Netzwerken, um Gefährdungen und Risiken zu reduzieren und weitestgehend zu beherrschen [Gärtner, 2012]. Dazu gehören vor allem eingesetzte Prozessketten im medizinischen Umfeld und die Definition von Verantwortlichkeiten [Deutsches Institut für Normung (Hg.), 2010].

Netzwerke besitzen unterschiedliche Risiken in Relation zu den darin verwendeten Anwendungen. Eine grobe Einstufung von Computernetzwerken in verschiedene Kritikalitätsklassen, die einem medizinischen Einsatz unterworfen werden, bieten [Schrempf & Zauner, 2009]:

- Netzwerkklasse A: allgemeine Computernetzwerke, die für die Erfassung, Bearbeitung und Weitergabe medizinischer und administrativer Daten benutzt werden [Wasem et al., 2013]
- Netzwerkklasse B: Computernetzwerke im direkten klinischen Einsatz für spezialisierte Anwendungen wie bildgebende Medizinprodukte, die einen Zugang zum Zentralarchiv besitzen
- Netzwerkklasse C: Computernetzwerke in hochsicherheitskritischen klinischen Bereichen, die eine isolierte Verbindung benötigen, ohne in Kontakt mit anderweitigen Geräten zu stehen

Die Netzwerkklassen A und B können unter Beachtung bestimmter Sicherheitsvorkehrungen miteinander verbunden werden. Netzwerkklasse C steht für eigenständige Funktionen, die sicherheitskritisch sind und deshalb isoliert in Kliniken installiert sein müssen [Schrempf & Zauner, 2009]. Hierfür gilt die DIN EN 60601-1 [Deutsches Institut für Normung (Hg.), 2006], die den Umgang und die Sicherheit mit medizinischen elektrischen Geräten festlegt.

2.2.4 Telerobotik in der Medizintechnik

Bei der Telerobotik bzw. Teleoperation geht es darum, eine Interaktion mit einem vom Benutzer entfernten Umfeld zu ermöglichen [Pala et al., 2012]. Hierfür werden Maschinen bzw. Roboter eingesetzt, die aus der Ferne gesteuert werden können.

Eine *Teleoperation* – hier zu unterscheiden von Tele-Operation im medizinischen Sinne[4] – ist eine Aktivität, bei der ein technisches Gerät mithilfe einer geeigneten Mensch-Computer-Schnittstelle ferngesteuert wird [Cui et al., 2003]. Das System selbst, das dazu verwendet wird, entfernte Interaktionen zu ermöglichen, wird *Teleoperator* genannt. Ein Teleoperator ist eine Maschine, die die Fähigkeiten eines menschlichen Anwenders auf eine Distanz erweitert und es erlaubt, eine Apparatur aus der Ferne zu bedienen oder mechanisch zu bewegen [Batsomboon & Tosunoglu, 1996].

Einer der ersten Teleoperatoren wurde im Jahr 1898 durch Nikola Tesla entworfen, um ein Schiff aus der Ferne elektronisch zu steuern [Tesla, 1898]. Eine andere Bezeichnung für ein System, bei dem eine Maschine durch einen Benutzer ferngesteuert wird, ist *Telemanipulator* [Fischer & Voges, 2011], wobei der Begriff „Manipulation“ bereits die Veränderung der Eigenschaften bzw. Beschaffenheit eines Objekts impliziert.

[4] Vgl. Kapitel 2.2.2

Ein System, bei dem ein Eingabegerät eine Arbeitseinheit kontrolliert und so vom Benutzer direkt vorgegebene Bewegungen an eine empfangende Einheit übertragen werden, wird auch *Master-Slave-Manipulatorsystem* genannt (ebd.): Diese Apparaturen sind im klassischen Sinne keine Roboter, da sie Bewegungen weitergeben und nicht einer festen Programmierung folgen. Die einfachste Variante ist eine mechanische Kopplung zwischen Master und Slave (ebd.).

Höher entwickelte Varianten sind fernsteuerbare Manipulatoren mit diversen Werkzeugen, die über ein Netzwerk gesteuert werden können (ebd.). Durch den Verein Deutscher Ingenieure wird ein Roboter als universell einsetzbarer Bewegungsautomat definiert, dessen Bewegungen frei programmierbar und gegebenenfalls sensorgeführt sind, und der mit Greifern, Werkzeugen oder anderen Fertigungsmitteln aufrüstbar ist, um Handhabungs- und Fertigungsaufgaben ausführen zu können [VDI, 1990].

Die Teleoperation eines Roboters und damit die virtuelle Anwesenheit unterscheidet sich nach wie vor von einer echten Anwesenheit – einer tatsächlichen live-Interaktion. Ein Forschungsfeld, das sich mit der Immersion eines Benutzers an einem entfernten Ort beschäftigt, ist die sogenannte *Telepräsenz*. Bei der Telepräsenz wird eine Interaktion über große Entfernung immersiv so ermöglicht, dass die Beteiligten möglichst den Eindruck gewinnen, dass sie sich am selben Ort befänden und sich wie bei einer normalen Interaktion fühlen und verhalten [Edwards, 2011]. Um einen möglichst hohen Grad der Telepräsenz zu erreichen, werden Technologien wie 3D-Ansichten und haptisches Feedback (Telehaptik) eingesetzt, die Menschen dabei unterstützen, so natürlich wie möglich eine Teleoperation durchführen zu können.

Bei der Telemedizin werden Teleoperationen vor allem in der Telechirurgie eingesetzt, die in dieser Arbeit als eine Unterkategorie der Tele-Operation definiert wurde[5], aber auch in der Telediagnose und Teletherapie. Bei der Telechirurgie geht es um eine medizinische Operation, die aus der Ferne unterstützt oder durchgeführt wird. Hier liegt der Fokus zumeist auf der Durchführung der Operation durch einen Spezialisten, der ein medizinisches Instrument bzw. einen medizinischen Roboter fernsteuert.

Nachteil ist bei einer direkten Steuerung eines telemedizinischen Roboters, dass der gesamte Teleoperator möglichst nah beieinander liegen sollte, um die Latenz zwischen den beiden Master- und Slave-Einheiten so gering wie möglich zu halten. Für viele Operationen ist mithilfe eines Teleoperators ein Klasse C Netz vorgeschrieben – also eine physikalisch unabhängige Steuerung – da es sich um Echtzeitinformationen handelt, die nicht durch andere Datenflüsse gestört werden dürfen. Ein Betreiben über das Internet ist kritisch, da die Latenzzeiten in Abhängigkeit vom Verkehr schwanken und niemals hinreichend garantiert werden können. Es ist Aufgabe zukünftiger Internetprotokolle, des Traffic Engineerings oder einer möglichen Andersbehandlung des Internetverkehrs diese Probleme zu lösen. Für viele Anwendungsbereiche kann es aber bereits ausreichend sein, wenn die Situation im Netzwerk zu einem gewissen Teil ausgewogen und überwacht werden kann.

[5] Vgl. Kapitel 2.2.2

2.3 Telepointertechnologie

Ein *Augmented Reality Telepointer (ARTP)* wurde im Rahmen eines gemeinsamen Forschungsprojekts im Bereich der Telemedizin zwischen der Universität Duisburg-Essen (UDE) und der Universiti Kebangsaan Malaysia (UKM) entwickelt. Hierbei handelt es sich um eine Apparatur, die aus einer Kamera, einem beweglichen Laserpointer und einer Steuerungssoftware besteht, die zur Fernbedienung des Laserpointers verwendet wird.

Eine räumlich entfernte Person ist damit in der Lage, ein Bild zu empfangen und mithilfe einer Maus auf signifikante Bereiche im Realraum zu zeigen. Der ARTP dient damit der Unterstützung der Zusammenarbeit. Durch die echtzeitliche Natur einer Maschinensteuerung stellt der ARTP hohe Ansprüche an die Dienstgüte. Der ARTP ist Teil des Basisszenarios zwischen UDE und UKM. Für die Arbeit bedeutende Gesichtspunkte sollen in den nächsten Unterkapiteln erörtert werden.

2.3.1 Grundsätzliche Funktionsweise und Fähigkeiten von Telepointern

Eine erste Definition für Telepointer bieten [Murthy & Krishnamurthy, 2005] in Erweiterung der Definition eines Fernzeigegeräts (a. d. Engl.) von [MacKenzie & Jusoh, 2001]:

Ein Telepointer ist eine Interaktionsart für ein Präsentationssystem, interaktives Fernsehen und andere Systeme, bei dem sich ein Benutzer an einem entfernten Standort hinsichtlich der eigentlichen Ausgabe befindet.

Mit einem Telepointer ist es einem Benutzer möglich, auf einen entfernten *Interessensbereich (AOI – Area of Interest)* zu zeigen. Ein Telepointer übernimmt die Darstellung eines Zeigers oder Zeigewerkzeugs in einer entfernten Umgebung [Sánchez et al., 2008]. Der Benutzer kann hiermit auf Objekte zeigen und von dort aus mit den Benutzern interagieren. Im einfachsten Fall kann ein Telepointer auch ein Zeigen mit der eigenen Hand sein, die per Video übertragen wird [Karim et al., 2013].

Eine Hauptanwendung besteht bei einem Remote-Zugriff von Rechner zu Rechner auf Software-Basis: Der Mauszeiger des entfernten Rechners wird durch Bewegungen der Maus auf dem lokalen Rechner gesteuert. In dieser Form funktioniert ein Telepointer nach dem *„What-You-See-Is-What-I-See“ Prinzip (WYSIWIS)* [Stefik et al., 1987] aus dem Bereich des Computer Supported Cooperative Work (CSCW) [Ellis et al., 1991]. Hierbei sieht der Benutzer des entfernten Rechners die Änderungen auf dem sogenannten *Shared Space*, die der Benutzer am anderen Ende durchführt. Die Beteiligten sehen innerhalb des Shared Space denselben Inhalt und vorgenommene Änderungen.

Der Shared Space ist ein Raum, ob virtuell oder real, den mehrere Benutzer beobachten und manipulieren können [Bannon, 2000], und der als Grundlage der Zusammenarbeit bei Groupware und CSCW-Applikationen dient. Im oben genannten Beispiel ist dies der Desktop, auf dem der Mauszeiger bewegt wird. Darin befinden sich in der Regel Objekte, die zum Ziel der Zusammenarbeit und der gemeinsamen Manipulation werden können – sogenannte *Shared Objects*. Die Anwender können beide durch Be-

wegungen des Telepointers nachvollziehen, was auf dem Bildschirm passiert. Auf diese Weise können Telepointer die Zusammenarbeit zwischen mehreren Personen unterstützen.

Ein gebräuchlicher und essenzieller Teil der menschlichen Kommunikation ist die eingesetzte Gestik [Gutwin & Penner, 2002]. Verschiedene Formen der Gestik können mithilfe eines Telepointers in einem Shared Space genutzt werden [Dyck et al., 2004]:

- Zeigen auf ein Objekt
- Zeigen auf einen Bereich
- Zeigen einer bestimmten Richtung
- Markieren von Pfaden
- Zeichnen von Formen oder Figuren

Anders als ein Telemanipulator hat ein Telepointer keinerlei Eigenschaften, die es erlauben würden, die Beschaffenheit eines Objekts in der Realität zu verändern. Erweiterungen der Eigenschaften eines Telepointers können mithilfe verschiedener Symboliken erreicht werden, die zur Telepräsenz und der Verkörperung des Benutzers (*Embodiment*) [Nacenta et al., 2007] sowie zu Informationen des Gewahrseins (*Awareness*) [Schmidt, 2002] beitragen. Dies wird mithilfe von zusätzlichen Informationen, wie Stärke, Farben [Karim et al., 2013], Texten oder der Protokollierung von Telepointeraktivitäten, die mit dem Benutzer in Verbindung stehen, erreicht.

Telepointer können Echtzeit-Zusammenarbeit auf nützliche Weise unterstützen [Dyck et al., 2004]. Dennoch werden sie verhältnismäßig selten in modernen Groupware-Systemen eingesetzt, die im Internet zur Verfügung stehen (ebd.). Es besteht die Gefahr, dass sie durch Überlastung des Netzwerks sprunghaft und langsam werden, sodass sie nicht mehr verwendbar sind (ebd.). Andererseits erfordert die Verwendung eines Telepointers nur eine relativ geringe Datenübertragungsrate, da in der Regel nur Koordinateninformationen übertragen werden müssen [Karim et al., 2013].

2.3.2 Telepointer in der Telemedizin und Ausbildung

Telepointer können in der Medizin, aber auch in der telemedizinischen Ausbildung eingesetzt werden. Es ist eine Frage des Arbeitsablaufs für ein spezifisches Szenario, inwieweit ein Telepointer in die Aktivitäten eingebunden werden kann und darin sinnvoll erscheint.

In der Medizin eignet sich die Benutzung eines Telepointers vorrangig für die Bereiche, in denen ein Spezialist aus der Ferne in Echtzeit hinzugezogen wird. Dies kann der Fall bei Konsultationen zwischen zwei Ärzten sein, bei denen auf dem Desktop gemeinsam Bilder angeschaut werden, aber auch in der Diagnostik, der Teletherapie und anderen telemedizinischen Anwendungsfällen in Zusammenhang mit einem Patienten.

Ein wichtiger Gesichtspunkt ist die Art, wie ein Telepointer innerhalb einer Arbeitsgruppe benutzt wird. [Karim et al., 2013] unterscheiden bei der gemeinsamen Arbeit mit Telepointern zwischen zwei Modi von Kommunikationsschemata:

1. Master-Slave-Schema
2. Groupware-Schema

Beim *Master-Slave-Schema* handelt es sich um die Kommunikation zwischen zwei einzelnen Entitäten (ebd.). Die Master-Entität bedient hierbei den Telepointer und markiert der Slave-Entität eine AOI. Dies kann für die Telekonsultation, die Telediagnostik und die Teletherapie zutreffen. Mithilfe eines Telepointers kann ein Experte auf signifikante Stellen zeigen bzw. wichtige Bereiche markieren. Beim *Groupware-Schema* sind mehr als zwei Personen beteiligt (ebd.). So kann ein externer Experte einer Gruppe Hilfestellung geben, wie z.B. bei der Tele-Operation.

Der Einsatz von Telepointern in der Medizin kann sich positiv auf die anvisierte Handlung auswirken. Moderne Konsultationssysteme besitzen in der Regel eine Funktion mit Telepointer [Gackowski et al., 2011]. Ein erstes im Jahr 1999 implementiertes System, bei dem ein Software-Telepointer zum Einsatz kam, wurde zur Telekonsultation im Bereich der angeborenen Herzkrankheiten eingesetzt [Julsrud et al., 1999]. Die Mehrheit der teilnehmenden Personen (67 %) äußerte sich anschließend positiv zur Nutzung des Telepointers (ebd.). Ähnliche Systeme wurden ebenfalls in späteren Konsultationssystemen mit vergleichbaren Ergebnissen eingesetzt [Kaidu et al., 2004].

Die Nutzung von Laserpointern in der Medizin bietet eine Möglichkeit, um aus der Ferne auf Objekte im Realraum zu zeigen. Durch Fernsteuerung des Lasers wird es einem entfernten Arzt ermöglicht, während einer Behandlung auf signifikante Stellen zu zeigen. Das zu beobachtende Sichtfeld wird hierbei über eine Videoverbindung übertragen. Die Vorteile eines ARTPs in einem Operationssaal beschreiben [Karim et al., 2013]:

- es ist möglich während der Konsultation schnell auf das Objekt des Interesses zu zeigen;
- es wird keine Leitung für hohen Datendurchsatz benötigt;
- keine invasive Prozedur;
- ein ARTP kann leicht in einem Operationssaal angebracht werden;
- relativ geringe Kosten für Installation und Wartung.

Ein erstes ARTP-System wurde von [Yamazaki et al., 1999] entwickelt. Es handelte sich um einen fernbedienbaren Laserpointer in Kombination mit einer Kamera zur besseren Zusammenarbeit in der realen Welt. Der Laserpointer und die Kamera befinden sich auf einem beweglichen Miniaturfahrzeug, das durch Wechseln der Position in der Lage ist, mehrere Objektblickwinkel zu erfassen. Ein ähnliches System wurde durch [Ohta et al., 2006] im Bereich der Notfallmedizin eingesetzt und erfolgreich getestet. Vorteile ergaben sich bei der Weitergabe von Anweisungen an die ausführenden Personen. Es konnte eine erhebliche Zeitverkürzung bei der Ausführung der Handlungen beobachtet werden. Die Fehlerwahrscheinlichkeit der ausführenden Person wurde gesenkt.

Ein Einsatz in der Teletherapie ermöglicht es, dass ein behandelnder Arzt einen Patienten über eine Fernverbindung beraten kann. Durch Nutzung eines Telepointers kann

der behandelnde Arzt dem Patienten die benötigten Maßnahmen erläutern. Diese Verwendung wäre z.B. für Patienten in ländlicher Umgebung eine Alternative zur direkten Behandlung in der Praxis.

Bei der Verwendung in der Teleausbildung erhält eine Gruppe von Fernstudierenden einen vom Dozierenden freischaltbaren Zugriff auf einen Telepointer. Die Studierenden können aus der Ferne Fragen zu spezifischen Bereichen auf der Präsentationsfolie oder mithilfe eines ARTP auch hinsichtlich eines realen Objekts stellen. Ein ARTP eignet sich insbesondere für medizinische Vorträge oder Vorführungen, bei denen Erklärungs- bzw. Versuchsobjekte vorliegen. Die Teleausbildung ist ein Spezialfall, da hierbei kein Patient vorhanden sein muss und die Bedienung eines Telepointers nicht durch den Arzt oder einen Experten stattfindet, sondern durch einen Lernenden.

Für Telepointer gibt es mehrere Ursachen, die eine Übertragung stören oder ineffektiv machen können. Dies ist hängt zum einen von der Erfüllung der Dienstgüteparameter ab, die eine Anwendung erfordert. Hierzu gehören Datenübertragungsrate, Verzögerungen, Verzögerungsvarianz und Fehlerrate. Andererseits werden bei gleichzeitigem Zugriff auf den Shared Space Probleme der verteilten Systeme adressiert, auf die hier aber nicht weiter eingegangen werden soll. Dienstgüteanforderungen, die ein Telepointer an eine Übertragung stellt, sind mit Echtzeit-Maschinensteuerungen vergleichbar. Diese und deren Dienstgüteanforderungen werden im nachfolgenden Kapitel 2.4 näher erörtert.

2.4 Anforderungen an die Dienstgüte in telemedizinischen Anwendungen

Der Einsatz von Telemedizin und Telepointern im Speziellen ist von bestimmten technischen Gegebenheiten abhängig, die den Leistungsansprüchen der Anwendungen genügen müssen. Bezogen auf die Telekommunikationstechnik ist dies die Dienstgüte bzw. Quality of Service (QoS) [Tanenbaum & Wetherall, 2012]. Hierunter wird die Beurteilung von Anforderungen an die Übertragungsqualität der Anwendung und der Bedürfnisse des Benutzers verstanden (ebd.). Es gibt mehrere Mechanismen und Architekturen, um Dienstgüte innerhalb eines Netzwerks zu garantieren.[6] Die Ansprüche an die Dienstgüte können je nach Anwendung unterschiedlich sein.

Die Dienstgüte lässt sich durch bestimmte qualitative und quantitative Parameter des Dienstes ausdrücken [Oodan et al., 2003]. Diese lassen sich in drei verschiedene Attribute unterteilen, die für Anwendungen von Bedeutung sein können [Ye, 2002]:

- Pünktlichkeit (*Timeliness*)
- Genauigkeit (*Precision*)
- Korrektheit (*Accuracy*)

Bei der *Pünktlichkeit* werden zeitliche Abläufe in den Mittelpunkt der Betrachtung gestellt (ebd.). Es geht um die Zeiten der Verarbeitung, des Empfangs oder Sendens und

[6] Vgl. Kapitel 3

die Variationen darin. Mit der *Genauigkeit* wird die Menge der Informationen ausgedrückt, die durch einen Prozess erzeugt werden (ebd.). Die Geschwindigkeit und die Qualität des Informationsflusses werden durch dieses Attribut bestimmt. Bei der *Korrektheit* geht es um die Integrität der Informationen (ebd.), d.h. um die korrekte Erstellung, Übertragung und den zuverlässigen Empfang der Information und um die Frage, wie mit dem Verlust von Information umgegangen wird.

Die **Anforderungen an einen Datenfluss** können hauptsächlich durch vier Parameter ausgedrückt werden, die den oberen Attributen folgen [Tanenbaum & Wetherall, 2012]:

1. Verzögerung
2. Verzögerungsvarianz (Jitter)
3. Datenübertragungsrate
4. Verlustrate und Fehlerrate

Es gibt mehrere Besonderheiten dieser Parameter, auf die in den nächsten Unterkapiteln eingegangen wird.

2.4.1 Verzögerung

Im Allgemeinen entstehen Verzögerungen durch die Zeitspanne, die eine gesendete Nachricht von der Quelle bis zum Empfänger benötigt. Die folgenden Ursachen haben Einfluss auf diesen Parameter [Gutwin, 2001]:

- die Rechenleistung der involvierten Entitäten
- die Datenübertragungsrate der verschiedenen Netzwerksegmente
- die Distanz, die eine Nachricht zurücklegen muss
- die Anzahl der Netzwerkknoten
- die momentane Situation im Netzwerk

Die sogenannte *Ende-zu-Ende-Verzögerung* beinhaltet alle Verzögerungen, die auf dem Weg entstehen können [Kurose & Ross, 2014] und ist für eine zeitkritische Anwendung die wichtigste Messgröße. Sie beschreibt die Zeit, die ein Netzwerkpaket vom Absenden einer Anwendung bis zur Zustellung an die Anwendung auf der entfernten Seite benötigt [Bolot, 1993]. Aus netzwerktechnischer Perspektive wird sie auch Latenzzeit oder genauer *Einweglaufzeit (One-Way-Delay – OWD)* genannt.

Die diversen Verzögerungen, die innerhalb eines Netzwerks auftreten können, lassen sich wie folgt unterscheiden [Kurose & Ross, 2014]:

- *Übertragungsverzögerung* ist die Zeit, die das (Los-)Senden der gesamten Bitfolge des Datenpakets an einer Netzwerk-Entität benötigt;
- *Verarbeitungsverzögerung* ist die Zeit, die ein Datenpaket benötigt, um innerhalb einer Netzwerk-Entität verarbeitet zu werden, d.h. die Verarbeitung, die nach dem Eintreffen eines Datenpakets erfolgt;

- *Ausbreitungsverzögerung* ist die Zeit, die benötigt wird, um über eine Leitung gesendet zu werden. Die Geschwindigkeit liegt abhängig vom Medium im Rahmen der Lichtgeschwindigkeit zwischen $2*10^8$ *m/s* und $3*10^8$ *m/s;*
- *Warteschlangenverzögerung* ist die Wartezeit, die durch eine Warteschlange in einem Sende- oder Empfangspuffer entsteht.

Dabei können diese Verzögerungen *statische* sowie *dynamisch*e Anteile in Netzwerkkomponenten einnehmen [Bolot, 1993]. Sie unterscheiden sich durch die Stärke, die äußere Einflüsse auf die Netzwerkkomponenten besitzen und damit zu Veränderungen in den Verzögerungen führen.

Übertragungsverzögerungen an Netzwerkknoten sind vor allem statischer Natur. Sie sind durch die Datenübertragungsrate und die Länge der Bitfolgen fest determiniert, weshalb eine Änderung über einen längeren Zeitraum nicht zu erwarten ist. Die Verarbeitungszeit an einem Netzwerk-Knoten ist hingegen dynamischer: Je nach Arbeitsaufwand, den ein Router bewältigen muss, kann sich die Zeit der Verarbeitung erhöhen. Die Rechenleistungsfähigkeit eines Routers ist allerdings in der Regel an die weiterzuleitende Datenübertragungsrate angepasst und daher weitestgehend vernachlässigbar. Ausbreitungsverzögerungen sind abhängig vom verwendeten Medium und von der Länge der Route. Da Überlastsituationen zu Routenänderungen führen können, besitzt die Ausbreitungsverzögerung eine deutlich dynamische Auswirkung. Warteschlangenverzögerungen besitzen den größten Anteil an dynamischen Einflüssen. Bei Volllaufen einer Warteschlange ergeben sich Wartezeiten für alle Datenpakete, die neu hinzukommen, bis sie vom Router weitergeleitet werden können. Ursache hierfür sind vor allem Überlastsituationen, bei denen am Router eine größere Datenmenge hereinkommt, als auf die gewünschten Pfade weitergeleitet werden kann.[7]

Fehler bzw. Maßnahmen zur Fehlerbekämpfung können dazu führen, dass die Verzögerung zunimmt. Dies geschieht z.B. beim erneuten Senden eines Datenpakets im Falle eines Datenfehlers oder Paketverlusts. Verzögerungen entstehen auch, wenn Pakete beim Empfänger in der falschen Reihenfolge eintreffen und zunächst abgewartet werden muss, bis die vorher nötigen Pakete eingetroffen sind [Sathiaseelan & Radzik, 2004]. Solche Pakete werden auch *Out-of-Order-Pakete* genannt.

Die *Reaktionszeit* ist für einen Benutzer die Zeit, die zwischen dem Absenden einer Anfrage bis zum Empfang der ersten Rückmeldung verstreicht [Chen et al., 2004]. Sie beinhaltet neben der sogenannten *Umlaufzeit (Round Trip Time, RTT)* auch die Verarbeitung der Daten innerhalb der Applikation selbst, z.B. die Kompression und Dekompression eines Audiodatenstroms, und gegebenenfalls die Reaktion des anderen Benutzers.

Die Umlaufzeit ist die Zeit, die benötigt wird, um ein Datenpaket von einem Sender zu einem Empfänger (t_{AB}) zu senden und im Anschluss daran die Empfangsbestätigung am Sender wieder zu empfangen (t_{BA}):

[7] Vgl. hierzu Ausführungen in Kapitel 3.3.1

$$RTT = t_{AB} + t_{BA} \tag{2.1}$$

$$Reaktionszeit = RTT + Benutzerreaktion + Applikationsverarbeitung \tag{2.2}$$

Das Alter der Daten ist bei Applikationen wie z.B. Telefonie entscheidend [Kurose & Ross, 2014]. Ansprüche eines Anwenders an die Verzögerungszeit einer Applikation stehen in engem Zusammenhang mit der menschlichen Wahrnehmung. Studien im Bereich der Human Computer Interaction (HCI) zeigen, dass Menschen bereits auf kleine Unterschiede der Verzögerungszeit reagieren [Gray & Boehm-Davis, 2000].[8] Eine Verzögerung wird mit der Einheit *ms* angegeben.

2.4.2 Verzögerungsvarianz (Jitter)

Die *Verzögerungsvarianz (Jitter)* ist bedingt durch die zeitlich variable Verarbeitung der Datenpakete innerhalb des Netzwerks [Tanenbaum & Wetherall, 2012]. Änderungen im zeitlichen Ablauf entstehen durch variierende Verzögerungen der Netzwerkelemente wie Warteschlangen oder Netzwerkleitungen, hervorgerufen durch unterschiedliches Datenaufkommen und variables Routing (ebd.). Ein Jitter wird durch die folgende mathematische Gleichung 2.3 definiert. Hier wird der Jitter *J* mithilfe der Zeitstempel T durch drei aufeinanderfolgende Datenpakete i berechnet:

$$J = (T_i - T_{i+1}) - (T_{i+1} - T_{i+2}) \tag{2.3}$$

Zeitkritische Anwendungen, die maßgeblich auf einen regelmäßigen Ablauf der Übertragung setzen, sind durch diesen Parameter am meisten betroffen. Dies ist z.B. bei der Telefonie der Fall, da unterschiedlich schnell zugestellte Pakete die Kommunikation stören würden. *Jitter* wird ebenfalls mit der Einheit *ms* angegeben und kann durch Vorpuffern von Daten minimiert werden, indem die Daten zunächst gespeichert und dann mit geglätteten Abständen an die Anwendung weitergeleitet werden.

2.4.3 Datenübertragungsrate

Die *Datenübertragungsrate* gibt die benötigte Datenmenge pro Zeiteinheit für eine bestimmte Applikation bzw. ein Netzwerk an. Sie ist das „Maß für die Geschwindigkeit, mit der Daten über ein Medium übertragen werden“ [Brockhaus Enzyklopädie Online, 2017]. Maßgeblich für die Dimension dieses Parameters ist die verschickte Datenmenge der Applikation. Diese kann durch Kompressionsverfahren oder geschickte Programmierung der Anwendung minimiert werden.

Der *Datendurchsatz* entspricht der Menge an Nutzdaten (netto), die an einem bestimmten Netzwerkpunkt übertragen werden [Ernst, 2015]. Dies bezieht sich auf alle Nutzdaten, die zwischen zwei Endpunkten verschickt werden. Anwendungen, die einen hohen Datendurchsatz erfordern, wie z.B. die Übertragung eines hochauflösenden Videodatenstroms, besitzen hinsichtlich dieses Parameters hohe Ansprüche an die zugrundeliegende Netzwerkverbindung. Das Maß für die *Datenübertragungsrate* oder den *Datendurchsatz* ist *Bit/s.*

[8] vgl. Kapitel 2.4.6

Der produzierte Datenverkehr einer Anwendung lässt sich in vier Kategorien einordnen, welche aus der Dienstgüteunterstützung von ATM-Netzwerken abgeleitet werden (nach [Tanenbaum & Wetherall, 2012]):

1. *Konstante Datenübertragungsrate* – eine Applikation erzeugt einen konstanten Datenstrom. Die Ansprüche an eine Datenübertragungsrate müssen stets erfüllt werden, damit die Applikation ihre volle Funktion behält. Ein Beispiel ist die Audio- oder Videoübertragung mit konstanter Datenübertragungsrate.
2. *Variable Datenübertragungsrate* – Ein kontinuierlicher Datenstrom ist in der Lage, mithilfe von Kodierung die Qualität adaptiv zu verändern und sich den Netzwerkbedingungen anzupassen.
3. *Verfügbare Datenübertragungsrate* – Eine Applikation nutzt die volle Datenübertragungsrate eines Netzwerks aus, die verfügbar ist. Diese Kategorie trifft vor allem auf Massendaten wie z.B. File Transfer Protocol (FTP) zu.

2.4.4 Verlust- und Fehlerrate

Die *Verlustrate* gibt die Menge an verloren gegangenen Daten pro Zeiteinheit an [Chen et al., 2004]. Fehler bei einer Netzwerkübertragung können ebenfalls als Verluste interpretiert werden, wenn das aufsetzende Protokoll über keine Fehlerkorrektur verfügt. Nichtkorrigierbare, fehlerhafte Daten müssen daher gelöscht und möglicherweise erneut übertragen werden. Bereits frühe Studien im Bereich der Netzwerktechnik haben gezeigt, dass ein wichtiger Grund für reduzierten Datendurchsatz die Datenpaketverlustrate ist [Bolot, 1993].

[Briscoe et al., 2014] zeigen, dass auch Verzögerungen im Internet häufig durch Datenpaketverluste entstehen. Diese wiederum lassen sich oft auf Überlastungen an stark verwendeten Verbindungen zurückführen (ebd.). Aus diesem Grund sind Paketverluste meist miteinander korreliert und werden mit einer Häufung von weiteren Paketverlusten begleitet [Bolot, 1993]. Weitere Gründe für Datenpaketverluste sind die Folgenden:

- fehlerhaftes Routing, bei dem der Zielort nicht gefunden werden kann [Zhang, 2005]
- Ausfall von Hardware, wie Netzwerkleitungen oder Netzwerkknoten (ebd.)
- die maximale Anzahl an Hops des IP-Datagramms wurde überschritten

Verluste von Informationen entstehen auch durch die Nutzung von verlustbehafteten Komprimierungsverfahren, die die Qualität herunterfahren, wenn Paketverluste eintreten [ITU, 11/2001].

Unter Datenverlusten leiden prinzipiell alle Anwendungen: Die Datenübertragungsrate sinkt, die Verzögerungszeit erhöht sich durch Kodierung oder erneut gesendete Pakete und die Qualität von zeitkritischen Anwendungen sinkt durch fehlende Pakete, z.B. Audioübertragung. Anwendungen wie Video- oder Audioübertragungen sind in der Lage, bis zu einem gewissen Grade mit fehlenden Daten umzugehen, ohne dass der Benutzer dies bemerkt. Für andere Anwendungen ist es hingegen kritisch, wenn es zu Datenverlusten kommt und es müssen Neuübertragungen folgen.

Die Verlustrate wird als *Bit-Fehler-Rate (Bit Error Rate, BER)* angegeben. Das Maß für die BER ist *Fehler/bit* und wird durch die folgende Gleichung bestimmt [Bossert & Breitbach, 1999]:

$$BER = \frac{Anzahl\ fehlerhafte\ Bits\ pro\ Zeiteinheit}{Anzahl\ \text{übertragene}\ Bits\ pro\ Zeiteinheit} \tag{2.4}$$

2.4.5 Dienstgüteklassifizierung der Anwendungen

Wie bereits angedeutet, stellen Anwendungen unterschiedliche Anforderungen an die bereitzustellende Dienstgüte. Eine erste Einschätzung der benötigten Dienstgüte für die einzelnen Felder der Telemedizin bietet die Zeit-Raum-Matrix in Abbildung 2.2. Anwendungen, die als asynchron eingestuft werden, besitzen in der Regel weniger kritische Anforderungen als synchrone. Eine Klassifizierung der Anwendungen kann durch die Einteilung in zeitkritische Echtzeitanwendungen und zeitunkritische Nicht-Echtzeitanwendungen erfolgen.

Ein Echtzeitsystem oder Realzeitsystem ist definiert als ein System, bei dem die Korrektheit nicht nur auf dem logischen Ergebnis einer Berechnung basiert, sondern auch auf dem Zeitpunkt, an dem das Ergebnis produziert wurde [Burns & Wellings, 2001]. Applikationen, die in Echtzeit funktionieren, haben eine Frist, die einen bestimmten Zeitpunkt markiert, an dem das Ergebnis vorhanden sein muss (ebd.). So findet z.B. eine Videokonferenz in Echtzeit statt, da hier ein kontinuierlicher Datenstrom zeitgerecht zugestellt werden muss. Zu spät eingetroffene Datenpakete können nicht mehr verwendet werden und müssen verworfen werden, da der Zeitpunkt ihres Nutzens bereits verstrichen ist.

Man unterscheidet zwischen harter Echtzeit und weicher Echtzeit [Burns & Wellings, 2001]:

- bei der *harten Echtzeit* sind die Fristen kritisch und müssen in jedem Fall eingehalten werden (ebd.). Die Überschreitung bedeutet einen fatalen Fehler für das System. Die Fernsteuerung einer Maschine fällt in die Richtung der harten Echtzeit;
- bei der *weichen Echtzeit* existieren ebenfalls Fristabläufe, bei denen es wünschenswert wäre, wenn sie eingehalten würden, es entstünde allerdings auch kein Nachteil, wenn die Frist hin und wieder überschritten würde (ebd.). Auch ein Qualitätsverlust der Daten könnte als Ausgleich einer zu hohen Überschreitung der Frist akzeptabel sein. Bei einer normalen Videokonferenz kann in der Regel von einer weichen Echtzeit gesprochen werden. In der Literatur wird dies auch *Interaktive Multimedia Applikation* genannt [Fluckiger, 1995];
- Anwendungen, die nicht von einem fristgerechten Ergebnis abhängen, sind *Nicht-Echtzeitapplikationen*. Das Hochladen einer Datei findet z.B. in der Regel nicht in Echtzeit statt, solange kein Prozess zeitkritisch davon abhängt.

Eine weitere Einteilung erfolgt gemäß der Symmetrie einer Netzwerkanwendung [Chen et al., 2004]. Ist eine Anwendung symmetrisch, dann sind Anfragen und Rück-

meldungen in ihrer Ressourcenverwendung vergleichbar (ebd.). Beide involvierten Standorte besitzen dann den gleichen oder einen ähnlichen Ressourcenbedarf. Zum Beispiel gibt es bei einer Videokonferenz zwei Videodatenströme, die zwischen den beiden Seiten gesendet werden. Bei einer asymmetrischen Verbindung ist der Ressourcenbedarf unterschiedlich. Ein Beispiel wäre hierfür das Video-Broadcasting, das nur von einem Standort aus gesendet wird (unidirektionale Video-Verbindung).

Der Fernzugriff eines Desktops oder auf Daten eines Computers erfordern unter den meisten Bedingungen nur weiche Echtzeit, falls es keine zeitkritischen Anwendungen auf dem entfernten Computer zu steuern gibt. Der Empfang von Sensordaten (Telemetrie) erfolgt in der Regel über harte Echtzeit, wenn der Datenstrom nicht unterbrochen werden darf und kontinuierlich sein muss.

	Harte Echtzeit	Weiche Echtzeit	Nicht-Echtzeit
asymmetrisch	• Maschinensteuerung • Telemetrie	• unidirektionale Video-/Audio-Sendung	• E-Mail • Transfer von Daten
symmetrisch		• Video-Konferenz • Audio-Konferenz • Chat • Desktop-Konferenz	

Abbildung 2.4: Echtzeit und Symmetrie von Netzwerkanwendungen

Abbildung 2.4 zeigt eine Einordnung verschiedener Anwendungen, die für den Bereich der Telemedizin relevant sind, in die oben genannten Klassifizierungen für Echtzeit und Symmetrie.

2.4.6 Anwendungen und ihre Dienstgüte-Metriken

Bei der Telemedizin müssen Anwendungen, die in einem telemedizinischen Szenario eingesetzt werden, bestimmte Anforderungen erfüllen können. Die Anforderungen richten sich nach den vier oben genannten Parametern und können je nach Anwendungsart sehr unterschiedlich ausfallen. Telemedizinische Anwendungen besitzen ein breites Spektrum hinsichtlich der erforderlichen Dienstgüte. Diese wird meist durch den Anwender bzw. durch die Grenzen und Bedürfnisse der menschlichen Wahrnehmung bei der Nutzung der Anwendung bestimmt. Die menschliche Wahrnehmung gestaltet sich individuell und muss mithilfe von Studien in Hinblick auf die Anwendungen beurteilt werden [ITU, 05/2003]. Das Ergebnis ist ein Spektrum, welches auf der Zufriedenheit der Anwender beruht. Im Folgenden wird auf mehrere Anwendungen der Telemedizin eingegangen und die Dienstgüteanforderungen werden erörtert.

2.4.6.1 Sprachkommunikation

Eine Sprachkommunikation läuft in Echtzeit so ab, dass mindestens zwei Benutzer miteinander verbal kommunizieren. Bei einer bidirektionalen Übertragung werden zur selben Zeit beide Audiokanäle übertragen. Ein Benutzer ist in der Lage, dem anderen direkt zu antworten oder ihn zu unterbrechen.

Verzögerungen im Bereich der Sprachübermittlung sollten möglichst kurz sein, da sie sich sonst auf die Qualität der Konversation auswirken können. Die Auswirkungen machen sich in zwei Formen bemerkbar [ITU, 11/2001]:

1. bei akustisch gekoppelten Endgeräten kann ein Echo produziert werden.
2. der Konversationsablauf verschiebt sich.

Ein Echo akustisch gekoppelter Endgeräte (Mikrofon und Lautsprecher) ist dann bemerkbar, wenn die Verzögerung größer als *24 ms* ist (ebd.). Ein Echokompensator, der mithilfe eines Matched-Filters identische vergangene Audiosignale aus den aktuellen Audiodaten herausfiltert, kann zur Minimierung dieser Störung eingesetzt werden (ebd.).

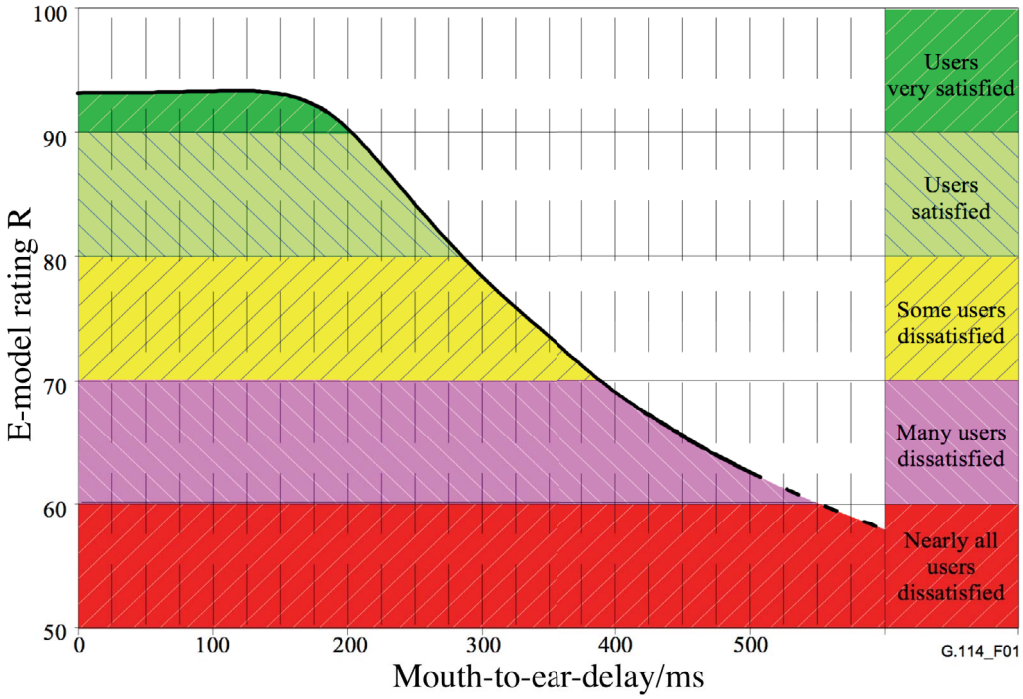

Abbildung 2.5: Effekte der Reaktionszeit bei Audioanwendungen auf den Benutzer[ITU, 05/2003]

Bei dem zweiten Effekt sind Verzögerungen nicht spürbar, solange nur ein Benutzer spricht. Dies ändert sich, wenn ein Benutzer auf die Antwort des anderen wartet oder ein Benutzer den anderen unterbrechen möchte. Ein direktes Unterbrechen der anderen Person gelingt nicht oder nur schwer, wenn die Verzögerungen ein gewisses Maß erreicht haben. Unerwünschte Pausen können entstehen, wenn ein Benutzer sein Gesagtes beendet hat und auf eine Reaktion des anderen wartet. Die International Telecommunication Union (ITU) [ITU, 2017] hat ein Modell für die Bewertung von Verzögerungs-

zeiten bei der Übertragung von Sprache entwickelt. Abbildung 2.5 zeigt die Anwenderzufriedenheit einer entsprechenden Testreihe (*E-model rating R*) bei verschiedenen Ende-zu-Ende-Verzögerungen [ITU, 05/2003].

Wie in Abbildung 2.5 zu sehen ist, sind interaktive Sprachanwendungen bei einer Verzögerung von bis zu *200 ms* unproblematisch. Darüber hinaus sind Verzögerungen bis zu *280 ms* nach wie vor zufriedenstellend. Der gerade noch erträgliche Bereich liegt bei *400 ms*, bei dem eine obere Grenze verläuft (ebd.). Eine empfohlene Grenze für telemedizinische Aktivitäten in Notfallsituationen liegt bei *150 ms* [Skorin-Kapov & Matijasevic, 2010].

Im Vergleich zu anderen Informationsarten reagieren Echtzeit-Audio-Anwendungen am empfindlichsten auf Jitter [Chen et al., 2004]. Dieser ist dort am leichtesten zu bemerken (ebd.). Jitter macht sich innerhalb von Audio durch eine Veränderung in der Tonfrequenz bemerkbar. Ein starker Jitter bei digitalen Audioübertragungen führt zu Aussetzern. Jitter sollte für Audioübermittlungen im besten Falle nach [Tobagi, 2005] *50ms* nicht übersteigen.

Die benötigte Datenübertragungsrate einer Audioübertragung hängt stark von der gewünschten Qualität und dem verwendeten Audiocodec ab. Die Datenmenge für eine Sprachübertragung beträgt zwischen *5 Kbit/s* bei Verwendung von Audiokomprimierungsverfahren und *64 Kbit/s* bei unkomprimierten „rohen" Sprachdaten. Die Verringerung der Datenübertragungsrate kann mithilfe von Audiokomprimierungsverfahren erfolgen, was allerdings zu einem Anstieg von zusätzlichen Verzögerungen führt. Eine unzureichende Datenübertragungsrate im Netzwerk führt zu starker Erhöhung der Gesamtverzögerung, des Jitters, und zu Aussetzern in der Übertragung.

Die menschliche Wahrnehmung von Verlusten ist sehr tolerant bei Audioübermittlungen [Chen et al., 2004]. Auch bei schlechten Bedingungen und hohen Informationsverlusten können Audioübertragungen noch verständlich sein. Bei Sprachübermittlungen entspricht der Verlust eines einzelnen Datenpakets in der Regel *20 ms* einer Übertragung [Tobagi, 2005]. Verluste sollten zwischenzeitlich *60 ms* nicht überschreiten [Gruber & Strawczynski, 1985]. Eine BER bei unkomprimierter Sprachübermittlung sollte daher nicht höher als 10^{-2} *Fehler/Bit* sein [Fluckiger, 1995].

Die Durchführung einer Audioübertragung und gleichzeitigem Datentransfer von Massendaten kann durch die Aggressivität des Transportprotokolls unter bestimmten Umständen zu Problemen mit der Audioübermittlung führen [Tobagi, 2005] [Järvinen et al., 2013]. In diesem Fall sollte die Übermittlung von Massendaten separat oder mithilfe eines Netzwerk-Schedulers[9] erfolgen.

E-zu-E Verzögerung	**Jitter**	**Datenübertragungsrate**	**Fehlerrate**
< 200 ms	< 50 ms	5 Kbit/s ~ 64 Kbit/s	10^{-2} Fehler/bit

Tabelle 2.1: Anforderungen von Audiokommunikation

[9] Vgl. Kapitel 3.3.1.2

2.4.6.2 Videokommunikation

Bei einer Videokonversation befinden sich mindestens zwei Benutzer vor ihren jeweiligen Video-Kommunikationseinrichtungen. Zeitgleich können sie sich gegenseitig auf einem Bildschirm sehen. Mimik, Gestik und Sprache werden an das Gegenüber übertragen. Die Videoqualität muss ausreichend sein, um dem gewünschten Zweck zu dienen. Dies ist vor allem dann wichtig, wenn ein Arzt auf der anderen Seite bestimmte Untersuchungen durchführen soll.

Eine Verzögerung des Videodatenstroms führt zu ähnlichen Phänomenen wie bei der Sprachübermittlung: Gestik und Mimik kommen verzögert an. Bei der Beobachtung von Vorgängen oder Untersuchungen liegen die beobachtbaren Aktionen verspätet vor. Dies stellt bei normaler Videokommunikation in der Regel kein Problem dar. Eine Videokommunikation ist meist mit einer Audioübertragung verknüpft. In diesem Fall fällt eine Einflussnahme auf die menschliche Wahrnehmung bei Video geringer aus als bei Audioübertragungen.

Bei der Videokommunikation ist vor allem eine Synchronisation der Lippen von hoher Wichtigkeit [Fluckiger, 1995]. Diese sogenannte Intermedia-Synchronisation [Zuberbühler et al., 2002] sollte unterhalb einer Verzögerung von *100 ms* bleiben [Fluckiger, 1995]. Alles in allem ergibt sich daraus für das reine Video eine Gesamtverzögerung von *250 ms* bei einem maximalen Optimum der Audiosignale von *200 ms*, wenn es nicht mit den nebenbei gesendeten Audiosignalen synchronisiert ist. Bei einer Audio-Video-synchronisierten Übertragung gelten die oben genannten Zahlen für die Sprachkommunikation.

Jitter führt bei Video zu Variationen in der Abspielgeschwindigkeit oder kurzzeitigen Fehlern im Bild. Ein gleichzeitiges Übertragen von Audiosignalen führt dazu, dass Jitter im Videodatenstrom weniger auffällt als innerhalb der Audiosignale (ebd.). Die Lippen-Synchronisation fällt hier am deutlichsten ins Gewicht. Die Wahrnehmung einer Verzögerungsveränderung steigt mit zunehmender Qualität des Videos. Die Variation der Verzögerung sollte bei einer Qualität in High Definition (1920 x 1080 Pixel) nicht mehr als *50 ms* betragen (ebd.). Niedrigere Auflösungen setzen ihre Grenzen bei *100 ms* (ca. 800 x 600 Pixel) und *400 ms* (ca. 300 x 350 Pixel) (ebd.).

Die Datenübertragungsrate unterscheidet sich deutlich in Hinblick auf die verschiedenen Qualitäten, Auflösungen und (verlustbehafteten) Komprimierungsverfahren. Diese können zwischen *64 Kbit/s* und ca. *10 Mbit/s* betragen [Tobagi, 2005]. Heutige Video-Komprimierungsverfahren sind in der Lage, Datendurchsatzengpässe im Netz zu erkennen und Qualität und Auflösung automatisch anzupassen. Eine Verringerung der Qualität erscheint dem Betrachter als eine Vergröberung des Bildsignals. Kleinere Datenübertragungsraten verursachen Abstriche bei der Qualität und resultieren im schlimmsten Fall in einem stockenden Video oder fehlenden Zwischenbildern.

Ein Verlust von Datenpaketen kann zu großflächigen Pixelfehlern führen oder sogar zu fehlenden Zwischenbildern. Je nach eingesetzter Komprimierungsmethode können fehlende Zwischenbilder dazu führen, nachfolgende Bildsequenzen zu stören. Eine Da-

tenpaketverlustrate von *3 %* kann sich bereits auf *30 %* der Rahmen auswirken [Boyce & Gaglianello, 1998]. Bei einer hohen Komprimierung ist es deshalb besser, wenn die BER so klein wie möglich ist. Die BER bei Videokommunikation sollte 10^{-4} Fehler/bit nicht überschreiten, wenn die Qualität einen hohen Stellenwert sowie mindestens 10^{-2} Fehler/bit bei geringen Qualitätsanforderungen besitzt.

Videoübertragung und gleichzeitiger Datentransfer von Massendaten führt zu Problemen bei der Videoübermittlung [Tobagi, 2005]. Die Übermittlung von Massendaten sollte daher separat oder mithilfe eines Netzwerk-Schedulers durchgeführt werden.

E-zu-E Verzögerung	**Jitter**	**Datenübertragungsrate**	**Fehlerrate**
< 300 ms	< 50 ms ~ < 400 ms	64 Kbit/s ~ 10 Mbit/s	10^{-4} Fehler/bit

Tabelle 2.2: Anforderungen von Videokommunikation

2.4.6.3 Textkommunikation (Chat)

Bei der Textkommunikation wird eine Textzeile von einem Benutzer A an einen anderen Benutzer B gesendet. Benutzer B liest den Text und kann darauf antworten. Er kann wiederum eine Textzeile schreiben und sie zurück an Benutzer A senden. Unter der Voraussetzung, dass bestimmte Konzepte der Awareness[10] im Programm implementiert sind, können die Benutzer beide verfolgen, wenn der jeweils andere in dem Moment etwas schreibt. Ein Gefühl der direkten Kommunikation wird aufrechterhalten und zugleich die Erwartung auf Antwort, wenn ein Benutzer sieht, dass der andere nicht mehr schreibt. In anderen Konzepten wird die abgeschickte Textnachricht von Benutzer A erst dann bei ihm selbst eingeblendet, wenn ein Bestätigungspaket vom anderen Endpunkt aus zurückgesendet wurde. Das Bestätigungspaket enthält die Daten für die Bildschirmanzeige.

Für eine Textkommunikation liegt die erwartete Verzögerung in diesem Fall höchstens bei einer Sekunde, bis der Benutzer, der die Nachricht abgesendet hat, eine Reaktion, d.h. eine Bestätigung der Ankunft erwartet [Chen et al., 2004]. Eine größere Zeitspanne ist für den Benutzer zwar nicht optimal, stellt aber in der Regel kein Problem dar. Jitter spielt für die Textkommunikation keine Rolle, da Pakete unregelmäßig und in nicht wahrnehmbarer Periode eintreffen (ebd.).

Eine Textnachricht kann in der Regel innerhalb eines einzelnen Datenpakets untergebracht werden. Die Datenübertragungsrate beträgt höchstens *1 Kbit/* (ebd.). Ein durch das Netzwerk eingeschränkter Datendurchsatz verlangsamt die Übertragung und damit den Kommunikationsfluss.

Datenverlust ist für eine Textkommunikation nicht tragbar. Sie muss daher Null betragen und wird durch darunterliegende zuverlässige Protokolle durch erneute Paketsendungen kompensiert.[11] Wenn eine Rückbestätigung vom Empfänger von Bedeutung

[10] Vgl. Kapitel 2.3.1
[11] Vgl. Kapitel 3.1.3

ist, dann sollte eine minimale Ende-zu-Ende-Verzögerung von *200 ms* nicht überschritten werden, um unter der zufriedenstellenden Reaktionszeit zu bleiben (ebd.).

Reaktions-zeit	E-zu-E Verzögerung	Jitter	Datenüber-tragungsrate	Fehlerrate
< 1 s	< 200 ms	unkritisch	~ 1 Kbit/s	0

Tabelle 2.3: Anforderungen von Textkommunikation

2.4.6.4 Datenübertragung von Massendaten

Bei der Datenübertragung von Massendaten lädt ein Benutzer eine Datei von einem entfernten Endpunkt herunter oder hoch. Die Größe der Datei ist durch die Anwendung gegeben und kann in der Telemedizin sehr schwanken. Bilder oder Patientendaten können relativ kleine Datenmengen im Bereich zwischen *20 KB* und *1 MB* besitzen. Hochauflösende Fotos im unkomprimierten Format dagegen können Größen von *25 MB* überschreiten und Videoaufnahmen in der Diagnostik bis zu mehrere Gigabyte. Beispiele für Bildaufnahmen sind Röntgenbilder, Magnet-Resonanz-Aufnahmen, Ultraschall- und Computer-Tomografie-Bilder. Eine Datenübertragung besitzt keine Echtzeit-Ansprüche. Zu Massendatenübermittlung zählen ebenfalls E-Mail-Sendungen, FTP-Server-Zugriffe und Cloud-Datensynchronisierung.

Verzögerungen hängen von der vom Netzwerk bereitgestellten Datendurchsatzrate ab. Diese sollte so hoch wie möglich sein, um die Übertragungszeit zu minimieren. Eine Verlustrate erhöht zusätzlich die Zeit zur Übertragung der Daten, da sie durch erneutes Senden kompensiert werden muss. Eine Reaktionszeit beim Benutzer, d.h. die Zeit, die der Benutzer maximal warten will, bis er sieht, dass die Datenübertragung startet, liegt zwischen *2* und *5 Sekunden*. [Chen et al., 2004]

Wie bereits bei Audio- und Videoübermittlung erwähnt, sollte eine Datenübertragung von Massendaten nur unter bestimmten Umständen, d.h. mit eingeschränkter Datenübertragungsrate, zeitgleich mit einer Audio- und Videoübermittlung durchgeführt werden [Tobagi, 2005]. Eine Massendatenübertragung über das dafür vorgesehene Transportprotokoll versucht, stets ein Maximum des möglichen Datendurchsatzes zu nutzen[12], was bei variablen Bitraten von Multimedia-Datenströmen zu Problemen führen kann.

Reaktionszeit	Jitter	Datenübertragungsrate	Fehlerrate
< 2 s – 5 s	unkritisch	möglichst hoch / anpassbar	0

Tabelle 2.4: Anforderungen von Massendatenübertragung

2.4.6.5 Videosendung (unidirektional)

Bei einer unidirektionalen Videosendung wird ein Videostrom von einem Endpunkt aus an einen oder mehrere Benutzer gesendet (Broadcasting). Die Videoqualität hängt von der gewünschten Anwendung ab. Wenn der Benutzer über eine Schnittstelle ver-

[12] Vgl. Kapitel 3.3.6.3

fügt, um in den entfernten Realraum einzugreifen, wie z.B. beim ARTP, sollte der Videodatenstrom mit der Schnittstelle möglichst synchron sein. In diesem Fall hängen die Anforderungen von der Anwendung mit den höchsten Ansprüchen ab.

In der Regel werden unidirektionale Videosendungen für Beobachtungs- bzw. Lehrzwecke benutzt oder in Verbindung mit einem Rückkanal, welcher Anweisungen oder das Steuern von Apparaturen erlaubt. Eine unidirektionale Videosendung stellt vor allem Anforderungen an die Datenübertragungsrate. Wie bei der Videokonversation ist sie maßgebend für die mögliche Qualität und Auflösung. Eine unkomprimierte Videosendung in High Definition Qualität (1920 x 1080) kann Datenübertragungsraten bis zu *1 Gbit/s* aufweisen.

Solange es keinen Rückkanal gibt, ist die Ende-zu-Ende-Verzögerung unwesentlich. Es geht vielmehr um die Reaktionszeit, die der Benutzer beim Aktivieren des Videos erwartet. Diese sollte nicht höher als *2 s* bis *5 s* sein [Chen et al., 2004]. Zusätzliche Ansprüche können an den Jitter gestellt werden, da nicht nur etwaige Lippenbewegungen verfolgt werden müssen, sondern auch andere Bewegungen, bei denen es von höchster Wichtigkeit ist, dass sie klar und korrekt wahrgenommen werden. Jitter-Wahrnehmungstests bei Fußballübertragungen lassen z.B. bereits Jitter mit *5 ms* erkennen [Minhas et al., 2012]. Eine weitverbreitete Empfehlung für Videosendungen in HD liegt bei < *50 ms* [Fluckiger, 1995].

Fehlerraten bei Videosendungen in HD fallen kleiner aus als bei Videokonferenzen. Diese liegen bei 10^{-5} *Fehler/Bit* bei einer Fehlerdauer von maximal *20 ms* [Fluckiger, 1995]. Verfahren für Fehlerkompensation innerhalb des Videocodecs sind angeraten.

Reaktionszeit	Jitter	Datenübertragungsrate	Fehlerrate
< 2 s – 5 s	< 5 ms ~ < 100 ms	64 Kbit/s ~ 1 Gbit/s	10^{-5} Fehler/bit

Tabelle 2.5: Anforderungen von unidirektionalen Videoübertragungen

2.4.6.6 Maschinensteuerung

Eine Maschinensteuerung in Echtzeit ist asymmetrisch, d.h., sie wird nur von einem Endpunkt aus durchgeführt. Hierbei steuert ein Benutzer eine Maschine aus der Ferne. Die Teleoperation ist Teil dieser Anwendungsklasse. Möglicherweise werden die Auswirkungen der Steuerung auf die Maschine mithilfe einer unidirektionalen Videoübertragung beobachtet. Auf eine Änderung im Realraum der Maschine muss der Benutzer sofort reagieren können. Die Steuerung bewirkt einen Datenstrom mit Befehlen für das Verhalten der Maschine. Eine ähnliche Konstellation ergibt sich bei der Steuerung eines Telepointers. Hier steuert der Benutzer einen Telepointer mit einem Eingabegerät und verfolgt zur gleichen Zeit die Änderungen auf dem im Bildschirm angezeigten Shared Space. Ein Telepointer wird über einen kontinuierlichen Datenstrom mit XY-Koordinaten gesteuert.

Verzögerungen spielen bei dieser Anwendungsklasse eine große Rolle. Die Übertragung der Bewegungsbefehle auf den Telemanipulator und die gleichzeitige Verfol-

gung der Bewegungen auf einem Monitor führen dazu, dass die Verzögerung, d.h. die Umlaufzeit, möglichst klein gehalten werden muss [Hanly & Broderick, 2005]. Schlechte Netzwerkkonditionen führen dazu, dass eine Steuerung nicht zufriedenstellend erfolgen kann. Das Verfolgen eines Ziels durch eine Maussteuerung wird durch hohe Verzögerungen, Verzögerungsvarianzen und Signalabbrüche erschwert und führt zu einer beeinträchtigten Treffgenauigkeit [Pavlovych & Stuerzlinger, 2011]. Während Verzögerungen und Signalabbrüche, die sich auf einen Telemanipulator auswirken, ein verspätetes Verhalten hervorrufen, führen Verzögerungsvarianzen zu einhaltendem oder ruckartigem Gebaren [Gutwin, 2001].

Studien haben gezeigt, dass die Umlaufzeit (Befehl zur Maschine und per Video zurück) *330 ms* nicht überschreiten sollte [Marescaux et al., 2001]. Ein relativ geringer Jitter von kleiner als *10 ms* ist tolerierbar, um Handbewegungen noch gut umsetzen zu können (ebd.). Beim Verfolgen eines Ziels mit einer Maus steigt die Fehlerquote der Bewegungen ab einer Umlaufzeit von über *110 ms* an [Pavlovych & Stuerzlinger, 2011]. Bewegungsfehler durch erhöhten Jitter sind signifikant ab *40 ms* (ebd.). Dies ist jedoch stark abhängig vom Schwierigkeitsgrad der zu bewältigenden Aufgabe (ebd.).

Um Probleme bei größeren Verzögerungen zu minimieren, können unterschiedliche Techniken angewandt werden. Eine Möglichkeit besteht darin, die erlaubten Bewegungen stark zu verlangsamen. Alternativ wird eine geschickte Arbeitsteilung mit einem an der entfernten Seite befindlichen Kollegen koordiniert – in Kombination mit einer zusätzlich geschalteten Videokommunikation [Skorin-Kapov & Matijasevic, 2010]. Weitere Möglichkeiten sind Bewegungsvorhersagen oder Interpolation der Bewegungen.

Befehle und der daraus folgende Datenstrom, um z.B. einen telemedizinischen Roboter zu steuern, sind relativ klein. Datenübertragungsraten von höchstens *50 Kbit/s* sind in diesen Fällen üblich [Szymanski, 2010]. Eine Verlustrate von Daten ist nicht zulässig, so dass der Datenstrom abgesichert werden muss, um Fehler zu vermeiden. Ein erneutes Senden bei fehlerhaften oder verlorenen Datenpaketen ist mit erhöhten Ende-zu-Ende-Verzögerungen verbunden.

Umlaufzeit	**Jitter**	**Datenüber-tragungsrate**	**Fehlerrate**
< 110 ms ~ < 330 ms bzw. E-zu-E: < 55 ms ~ < 165 ms	< 10 ms ~ < 40 ms	~ 50 Kbit/s	0

Tabelle 2.6: Anforderungen von Maschinensteuerung

2.4.6.7 Telemetrie

Die Telemetrie ist die „Übertragung elektrischer Messwerte auf drahtgebundenem oder drahtlosem Weg“ [Brockhaus Enzyklopädie Online, 2017]. In der Telemedizin kommt es häufig vor, dass Sensor-Daten wie z.B. von einem Elektroenzephalogramm (EEG) oder einem Elektrokardiogramm (EKG) zur Anwendung gesendet werden sollen.

Die Daten können direkt an einen entfernten Benutzer versendet werden. Die Anwendung ist daher asymmetrischer Natur. Bei einer echtzeitigen Übertragung wird durch die Sensoren ein stetiger Datenfluss bestehend aus Messwerten produziert. Wird Telemetrie innerhalb einer medizinischen Behandlung eingesetzt, so müssen die Werte mit anderen Anwendungen synchron sein.

Nach [Skorin-Kapov & Matijasevic, 2010] sind vier Klassen mit unterschiedlichen Stärkegraden in Dienstgüte-Anforderungen zu definieren:

1. Klasse 0: höchste Priorität für Echtzeit Überwachung (vor allem in Notfallsituationen) und stetiger Datenstrom
2. Klasse 1: nahe-Echtzeit Überwachung (mehrere Male pro Stunde)
3. Klasse 2: periodische Überwachung (mehrere Male am Tag)
4. Klasse 3: zeitweilige Überwachung

Bei dieser Klasseneinteilung ist zu beachten, dass die Datenströme in ihrer Form variieren. Bei Klasse 0 ist es so, dass ein stetiger Datenstrom mit Messungen notwendig ist. Bei anwachsender Klasse sinken die Anzahl der Messungen und damit auch die Anforderungen an die Datenübertragungsrate und etwaige Verzögerungen. Bei Klasse 0 liegen die Empfehlungen nach [Chen et al., 2004] bei höchstens *250 ms* als Ende-zu-Ende-Verzögerung mit einem Jitter nicht größer als *100 ms*.

Datenübertragungsraten für medizinische Überwachungsgeräte können stark variieren. Auch Videoüberwachung oder Maschinen- bzw. Prozesssteuerung kann ebenfalls unter Telemetrie fallen und zum Teil hohe Datenübertragungsraten produzieren. Die Datenübertragungsraten für den Empfang von Datenströmen telemedizinischer Überwachungsgeräte in Echtzeit, wie z.B. eines EKGs, liegen unterhalb von *120 Kbit/s*. [Skorin-Kapov & Matijasevic, 2010] Die Verlustrate der Datenpakete muss *0* betragen.

E-zu-E Verzögerung	**Jitter**	**Datenübertragungsrate**	**Fehlerrate**
< 250 ms	< 100 ms	~ 120 Kbit/s	0

Tabelle 2.7: Anforderungen von Telemetrie

2.4.6.8 Immersive haptische Übertragung

Eine haptische Übertragung ist in der Telemedizin die Übertragung von Druck- oder Impulsempfindungen [Kurogi et al., 2013]. Ein aus Telemetrie-Werten bestehender Datenstrom wird an den haptischen Aktuator auf der Benutzerseite gesendet. Der Benutzer empfängt die Druckempfindungen und kann darauf reagieren.

Haptische Aktuatoren werden in der Telemedizin eingesetzt, um ein besseres Gefühl und bessere Kontrolle bei Tele-Operationen während der Bedienung eines telemedizinischen Roboters zu ermöglichen [Fischer & Voges, 2011]. Eine haptische Übertragung wird daher in der Regel in Kombination mit anderen Anwendungen wie der Maschinensteuerung verwendet. Eine Unterbrechung des Datenstroms führt zu einem kurzzeitigen

Aussetzen des Aktuators. Diese Systeme zählen zu den kritischsten Anwendungen in der Telechirurgie.

Nach [Heller & Schiff, 1991] ist die Berührungsempfindlichkeit zwanzig Mal schneller als das Auge. Ein Mensch kann zwischen Verzögerungen, die *5 ms* auseinanderliegen unterscheiden [Saddik et al., 2011]. Eine Verzögerung ist daher bei haptischen Anwendungen kritisch. Eine Empfehlung für die Ende-zu-Ende-Verzögerung für teleimmersive Systeme mit haptischem Interface durch [Berliner et al., o.D.] liegt bei *10 ms*. [Jay & Hubbold, 2006] schlagen einen Schwellwert von *45 ms* als Umlaufzeit vor, bei dem ein haptisches Feedback keinen Nutzen mehr besitzt. Ein Jitter darf bei dieser geringen Verzögerung ebenfalls nur minimal sein, z.B. *1 ms.*

Die benötigte Datenübertragungsrate kann derzeit bis zu *1 Mbit/s* betragen [Berliner et al., o.D.]. Die Fehlerrate eines haptischen Datenstroms sollte möglichst klein sein, z.B. 10^{-5} Fehler/Bit.

E-zu-E Verzögerung	Jitter	Datenübertragungsrate	Fehlerrate
< 10 ms ~ < 45 ms	~ 1 ms	~ 1 Mbit/s	10^{-5} Fehler/bit

Tabelle 2.8: Anforderungen von haptischer Übertragung

2.4.6.9 Desktop-Konferenz

Eine Desktop-Konferenz funktioniert mit einer Software, die in einem Webbrowser ausgeführt wird oder die auf mindestens zwei voneinander entfernten Rechnern installiert ist. Benutzer sind so in der Lage, über einen Gruppeneditor Zugriff auf verschiedene Objekte zu erhalten, diese zu manipulieren und dabei gleichzeitig mit den anderen Benutzern zu kommunizieren. Benutzer können z.B. gemeinsam auf Dokumente zugreifen und Änderungen zusammen mit anderen Benutzern durchführen.

Die Daten-Kommunikation ist symmetrisch, da alle Rechner einen gleichgestalteten Datenfluss besitzen. Durch die gegebene Gleichzeitigkeit existiert eine synchrone Arbeitsweise, die der weichen Echtzeit zuzuordnen ist. Während gleichzeitig Video- oder Audiokommunikation durchgeführt wird und die Benutzer miteinander kommunizieren, können sie die Daten auf dem virtuellen Shared Space manipulieren. Abgesehen von der Video- bzw. Audiokommunikation wird ein Datenstrom für die Manipulation der Objekte und die Koordination der getätigten Aktivitäten gesendet und empfangen. Die gemeinsame Nutzung einer Applikation fällt ebenfalls unter den Begriff Desktop-Konferenz.

Ende-zu-Ende-Verzögerungen sollten nicht größer als *200 ms* in Hinblick auf die Video- bzw. Audioübertragung sein. Da es sich hier um eine Datenübertragung von Massendaten handelt, spielt Jitter nur eine untergeordnete Rolle, die sich eher auf die Audio- bzw. Videokommunikation auswirkt. Diese besitzt keine bestimmende Rolle wie bei einer tatsächlichen Video-Konferenz.

Die Datenübertragungsrate bei der Manipulation von Objekten ist in der Regel gering und besitzt höchstens einen Wert von *120 Kbit/s*.[13] Bei der Übertragung der Daten für die Manipulation von Objekten dürfen keine Fehler auftreten, d.h. die Übertragung muss zuverlässig gestaltet sein. Die BER für die Video- und Audiokommunikation darf in diesem Fall 10^{-2} Fehler/bit betragen – gegeben durch die Audioanforderungen und einer eher vernachlässigbaren Videokommunikation [Fluckiger, 1995]. Überschreitungen der Werte sind zwar nicht wünschenswert, dürfen aber gelegentlich vorkommen.

E-zu-E Verzögerung	Jitter	Datenübertragungsrate	Fehlerrate
< 200ms	< 50ms	~ 120 Kbit/s	0, 10^{-2} Fehler/bit

Tabelle 2.9: Anforderungen von Desktop-Konferenzen

2.4.7 Zusammenfassung der Dienstgüteansprüche in telemedizinischen Anwendungen

Tabelle 2.10 zeigt eine Übersicht über die hier diskutierten Anwendungen und ihre Anforderungen. Die unterschiedlichen Verzögerungsparameter der Anwendungen beziehen sich auf die Rückkopplung mit der Gegenstelle. Für einige Anwendungen ist die Ende-zu-Ende-Verzögerung bzw. die Reaktionszeit nicht von primärer Wichtigkeit.

Anwendung	E-zu-E Verzögerung	Reaktionszeit	Jitter	Datenübertragungsrate	Fehlerrate
Audiokommunikation	< 200 ms		< 50 ms	5 Kbit/s ~ 64 Kbit/s	10^{-2} Fehler/bit
Videokommunikation	< 300 ms		< 50 ms ~ <400 ms	64 Kbit/s ~ 10 Mbit/s	10^{-4} Fehler/bit
Videoübertragung		< 2s – 5s	<5 ms ~ <100 ms	64 Kbit/s ~ 1 Gbit/s	10^{-5} Fehler/bit
Textkommunikation	< 200 ms	< 1s	unkritisch	~ 1 Kbit/s	0
Massendaten-Übertragung		< 2s – 5s	unkritisch	möglichst hoch / anpassbar	0
Maschinensteuerung	< 55 ms ~ < 165 ms		< 10 ms ~ < 40 ms	~ 50 Kbit/s	0
Telemetrie	< 250 ms		< 100 ms	~ 120 Kbit/s	0
Haptische Übertragung	< 10 ms ~ < 45 ms		~ 1 ms	~ 1 Mbit/s	10^{-5} Fehler/bit
Desktop Konferenz	< 200 ms		< 50 ms	~ 120 Kbit/s	0 und 10^{-2} Fehler/bit

Tabelle 2.10: Dienstgüteanforderungen für Anwendungen in der Telemedizin

2.5 Telemedizinisches Basisszenario zwischen UDE und UKM

Das telemedizinische Basisszenario zwischen UDE und UKM folgt den folgenden Arbeitsgebieten:

- Teleausbildung
- Telediagnostik

[13] eigene Schätzung durch Tests mit Desktop-Konferenzsystemen: [WebEx, 2017], [TeamViewer, o.D.], Passenger [Werner, 2002]

- Teletherapie
- Tele-Operation

Die verschiedenen Anwendungsfelder werden mithilfe eines ARTP umgesetzt. Der ARTP bietet eine leicht zu handhabende Technik, mit der entsprechende Versuche und Evaluationen möglich werden. Speziell auf dem Gebiet der Tele-Operation bietet der ARTP anstatt eines komplexen Geräts, wie einem chirurgischen Roboter, eine Hilfestellung bei Nichtvorhandensein eines lokalen Spezialisten.

Von den vorgestellten Anwendungen der Telemedizin erfordert die Tele-Operation die komplexeste Anordnung. Die Telediagnose und die Teletherapie folgen einem einfachen Aufbau und einem vergleichsweise schlichten Arbeitsablauf. Die Teleausbildung besitzt fast einen ähnlichen Aufbau wie die Tele-Operation, ist jedoch weniger kritisch hinsichtlich der Dienstgüte.

2.5.1 Aufbau des telemedizinischen Systems

Das hier vorgestellte telemedizinische System setzt sich aus mindestens zwei Standorten bzw. Endpunkten zusammen. Am ersten Endpunkt befindet sich ein Untersuchungsraum bzw. ein Operationssaal mit einem Patienten und einem bis mehreren Assistenzärzten. Dieser Standort wird bei dem hier beschriebenen telemedizinischen System auch als lokaler Endpunkt bezeichnet, da die Aktivität, welche am Patienten durchgeführt wird, im Mittelpunkt der Anwendung steht. Weitere Standorte sind externe Endpunkte, an denen sich Experten mit besonderem Fachwissen befinden. Der ARTP befindet sich am lokalen Endpunkt in der Nähe des Patienten und wird vom externen Endpunkt aus gesteuert.

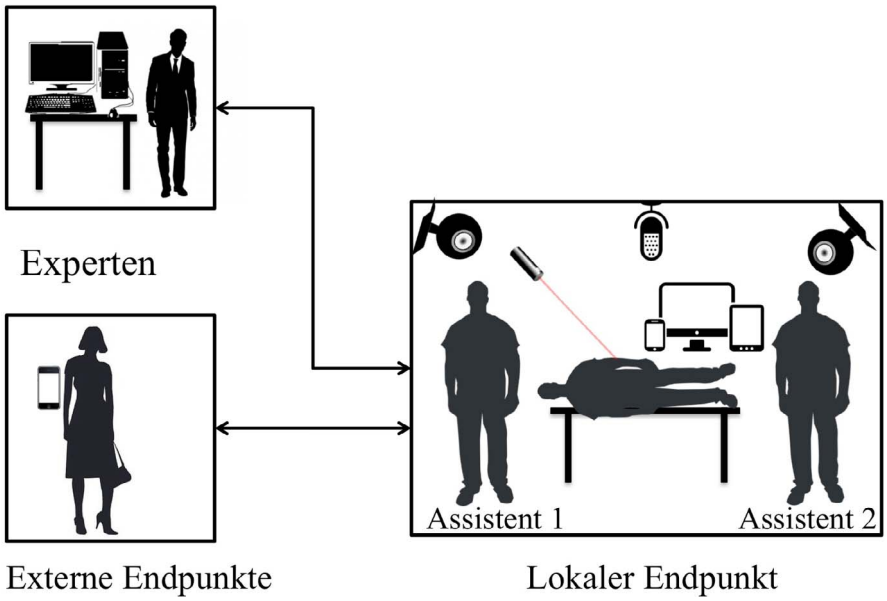

Abbildung 2.6: Telemedizinisches Basisszenario zwischen UDE und UKM

Bei den verwendeten Dienstarten wird eine Behandlung aus der Ferne durchgeführt oder zumindest durch Experten beratend betreut. Es werden hohe Ansprüche an die Dienstgüte gestellt, da die Gesundheit bzw. das Leben des Patienten maßgeblich von der

Verbindung selbst abhängen. Der in Abbildung 2.6 dargestellte Aufbau zeigt eine schematische Darstellung des Systems im Basisaufbau.

Der Behandlungsraum ist mit verschiedenen Gerätschaften ausgestattet, die zur Telekonferenz verwendet werden. Dazu zählen Kameras, Mikrofone sowie Ausgabegeräte wie Lautsprecher und Monitore. Innerhalb des Behandlungsraums befindet sich ein Patient. Der ARTP wird durch die Experten aus der Ferne als Zeigegerät verwendet. Die Experten sind damit in der Lage

- die Behandlung über Video (mehrere Kameramöglichkeiten) zu verfolgen;
- mit den Assistenzärzten über Audioverbindung zu konferieren;
- den ARTP aus der Ferne zu steuern, um den Assistenzärzten Anweisungen anzuzeigen;
- sich Sensordaten wie z.B. EKG in Echtzeit anzeigen zu lassen;
- auf wichtige Daten des Shared Space zuzugreifen, wie z.B. auf
    - die Patientenakte,
    - oder eine Informationsdatenbank.

Eine Klassifikation der Applikationen für das Basisszenario erfolgt mithilfe der Zeit-Raum-Matrix nach [Johansen, 1991]. Die Applikationen werden dazu in Abbildung 2.7 den entsprechenden Quadranten zugeordnet. Es zeigt sich, dass die Ausrichtung des Szenarios synchroner Natur ist. Eine Ausnahme bildet der Zugriff auf den gemeinsamen Daten-Raum sowie auf die Patientendaten, die bereits vorher aufgenommen bzw. in Form einer Akte zu einem früheren Zeitpunkt erstellt wurden.

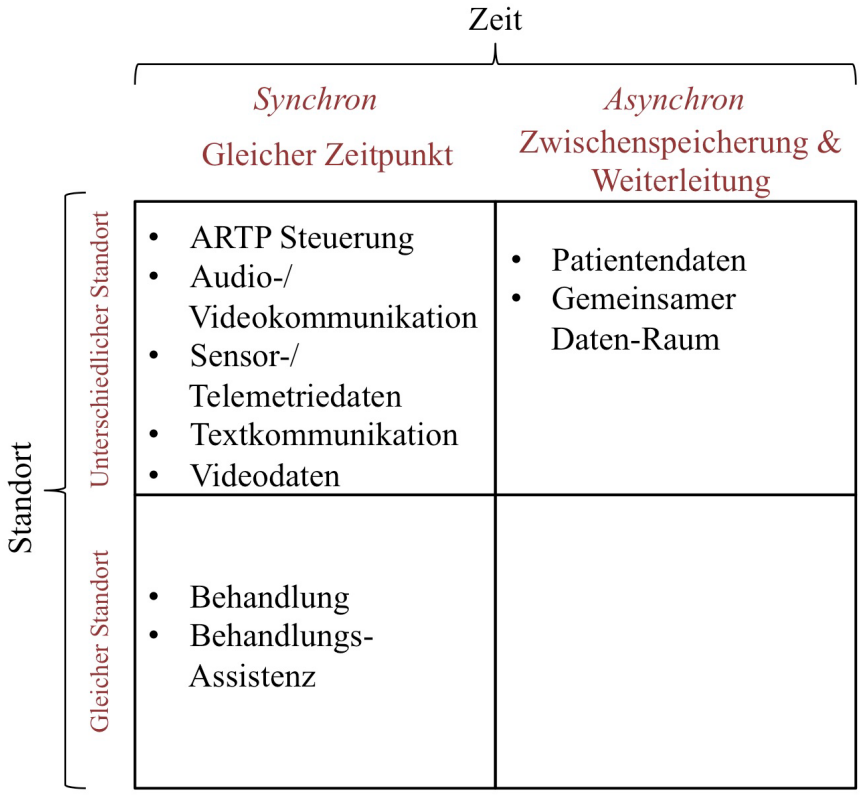

Abbildung 2.7: Zeit-Raum-Matrix nach [Johansen, 1991] mit Einordnung des Basisszenarios

Laut der in Abbildung 2.2 entwickelten Matrix können die Patientendaten auch der „Nahen Echtzeit“ zugeordnet werden, da sie möglicherweise kurz vor der Behandlung erhoben wurden. In Abbildung 2.6 kann eine Standortunterscheidung beobachtet werden, die die Aktivitäten zwischen den Akteuren im Behandlungsraum, also auf lokaler Ebene, und die Aktivitäten zwischen entfernten Akteuren trennt.

2.5.2 Anforderungen an die Dienstgüte im telemedizinischen Basisszenario

Das telemedizinische Basisszenario zwischen UDE und UKM stellt bestimmte Anforderungen, welche sich aus den folgenden involvierten Anwendungen ergeben:

- Video-/Audiokommunikation
- Unidirektionale Videosendung (zumindest des ARTP)
- Maschinensteuerung des ARTP
- Sensor- und Telemetrie-Übertragung
- Textkommunikation
- Datenübertragung

Diese Anwendungen besitzen zum Teil höchst unterschiedliche Anforderungen. In Abbildung 2.7 konnte mithilfe der Zeit-Raum-Matrix bereits eine Einordnung in zeitlich kritische bzw. unkritische Übertragungen erfolgen. Diese werden maßgeblich über Anforderungen hinsichtlich von Latenzzeiten und Jitter bestimmt. Weitere Einordnungen erfolgen durch die anderen Dienstgüte-Parameter: Datenübertragungsrate und Datenverlustrate. Anwendungen mit einer festen Datenübertragungsrate sind hierbei kritisch zu betrachten, da ein bestimmter Datenstrom vorausgesetzt wird und möglichst nicht unterbrochen oder verringert werden darf.

Anwendung	**Dienstgüte Anforderungen**			
	Datenübertragungs-rate	**Ende-zu-Ende Verzögerung**	**Jitter**	**Fehlerrate**
Audiokommunikation	5 Kbit/s ~ 64 Kbit/s	< 200 ms	< 50 ms	10^{-2} Fehler/Bit
Video-/ Audiokommunikation	64 Kbit/s ~ 10 Mbit/s	< 300 ms	< 50 ms ~ < 400 ms	10^{-2} Fehler/Bit
Unidirektionale Videosendung	64 Kbit/s ~ 1 Gbit/s	ARTP Anforderungen	< 5 ms ~ < 100 ms	10^{-5} Fehler/bit
ARTP Maschinensteuerung	~ 50 Kbit/s	< 55 ms ~ < 165 ms	< 10 ms ~ < 40 ms	0
Sensor- und Telemetrie	~ 120 Kbit/s	< 250 ms	< 100 ms	0
Textkommunikation	< 1 Kbit/s	< 1 s Reaktionszeit	unkritisch	0
Datenübertragung	anpassbar	< 2 s – 5 s Reaktionszeit	unkritisch	0

Tabelle 2.11: Übersicht der Dienstgüte-Anforderungen im telemedizinischen Basisszenario

Tabelle 2.11 zeigt eine tabellarische Übersicht aller Dienstgüte-Anforderungen der Applikationen für das telemedizinische Basisszenario. Die hier abgebildeten Dienstgüte-Metriken wurden aus Kapitel 2.4.6 übernommen. Eine genaue Anpassung der Parameter muss speziell auf die tatsächlich verwendeten Applikationen vorgenommen werden. Die Parameter hinsichtlich der Datenübertragungsrate können sich in der Realität durch die folgenden Variablen unterscheiden:

- verwendeter Video-Codec
- erlaubte Video- und Audio-Qualität
- verwendete Sensoren zur Telemetrieübertragung
- genutztes Telepointerprotokoll

Die Ende-zu-Ende-Verzögerung der unidirektionalen Videoverbindung hängt von der Verzögerungszeit ab, die der ARTP benötigt, um eine Aktion auszuführen. Die Videoübertragung wird als Steuer- und Kontroll-Feedback für den ARTP verwendet. Ein Steuerimpuls, der an der Steuerungseinheit des ARTPs aus der Ferne ausgelöst wurde, benötigt ein bestimmtes OWD zum ARTP. Die Videoübertragung sendet die Bewegung des ARTPs zurück – wiederum für die Dauer eines bestimmten OWD. Die Verzögerungszeit des ARTPs ist also am besten mit der Umlaufzeit zu bemessen. Die Umlaufzeit beinhaltet die Ende-zu-Ende-Verzögerungen – zum einen für den ARTP und zum anderen für die Videoübertragung.

$$\textit{Umlaufzeit} < \textit{E-zu-E-Verzögerung (ARTP)} + \textit{E-zu-E-Verzögerung (Video)} \quad (2.5)$$

Da es sich bei dem Szenario generell um eine synchrone Anwendung handelt, sind zeitkritische Parameter zu beachten. Die Zeitkritikalität ergibt sich aus der benötigten minimalen Verzögerung. Diese muss vor allem beim ARTP unter allen Umständen eingehalten werden. Probleme können hierbei fehlerhafte oder verlorengegangene Daten bereiten. In diesem Fall müssen sie erneut gesendet oder Verfahren zur Wiederherstellung von Daten verwendet werden. Diese Verfahren benötigen höhere Laufzeiten und können die Verzögerungszeit erhöhen. Ein Jitter sollte ebenfalls gering gehalten werden, um eine flüssige Steuerung zu ermöglichen. Dasselbe gilt für Telemetrie-Daten, die ebenfalls hochgradig zeitkritisch sind und keine fehlerhaften Daten erlauben. Ein erneutes Senden bei verlorengegangenen Daten muss innerhalb der vorgeschriebenen Verzögerung stattfinden.

Erneute Datenübertragung bei Video- oder Audiokommunikation ist nicht nötig, jedoch sollte die Fehlerrate gering gehalten werden, um die Qualität nicht zu sehr in Mitleidenschaft zu ziehen. Bei der Textkommunikation sind zum einen die gesendeten Daten sehr klein und zum anderen ist die Zeitkritikalität nicht so hoch wie beim ARTP oder bei der Video-/ Audiokommunikation. Jitter ist hierbei unkritisch, da die Daten keinen durchgängigen Fluss zum Empfänger besitzen. Dies gilt ebenfalls für die Datenübertragung.

Ein Problem entsteht, wenn mehrere Anwendungen zur selben Zeit übertragen. Wie bereits in Kapitel 2.4.6 angedeutet, haben die unterschiedlichen Datenflüsse Einfluss

aufeinander und können zu erhöhten Verkehrsaufkommen auf dem Datenpfad führen. Vor allem Datenflüsse mit festen Datenübertragungsraten, wie es bei den Prioritäten 1 bis 4 der Fall ist, können durch ein hohes Datenaufkommen gestört werden. Für den ARTP ist dies hochgradig kritisch, Video- und Audioübertragungen können ihre Qualität bis zu einem gewissen Grad automatisch mindern. Die Fehlerraten erhöhen sich, wenn zu viele Datenpakete auf einem Datenpfad gesendet werden. Ein priorisiertes Senden innerhalb einer serialisierten Übertragung der Datenströme wäre eine Möglichkeit diese Probleme zu minimieren. Eine dahingehende Priorisierung bietet Tabelle 2.12, die eine Anordnung der Anwendungen nach ihrer Zeitkritikalität darstellt.

Priorität	Anwendung	Zeitkritikalität
1	ARTP Maschinensteuerung	Sehr sehr hoch
2	Sensor- und Telemetrie	Sehr hoch
3	Video-/Audiokommunikation	Hoch
4	Audiokommunikation	Hoch
5	Textkommunikation	Medium
6	Datenübertragung	Gering

Tabelle 2.12: Priorisierung der Übertragungen im Basisszenario

Die Priorisierung der Datenströme unter Verwendung eines Netzwerk-Schedulers ist ein Ansatz, der im Zuge der hier durchgeführten Forschung untersucht, aber nicht weiter verfolgt wurde. Er bietet jedoch die Möglichkeit, das in dieser Arbeit entwickelte Netzwerkprotokoll zu ergänzen und die Verbindung bei mehreren unterschiedlichen Datenströmen zu verbessern.

3 Grundlagen von Dienstgüte im Internet

Dienstgüte bzw. Quality of Service sind Anforderungen der Anwendung und der Bedürfnisse des Benutzers an die Übertragungsqualität [Tanenbaum & Wetherall, 2012]. Um diese in einem gegebenen Rahmen in einem Netzwerk zuzusichern, werden verschiedene Mechanismen angewandt, die Basis-Dienstgüten bereitstellen oder über ein standardisiertes Niveau heben. Diese sollen in diesem Kapitel dargestellt werden. Weitergehend werden Möglichkeiten beschrieben, die Dienstgüte weiter zu verbessern. Der aktuelle Stand der Technik wird vorgestellt.

Im Folgenden werden zunächst die grundlegenden Mechanismen von paketvermittelnden Netzwerken dargestellt. Dies beinhaltet eine kurze Beschreibung des zugrundeliegenden Architekturmodells und einiger der darin definierten Protokolle. Die Grundsätze des Routings werden beschrieben, um ein Bild der Mechanismen zu bieten, die eine Vermittlung von Daten in einem mehrfach vermaschten Netzwerk ermöglichen. Der grundsätzliche Umgang mit Datenpaketen, die weitergeleitet werden müssen, und die Entstehung von Überlastungen werden diskutiert. Verschiedene Techniken werden dargestellt, die eine Anhebung der Dienstgüte über das normale Niveau ermöglichen. Diese können zum einen vom Netzwerkanbieter genutzt werden, zum anderen ist es dem Benutzer möglich durch das Anwenden bestimmter Techniken die Leistung für gegebene Anwendungen zu verbessern.

Die Nutzung von Multihoming- und Mehrwegprotokollen ist ein Ansatz, im Netzwerk entstandene Probleme mithilfe von mehreren Verbindungen zu umgehen. Eine Einführung in die bekannten Techniken wird geboten und der momentane Stand der Technik wird dargestellt.

3.1 Dienstgüte im ISO/OSI Referenzmodell

Eine standardisierte Entwicklung von Netzwerkfunktionen muss sich nach einem bestimmten Architekturmodell richten. Eine ganze Reihe von Modellen wurde im Zuge der Entwicklung der Rechnernetze hervorgebracht. Diese parallele Entwicklung unterschiedlicher Netzwerksysteme in den 1960er und 1970er Jahren erforderte einen gemeinsamen Standard, der eine Verbindung mehrerer Netzwerke mit unterschiedlichen Technologien und Protokollen ermöglicht [Zimmermann, 1980]. Eines der wichtigsten Modelle ist daher das *OSI-Referenzmodell* (Open Systems Interconnection) [ITU, 07/1994], dessen Entwicklung als „grundlegendes Modell für die Architektur von Rechnernetzen“ [Brockhaus Enzyklopädie Online, 2017] durch die International Organization for Standardization (ISO) im Jahre 1978 begonnen wurde [Zimmermann, 1980]. Das Modell wurde 1983 durch die ITU veröffentlicht. Seine Struktur besteht aus

sieben Schichten, die eine genormte Nutzung von Netzwerkprotokollen und -diensten festlegen [Brockhaus Enzyklopädie Online, 2017].

Es ermöglicht die Nutzung verschiedener Technologien, die innerhalb einer Schicht unter Einbindung der vordefinierten Schnittstellen verwendet werden können. Eine Schicht stützt sich auf die Leistungen der darunterliegenden Schicht und bietet wiederum Leistungen für die darüberliegende an [Brockhaus Enzyklopädie Online, 2017]. Das Modell ist „[...] prinzipiell, abstrakt und allgemein gehalten, damit es verschiedene Umsetzungen zulässt“ (ebd.). Bei der Entwicklung des OSI-Referenzmodells wurden die folgenden Ziele verfolgt [Bossert & Breitbach, 1999]:

- ein System wird in so viele Schichten aufgeteilt, dass jede Schicht nur wenige, klar definierte Aufgaben wahrnehmen muss
- alle in einer Schicht enthaltenen Funktionen müssen symmetrisch auf Senderseite wie auch auf Empfangsseite in derselben Schicht vorhanden sein
- die Anzahl der Schnittstellen zwischen den Schichten soll minimal sein
- die Gestaltung von Schnittstellen zwischen den Schichten soll so einfach wie möglich gestaltet sein

In jeder der sieben Schichten des OSI-Referenzmodells sind Schnittstellen definiert, um mit darüber- bzw. darunterliegenden Schichten zu kommunizieren [ITU, 07/1994]. Diese Kommunikation wird über sogenannte *Dienstprimitive* realisiert. Dienstprimitive übernehmen die Aufgabe, Anweisungen und Bestätigungen von den anliegenden Schichten entgegenzunehmen oder weiterzugeben [Bossert & Breitbach, 1999]. Die Kommunikation zwischen unterschiedlichen Endpunkten innerhalb derselben Schicht wird über *Protokolle* realisiert (ebd.). Für eine einheitliche Terminologie im OSI-Referenzmodell auf allen Schichten werden sowohl Benutzer als auch Prozesse, Endgeräte oder Netzwerkadapter als *Entitäten* bezeichnet (ebd.). Entitäten auf derselben Schicht werden *Peer-Entitäten* genannt [ITU, 07/1994].

Jede der sieben Schichten ist für Netzwerkprotokolle nach vordefinierter Klassifikation mit bestimmter Funktionalität definiert. Eine erste Einteilung bietet sich dadurch an, die Schichten in transportorientierte Schichten (Schichten 1 bis 4) und anwendungsorientierte Schichten (Schichten 5 bis 7) einzuteilen [ITU, 07/1994]. Eine weitere Möglichkeit der Einteilung bietet eine Abspaltung von Ende-zu-Ende-Protokollen (ebd.). Diese befinden sich in den oberen Schichten 4 bis 7, da sich die dort genutzten Protokolle durch ihre Ende-zu-Ende-Kommunikation auszeichnen: Hierbei werden die für die Entität einer Schicht bestimmten Daten erst dann ausgelesen und verwendet, wenn diese durch die internen Schichten und über das Netzwerk weitergereicht wurden [Leon-Garcia & Widjaja, 2004]. Die eigentlichen Daten werden dabei nur vom Empfänger und der zu empfangenen Schicht gelesen und in der Regel nicht von anderen Netzwerkgeräten. Abbildung 3.1 zeigt die Kommunikation im OSI-Referenzmodell.

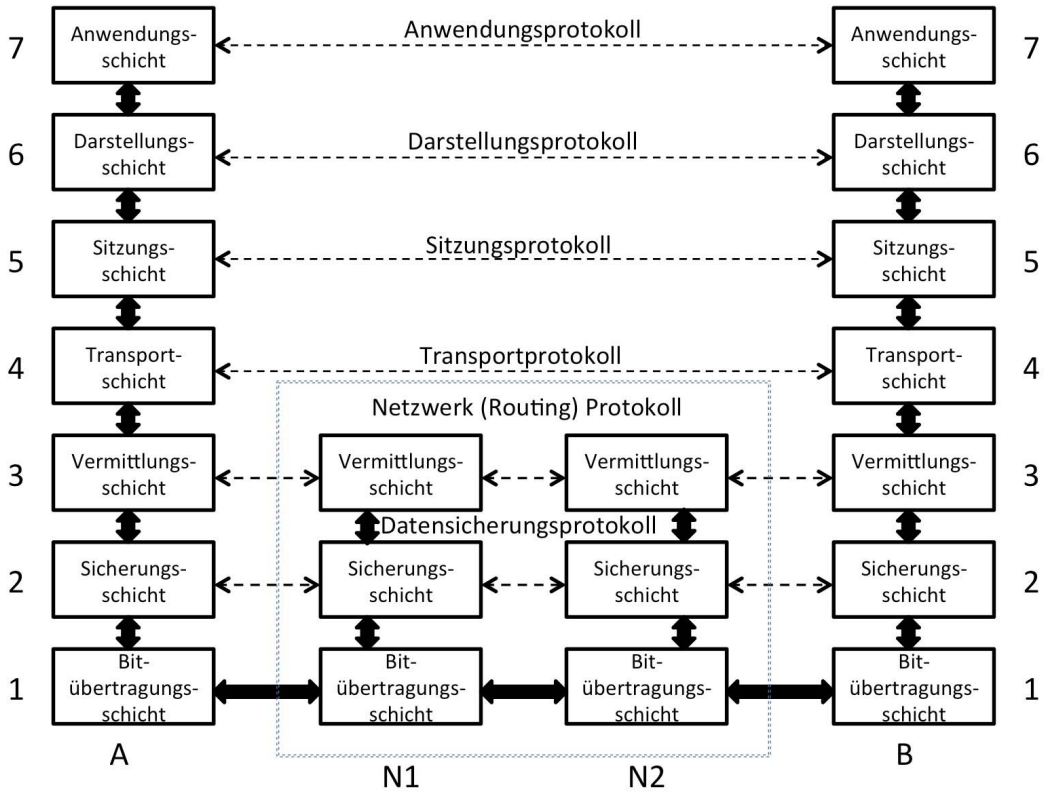

Abbildung 3.1: Kommunikation im ISO/OSI-Referenzmodell nach [ITU, 07/1994]

Die Grafik zeigt eine Anwendung der Entität A, die mit einer Anwendung der Entität B in Kontakt steht. Daten, die innerhalb einer Schicht erzeugt werden, werden mit Steuerungsdaten der jeweiligen Schicht zu einer gemeinsamen *Protocol Data Unit (PDU)* zusammengefasst und an die nächstkleinere Schicht weitergegeben [ITU, 07/1994]. Dieser Vorgang wiederholt sich, bis die Daten durch die unterste Schicht, die Bitübertragungsschicht, an die nächste Netzwerkentität gesendet werden.

Die Netzwerkentitäten N1 und N2 benötigen für die Weiterleitung eines Datenpakets Empfängerinformationen, die in der Vermittlungsschicht ausgelesen und verändert werden. Auf Seiten des Empfängers wird die PDU der untersten Schicht ausgepackt und an die nächsthöhere weitergegeben, bis die eigentlichen Daten die Anwendung erreichen. Die Ende-zu-Ende-Protokolle der beiden Entitäten kommunizieren hierbei sozusagen miteinander. Die Protokolle der unteren drei Schichten beschreiben eine Hop-zu-Hop-Kommunikation, die nur bis zur nächsten Netzwerkentität gehalten und dort für die nächste Teilstrecke geändert wird.

Das *Transport Control Protocol/Internet Protocol-Referenzmodell* (*TCP/IP)* ist ein ähnliches Modell für die Entwicklung von Protokollen aus der Internetprotokollfamilie [Braden, 10/1989]. Die Funktionsweise des Modells ist vergleichbar mit dem OSI-Referenzmodell, besitzt aber lediglich vier Schichten (ebd.). Diese vier Schichten ähneln den sieben Schichten des OSI-Referenzmodells, wobei hier die drei obersten Schichten des OSI-Referenzmodells mehr oder weniger zusammengefasst und weniger streng definiert sind und die unterste Schicht fehlt. Zudem bauen einige Protokolle der

Vermittlungsschicht auf Protokollen der selbigen auf und benötigen daher eine Art Zwischenschicht. Das Modell wurde im Rahmen der Entwicklung des Internetvorgängers ARPANET entwickelt, was dazu führte, dass die meisten heutigen Internetprotokolle auf diesem Modell basieren [Fall & Stevens, 2012]. Dennoch sind die meisten Internetprotokolle ebenfalls durch das ISO/OSI-Modell abbildbar. Abbildung 3.2 veranschaulicht die Analogien und Bezeichnungen der Schichten.

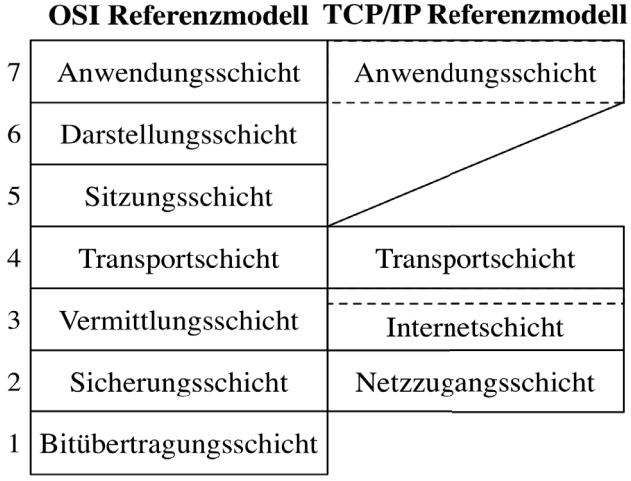

Abbildung 3.2: OSI-TCP/IP Konversion nach [Tanenbaum & Wetherall, 2012], [Fall & Stevens, 2012]

Im weiteren Verlauf der Arbeit wird zur vollständigeren Veranschaulichung das OSI-Referenzmodell verwendet und Protokolle der TCP/IP-Protokollfamilie den korrespondierenden Schichten darin zugeordnet. Die in dieser Arbeit hauptsächlich behandelten Protokolle sind Teil der Transportschicht und der Vermittlungsschicht (Internetschicht).

Dienstgüte kann prinzipiell auf allen Schichten realisiert werden [Kurose & Ross, 2014], wobei jede höhere Schicht auf den gegebenen Mechanismen der unteren Schichten aufsetzt. Dienstgüte wird aber hauptsächlich in den OSI-Schichten 2 und 3 bereitgestellt. Im Falle des Internets kann ein Benutzer diese Schichten nicht zu seinen Gunsten verändern, da sie für alle Netzwerkentitäten gleich funktionieren müssen. Wohl aber kann er die angebotenen Dienste einer Benutzer-Gleichbehandlung nutzen und mit Techniken in höheren Schichten ergänzen oder verbesserte Dienste beim Netzwerk Service Provider (NSP) einkaufen.

3.1.1 Bitübertragungsschicht und Sicherungsschicht (Netzzugangsschicht)

Durch die *Bitübertragungsschicht* wird eine mechanische, elektrische, funktionale und prozedurale Charakteristik zur Verfügung gestellt, um eine physikalische Verbindung zwischen Netzwerkknoten herzustellen, aufrechtzuerhalten und wieder freizugeben [Zimmermann, 1980].

Die physikalische Datenleitung betrifft die Art der Übertragungstechnologie sowie die damit einhergehende Kodierung der Bitabfolgen [Tanenbaum & Wetherall, 2012]. Bei der Bitübertragungsschicht, die das Übertragungsmedium für die Datenkommunika-

tion bereitstellt, kann die Wahl des Mediums bereits bestimmend für den Standard der angebotenen Dienstgüte sein. So besitzt ein kabelloses Medium in der Regel eine höhere Fehlerrate als eine Glasfaserleitung. Diese Merkmale werden durch die übertragungseigene Kodierung mithilfe der Sicherungsschicht in einen gegebenen Übertragungsstandard als grundlegende Eigenschaften des Übertragungsmediums, wie z.B. einer maximalen Datendurchsatzrate, umgewandelt [IEEE, 1985].

Eine Möglichkeit die Dienstgüte in der Bitübertragungsschicht über die gegebenen Übertragungseigenschaften zu heben, ist die Nutzung mehrerer physikalischer Datenleitungen. Dies kann entweder zu einer verbesserten Ausfallsicherheit beitragen oder die Datenübertragungsrate durch gemeinsame Nutzung erhöhen [IEEE, 2015] [Evensen et al., 2009]. Eine Aufteilung von unterschiedlichem Datenverkehr auf verschiedene Datenleitungen bzw. die Nutzung von Exklusivleitungen für Datenverkehr mit speziellen Dienstgüteanforderungen ist ebenfalls möglich (ebd.). Diese physikalische Schicht ist innerhalb des TCP/IP-Referenzmodells nicht definiert [Tanenbaum & Wetherall, 2012].

In der *Sicherungsschicht* geht es darum, die von der Bitübertragungsschicht angebotene Übertragung soweit aufzubereiten, dass Daten gesendet werden können und ein unverfälschter Datentransport über einen einzelnen Übermittlungsabschnitt möglich ist [Hübscher et al., 1999]. Dazu werden Bits in Rahmen eingeteilt und sowohl mit verschiedenen Techniken zur Fehlerkorrektur und Fehlervermeidung *(Logical Link Control, LLC)* als auch Mechanismen zur Datenfluss- und Zugriffskontrolle *(Medium Access Control, MAC)* gesendet [IEEE, 5/2008]. Mechanismen zum wiederholten Senden von Daten im Fehlerfall *(Automatic Repeat reQuest, ARQ)* können die BER ausgleichen, aber insgesamt die Datenübertragungsrate verringern.

Für die Dienstgüte in der Sicherungsschicht sind die gewählte Topologie und die zugrundeliegenden Zugriffsmechanismen der Übertragungstechnologie mit jeweils spezialisierten Möglichkeiten zur Fehlerkontrolle von zentraler Bedeutung. Hieraus ergibt sich die Adressierung, die in der Vermittlungsschicht durch das genutzte Vermittlungsprotokoll verwendet wird. Eine Zuordnung der Adressierung in der Sicherungsschicht wird im Falle von IPv4 mit dem Address Resolution Protocol (ARP) durchgeführt [Plummer, 11/1982]. Zugriffsmechanismen können bereits Implementierungen enthalten, die eine Priorisierung unterschiedlicher Datenströme bis zum nächsten Netzabschnitt ermöglichen.

3.1.2 Vermittlungsschicht (Internetschicht)

In der *Vermittlungsschicht* werden die Daten in Form von Datenpaketen an ihr Ziel geleitet. Die Netzwerkadressierung wird durch diese Schicht vorgenommen. Netzwerkknoten, die eine Weiterleitung vornehmen, untersuchen das Datenpaket, um das Ziel zu bestimmen, lesen andere Routing- und Dienstgüteinformationen aus und tragen das nächste Zwischenziel ein [Ibe, 2002]. Voraussetzung ist eine effiziente Routingstrategie, die auf der Erfassung und Zuweisung von Routen basiert. Kern des Routings ist die Adressierung, die im Falle des Internets auf dem Internet Protokoll (IP) basiert [Fall & Stevens, 2012]. Eine Erläuterung dieser Funktionsweise wird in Kapitel 3.2 gegeben.

Der Umgang mit Datenströmen, Strategien zur Überlastkontrolle, Flusskontrolle und Datenverkehrsformung ist ebenfalls Teil der Vermittlungsschicht. Diese Techniken bieten sich an, um eine erweiterte Dienstgüte in Form von Priorisierung bestimmter Datenströme auf NSP-Ebene zu implementieren. Weitere Erläuterungen zu diesen Konzepten folgen in Kapitel 3.3.

Neben dem IP-Protokoll sind weitere wichtige Protokolle das *Internet Control Message Protocol (ICMP)* für Funktionen zur Diagnose und Erkennung von Fehlern sowie das *Internet Group Management Protocol (IGMP)* für das Management von IP Multicast Gruppen [Postel, 9/1981]. ICMP wird durch die TCP/IP-Protokollfamilie genutzt, um Kontrollnachrichten zwischen dem Host und dem Router zu übertragen [Seth & Venkatesulu, 2008]. Diese beinhalten keinerlei Benutzerdaten, transportieren aber z.B. Fehlermeldungen, die die Unerreichbarkeit eines Endpunkts anzeigen (ebd.). Sie sind Teil der im TCP/IP-Referenzmodell definierten Zwischenschicht.

3.1.3 Transportschicht

In der *Transportschicht* erfolgt der Transport der Daten zwischen Endpunkten. Die Daten werden in *Segmente* oder *Datagramme* aufgeteilt und mithilfe von Transportkontrollmechanismen zur Datenflusskontrolle sowie Stauvermeidung an das empfangende Endsystem gesendet [Ibe, 2002]. Ein *Datenfluss* entsteht als ein Strom von Segmenten von einer Quelle zum Ziel unter der Nutzung einer logischen *Datenverbindung* zwischen den beiden Entitäten [Tanenbaum & Wetherall, 2012]. Unter Verwendung der Adressierung in der Vermittlungsschicht wird durch einen sogenannten *Port* eine zusätzliche Adressierung für den Datenzugang der anwendungsorientierten Schichten definiert [Fall & Stevens, 2012]. Eine Fehlererkennung bzw. Fehlerkorrektur der Segmente wird mithilfe von Kodierung realisiert (ebd.).

Dienstgüte lässt sich unter anderem mit den Mechanismen für verbindungsorientierte sowie verbindungslose Protokolle herstellen: Bei der verbindungsorientierten Kommunikation, die innerhalb der Internetprotokollfamilie mit dem *Transport Control Protocol (TCP)* verwirklicht wird, unterliegen die Mechanismen einem Zustandsmodell, welches die Phasen Verbindungsaufbau, Datenübertragung sowie Verbindungsabbau enthält.[14] Datenpakete der Vermittlungsschicht werden in Form von Segmenten abgesichert und bei Fehlern oder Datenverlust mithilfe von ARQ-Mechanismen erneut übertragen. Außerdem wird sichergestellt, dass die Datensegmente in der richtigen Reihenfolge eintreffen.

Eine verbindungslose Kommunikation sendet Daten, ohne dass ein vorheriger Verbindungsaufbau stattgefunden hat. ARQ-Mechanismen werden hierbei nicht angeboten. Ein verlustbehaftetes oder abhanden gekommenes Datagramm ist damit verloren, wenn es keine zusätzlichen Absicherungen in den höheren Modellschichten gibt. Dies eignet sich vor allem für Datenströme, die eine gewisse Fehlertoleranz besitzen. Ein Vorteil ist dabei für bestimmte Applikationen die direkte Zustellung der empfangenen Datensegmente ohne Empfangsbestätigungen. Dies hilft Latenzzeiten zu minimieren. Im Internet

[14] Vgl. Kapitel 3.3.6.3

wird für die verbindungslose Kommunikation das *User Datagram Protocol (UDP)* verwendet. UDP besitzt keine Kontrollmechanismen gegen Überlast [Tanenbaum & Wetherall, 2012]. Es kann also den Datenstrom hinsichtlich der Situation im Netzwerk nicht regulieren und andere Mechanismen müssen zum Tragen kommen.

3.1.4 Sitzungsschicht, Darstellungsschicht und Anwendungsschicht (Anwendungsschicht)

In der *Sitzungsschicht* wird die Prozesskommunikation zwischen den Endgeräten verbindungslos überwacht und gesteuert [Bossert & Breitbach, 1999]. Sie ist für den Aufbau, die Aufrechterhaltung und den Abbau einer Sitzung zwischen den laufenden Benutzerapplikationen zuständig [Ibe, 2002]. Daten werden synchronisiert und können damit unabhängig von Verbindungsabbrüchen der unteren Schichten übertragen werden (ebd.).

Die *Darstellungsschicht* ist für die Übersetzung der Informationen zuständig, die zwischen den beiden Systemen ausgetauscht und innerhalb der Anwendung verwendet werden [Tanenbaum & Wetherall, 2012]. In dieser Schicht werden unter anderem Datenkonvertierung, Verschlüsselung sowie Datenkompression durchgeführt [Ibe, 2002].

Auf der Ebene der *Anwendungsschicht* werden schließlich Dienste für Endbenutzeranwendungen bereitgestellt, um den Zugang zu Netzwerkressourcen zu regeln (ebd.). Dazu gehören alle Aktionen, die ein Benutzer auf Oberflächen durchführen kann und die Netzwerkaktivitäten hervorrufen.

Die genannten drei Schichten werden auch als anwendungsorientierte Schichten bezeichnet, da sie anwendungsspezifische Protokolle verwenden, die keine netzwerkspezifischen Mechanismen besitzen. Es liegt an diesen anwendungsorientierten Schichten, inwiefern die angebotenen Protokolle der unteren vier Schichten genutzt werden. So ist es z.B. denkbar, dass Applikationen erst eine gewisse Datenmenge speichern müssen, bevor diese dem Benutzer angezeigt werden kann, z.B. Internetbrowser [W3C, 2014]. Auch die Logik eines Peer-to-Peer-Systems mit einer Overlay-Topologie [Liu & Lu, 2007], bei dem Endpunkte auf eine gemeinsame Datenhaltung zugreifen, wird innerhalb dieser Schichten abgebildet.

Das *Real Time Protocol (RTP)* ist eines der anwendungsorientierten Internetprotokolle, das für die Übertragung von Echtzeitdaten verwendet wird [Schulzrinne et al., 7/2003]. Dabei wird in der Regel auf dem nicht zuverlässigen UDP-Protokoll in der Transportschicht aufgesetzt und die Daten werden für die Anwendung optimiert aufbereitet, z.B. durch die Nutzung von Zeitstempeln und Warteschlangenpuffern gegen ungeordneten Empfang oder Jitter.

RTP ist ein Protokoll für die optimierte Übertragung von medialen Echtzeit-Datenströmen, die keinen zwingenden Bedarf an fehlerloser Übertragung besitzen. RTP und das *RTP Control Protocol (RTCP)* [Schulzrinne et al., 7/2003] errechnen Statistiken für Dienstgüte-Informationen durch Sondierung: In regelmäßigen Abständen wird ein Bericht mit den die Anwendung betreffenden Parametern zwischen Empfänger und

Sender erstellt und dem jeweils anderen Partner zugesandt. Damit ist es möglich, auf verschiedene Qualitätslevels einzugehen und den Datenstrom somit anwendungsgesteuert und je nach Netzwerksituation in Hinblick auf die tatsächliche Nutzung der Daten zu regulieren.

3.2 Grundsätze des Routings

Routing bildet im Internet die Grundlage für das Übertragen von Daten zwischen zwei Endpunkten. Um die grundlegenden Mechanismen verstehen und verbessern zu können, bedarf es einer Einführung in Routingmechanismen. Diese sind in der Vermittlungsschicht des OSI-Referenzmodells definiert.

Routing ist in einem Netzwerk die Verkehrswegesuche, die Wegevermittlung und die Weitervermittlung von Daten [Brockhaus Enzyklopädie Online, 2017]. Das Routing entscheidet im Internet, welchen Weg Datenpakete innerhalb eines Teilnetzes *(Autonomes System, AS)* sowie an den Schnittstellen zwischen den Teilnetzen nehmen (ebd.).

Ein Router ist eine Entität, die mehrere Netzwerkabschnitte miteinander verbindet und mithilfe von *Routingtabellen* und der Zieladresse eines weiterzuleitenden Datenpakets den auf bestimmten Kriterien basierenden besten Pfad zur nächsten Routing-Entität bestimmt [Brockhaus Enzyklopädie Online, 2017].

Ein AS ist eine Ansammlung von Routern und Verbindungen innerhalb einer administrativen Domäne und ein Netzwerk eine Zusammenschaltung verschiedener AS [Willinger & Doyle, 2002]. AS im Internet sind über eine dreischichtige Hierarchie (*Tiers*) organisiert [Pohlmann & Dierichs, 2008]: AS der obersten Ebene (Tier-1) bieten weltumspannende Backbones, die wiederum andere AS miteinander verbinden und durch Transitverträge Datenaufkommen der unteren Tiers erlauben (Tier-2) (ebd.). Die unterste Ebene (Tier-3) bilden den Zugang für Endkunden (ebd.).

Es können zwei Hauptaktivitäten eines Routers unterschieden werden [Tanenbaum & Wetherall, 2012]:

1. Entscheidung der Weiterleitung eines Datenpakets (Zielermittlung)
2. Pflege der Routingtabellen (Befüllung und Aktualisierung)

Das Auffinden des nächsten Ziels eines Datenpakets wird mittels einer Routingtabelle durchgeführt, in der sich entsprechende Zieladressen und respektive Dienstgütekosten befinden. Kernstück des Routingvorgangs ist die Adressierung, die im Internet mit dem *Internet Protokoll* (IP) realisiert wird [Information Sciences Institute, University of Southern California, 9/1981]. Dabei geht es um eine eindeutige Adressierung der Entität und des Teilnetzwerks. IP findet in zwei Versionen Verwendung: IPv4 und IPv6 [Deering & Hinden, 12/1998]. Diese unterscheiden sich hinsichtlich ihrer Eigenschaften auf die mögliche Netzwerkgröße, aber auch in anderen Optionen innerhalb des Headers.

In den weiterführenden Experimenten in dieser Arbeit wurde hauptsächlich IPv4 verwendet. Eine Erläuterung der Dienstgüteoptionen von IPv4 wird in Kapitel 3.3.3 gegeben. Eine IPv4-Adresse beruht auf einem Tupel, welches aus vier Bytes für den Rechner und vier weiteren Bytes für die Markierung des Netzwerks innerhalb der Adresse besteht.

Für die Befüllung und Aktualisierung der Routingtabelle werden verschiedene Protokolle verwendet. Diese folgen meist Konzepten des „kürzesten Pfades“ (shortest-path Algorithmen). Darunter gibt es Techniken wie der Bellman-Ford-Algorithmus [Ford, 1956] oder der Dijkstra-Algorithmus [Dijkstra, 1959] für die Berechnung des kostengünstigsten Pfades. Welche verwendeten Dienstgüte-Metriken für die Kostenermittlung verwendet werden wird dabei durch den NSP festgelegt.

Routingprozeduren können zwischen statischen und dynamischen bzw. adaptiven Prozeduren unterschieden werden [Leon-Garcia & Widjaja, 2004]. Beim statischen Routing werden die Kosten „offline“ aufgrund der Topologie ermittelt und dann an die Router weitergegeben (ebd.). Beim dynamischen Routing erfolgt eine Berechnung durch die Router selbst, bezogen auf die derzeitige Situation, die dann mithilfe verschiedener Protokolle unter den Routern ausgetauscht wird (ebd.). Die beiden wichtigsten dynamischen Routingalgorithmen sind [Tanenbaum & Wetherall, 2012]:

1. der Distanzvektor-Algorithmus
2. der Link-State-Algorithmus

Der *Distanzvektor-Algorithmus* funktioniert nach einem verteilten Bellman-Ford-Algorithmus. Das *Routing Information Protokoll (RIP)* [Malkin, 11/1998] bildet die technische Umsetzung für das Internet. Hierbei tauschen sich Netzwerkknoten mit ihren Nachbarn über die Kosten der Wege zu ihren bekannten Netzwerkknoten aus [Leon-Garcia & Widjaja, 2004].

Aufgrund der zu langsamen Ausbreitung von Störungsmeldungen (Count-to-Infinity-Problem [Schmid & Steigner, 2002]) wurde weitestgehend auf Link-State-Algorithmen wie *Intermediate System to Intermediate System Protocol (IS-IS)* [Oran, 2/1990] [Callon, 12/1990] und *Open Shortest Path First (OSPF)* [Williams & Melnikov, 5/2008] umgestellt [Tanenbaum & Wetherall, 2012]. Bei den *Link-State-Algorithmen* werden zunächst die Wegkosten zu den Nachbarknoten ermittelt, dann an jeden Netzwerkknoten innerhalb des Teilnetzwerks geschickt und schließlich in jedem Knoten die kürzesten Strecken zu allen im Netzwerk enthaltenen Knoten berechnet.

Für den Austausch von Pfadinformationen zwischen Teilnetzwerken nutzen Netzwerkknoten an den Grenzzugängen das *Border Gateway Protocol (BGP)* [Rekhter et al., 01/2006]. Hiermit wird der Austausch der Routingtabellen mit den benachbarten Netzwerken durchgeführt, die ein hierarchisches Routing über Netzwerkgrenzen hinweg ermöglichen. Außerdem gibt es Protokolle für das Routing eines Datenpakets an mehrere Zieladressen. Diese arbeiten mit Spannbäumen, um die kürzesten Wege zu Benutzergruppen ausfindig zu machen.

Die Verwendung von mehreren physikalischen Leitungen innerhalb der Bitübertragungsschicht ist eine naheliegende Möglichkeit, um die Dienstgüte zu verbessern.[15] Beim Multipath-Routingverfahren werden mehrere Datenpakete auf verschiedene Pfade verteilt, um

- eine gleichmäßige Auslastung des Netzwerks zu gewährleisten
- Datenströme robuster gegen Ausfälle zu machen
- eine erhöhte Datenübertragungsrate zu erlangen

Ein weitere Möglichkeit sind Source Routing-Techniken, mit deren Hilfe Datenpakete von einem Steuerungsrechner gezielt auf einem bestimmten Pfad übertragen werden können. Bei diesen Techniken fallen jedoch ein erhöhter Overhead sowie ein größerer Rechenaufwand an.

Überlastsituationen entstehen vor allem an Netzwerk-Engpässen. Diese sind abhängig von der Lokalisierung – dort, wo viel Netzwerkverkehr auf einen bestimmten Netzwerkknoten trifft, und abhängig von der Zeit – dann, wenn viele Daten verschickt werden müssen. Überlastsituationen lassen sich grundsätzlich minimieren, indem NSPs für bestimmte Netzwerkknoten höhere Kapazitäten installieren oder mithilfe von Traffic-Engineering-Techniken[16] andere Routingpfade ermöglichen, die die Kapazitäten des Netzwerks gleichmäßiger auslasten können. Hierdurch treten zwar weniger Überlastungen auf, dies kann aber mit einer erhöhten Latenz durch suboptimale Routen einhergehen.

3.3 Dienstgüte in paketvermittelnden Best-Effort Netzwerken

In diesem Kapitel werden die Grundlagen der Dienstgüte-Mechanismen beschrieben. Diese Mechanismen sind die Basis für die Durchsetzung der Anforderungen der Anwendungen innerhalb eines Netzwerks. Die Dienstgüte folgt der in Kapitel 2.4 beschriebenen Definition.

Die geschichtliche Entwicklung von Netzwerken beginnt zunächst mit der Entwicklung von Telegrafen- und Telefonnetzen [Bunz, 2009]. Diese leitungsvermittelten Techniken sind durch die physikalische Schaltung einer eindeutigen Verbindung zwischen einem Sender und einem Empfänger definiert [Tanenbaum & Wetherall, 2012]. Im späteren geschichtlichen Verlauf wurden Netzwerke vermehrt – aufgrund der Einführung von Datenübertragung – paketvermittelnd umgesetzt (ebd.). Bei der paketvermittelnden Technik werden Daten bzw. Informationen in kleine Pakete zerstückelt und einzeln in ein Netzwerk eingespeist (ebd.).

Die Pakete können dabei unterschiedliche Pfade eines Netzwerks nutzen – abhängig von den Regeln und Eigenschaften der verwendbaren Netzwerkressourcen. Auf Empfangsseite werden die Pakete wieder zur ursprünglichen Nachricht zusammengesetzt. Die Nutzung von paketvermittelnden Netzwerken hat mehrere Vorteile. Zum einen kann sie effizienter, einfacher und kostengünstiger implementiert werden, zum anderen bietet

[15] Vgl. hierzu die Ausführungen in Kapitel 3.1.1
[16] Vgl. hierzu die Ausführungen in Kapitel 3.3.5

sie eine bessere Aufteilung der Datenübertragungsrate, als es die Leitungsvermittlung ermöglicht [Kurose & Ross, 2014]. Der größte Nachteil der Paketvermittlung liegt in den unterschiedlichen und nicht vorhersagbaren Laufzeiten, die einzelne Pakete nehmen können (ebd.).

Knotenpunkte können bei zu hohem Datenaufkommen überlastet werden und es können keine konstanten Datenübertragungsraten garantiert werden (ebd.). Ein Spezialfall ist die Zellvermittlung. Hierbei werden Daten in kleine Pakete bzw. Zellen zerlegt und diese im Zeitschlitzverfahren auf festen virtuellen Pfaden durch das Netzwerk gesendet [Ibe, 2002].[17] Damit können Vorhersagen bezüglich der Dienstgüte getroffen und garantiert werden.

Die Übertragung von Daten, die einer harten Echtzeit unterliegen, wird meist über leitungsvermittelte oder zellvermittelnde Netzwerke realisiert, da diese grundsätzlich als berechenbar gelten. Das Kombinieren dieser Vermittlungsart innerhalb eines paketvermittelnden Datennetzes wird *Netzwerkkonvergenz* genannt (ebd.). Mit verschiedenen Mechanismen können die Eigenschaften von paketvermittelnden Netzwerken verbessert werden, um entsprechende Applikationen und Echtzeit-Anwendungen zu unterstützen [Leon-Garcia & Widjaja, 2004].

Das Internet ist in seiner ursprünglichen Technologie ein paketvermittelndes Netzwerk. Es kann aus zwei Perspektiven – einer externen und einer internen – betrachtet werden, die zur Beschreibung der benötigten Mechanismen genutzt werden können [Leon-Garcia & Widjaja, 2004]:

Bei der *externen Sichtweise* eines Netzwerks werden vom NSP auf Endbenutzerebene mögliche Dienste oder Dienstklassen für das Endbenutzersystem bereitgestellt (ebd.). Diese Dienste setzen auf den Netzwerkschichten auf und zwar ab der Transportschicht (ISO/OSI-Referenzmodell Schicht 4) [Leon-Garcia & Widjaja, 2004]. Das *Ende-zu-Ende-Prinzip* wird innerhalb dieser Schichten realisiert [Saltzer et al., 1984]. Hier geht es darum, dass die Anwendungen der Endbenutzersysteme eine Verbindung miteinander schließen und die vom Netzwerk bereitgestellte Dienstgüte ausnutzen können. Es soll möglich sein, beliebige Funktionen zu implementieren, die beide Endsysteme verstehen können. Darunterliegende Schichten sollen hierbei nicht beeinträchtigt werden oder umgekehrt die eingesetzten Funktionen der Endgeräte beeinflussen (ebd.).

Idealerweise ist die Definition von Netzwerkdiensten unabhängig von dem darunterliegenden Netzwerk und seinen Schichten zu betrachten, welches die *interne Sichtweise* beschreibt [Leon-Garcia & Widjaja, 2004]. Diese zeichnet sich durch die physikalische Topologie sowie die Vernetzung der Endpunkte und Netzwerkgeräte aus (ebd.). Sie beinhaltet die Adressierung und das Routing für die Versendung von Datenpaketen und verwaltet den Datenverkehr sowie Mechanismen zum Umgang mit Datenverkehrsüberlastung (ebd.). Die interne Sichtweise bildet die Basis für die vom Endbenutzer verwendeten Dienste.

[17] wie z.B. bei Asynchronous Transfer Mode (ATM)

Das Internet wurde ursprünglich für die Versendung von zeitunkritischen Daten erdacht. Es ist eine Verbindung aus mehreren Netzwerken mit teilweise unterschiedlichen Technologien in den unteren Netzwerkschichten. Es basiert auf dem Internet Protokoll (IP), welches in der dritten Schicht des OSI-Referenzmodells implementiert ist [Fall & Stevens, 2012]. Viele Internet Protokoll-Netzwerke wurden traditionell in Hinblick auf ihre Datenübertragungsrate und ihre Ressourcennutzung entwickelt [Briscoe et al., 2014]. Allgemein gilt für im Internet versendete Daten das sogenannte „Best-Effort-Prinzip“ [Kurose & Ross, 2014].

Beim **Best-Effort-Prinzip** gibt es keine Garantien hinsichtlich des Timings, der Datenübertragungsrate, der eintreffenden Reihenfolge oder der Zustellung der Datenpakete (ebd.). Datenpakete werden nach bestem Bemühen und möglichst fair ausgeliefert. Probleme im Netzwerks betreffen alle Netzwerkteilnehmer. Um die Übertragungseigenschaften der Datenversendung sowohl für die allgemeine Funktion des Best-Effort-Prinzips zu optimieren als auch für individuelle Dienste zu verbessern, werden Techniken der Dienstgüte angewandt.

Dienstgütegarantien können durch die im Netzwerk installierten Zwischengeräte sowie Dienste eines Netzwerks erzielt werden [Aidarous & Plevyak, 2003]. Grundlegende Mechanismen können zur Verbesserung der Eigenschaften eines Netzwerkes beitragen. Eine Möglichkeit ist die *Überdimensionierung* eines Netzwerks (ebd.), wie es bei einem leitungsvermittelten Netzwerk automatisch der Fall ist [Tanenbaum & Wetherall, 2012]. Benötigte Dienstgüten stehen hierbei immer für die genutzten Dienste zur Verfügung. Diese Möglichkeit wird vor allem aufgrund hoher Kosten nur bei kleineren *Local Area Netzwerken (LAN)* genutzt [Aidarous & Plevyak, 2003].

Im Falle des Internets werden zunächst Best-Effort-Verfahren eingesetzt, um eine möglichst effiziente und gerechte Weiterleitung von Daten aller Netzwerkentitäten an Knotenpunkten zu erreichen sowie Datenverkehrsstaus zu kontrollieren und zu vermeiden. Zusätzlich bietet es sich an, diese Mechanismen soweit zu steuern, dass Datenströme priorisiert oder Ressourcen explizit reserviert werden können. Die Verfahren werden im Internet vornehmlich in der dritten Schicht des OSI-Referenzmodells durchgeführt. Zusätzliche Erweiterungen für verbesserte Eigenschaften des Ende-zu-Ende-Betriebs können auch in höheren Schichten erreicht werden.

3.3.1 Warteschlangenverwaltung

Eine *Warteschlange* ist eine Datenstruktur, die hereinkommende Datenpakete vor einer Weiterverarbeitung speichert [Brockhaus Enzyklopädie Online, 2017]. Warteschlangen werden in Best-Effort-Netzwerken für mehrere Mechanismen genutzt, um Dienstgüte zu begrenzen, zu optimieren oder zuzusichern. Sie werden im Forschungsbereich der *Warteraumtheorie* definiert.

Eine Warteschlange bzw. ein Warteraum besteht aus einem *Wartebereich* und einer *Bedienstation* [Bossert & Breitbach, 1999]. Der Wartebereich besitzt eine durch einen externen Prozess ausgelöste Eingaberate von Datenpaketen, die den Wartebereich füllt (ebd.). Die Bedienstation nimmt die durch den Wartebereich gespeicherten Datenpakete

mit einer bestimmten Bedienrate und Reihenfolge entgegen (ebd.). Über die Reihenfolge der Bedienung entscheiden Algorithmen, die Einfluss auf das Verhalten der Ausgabe haben.

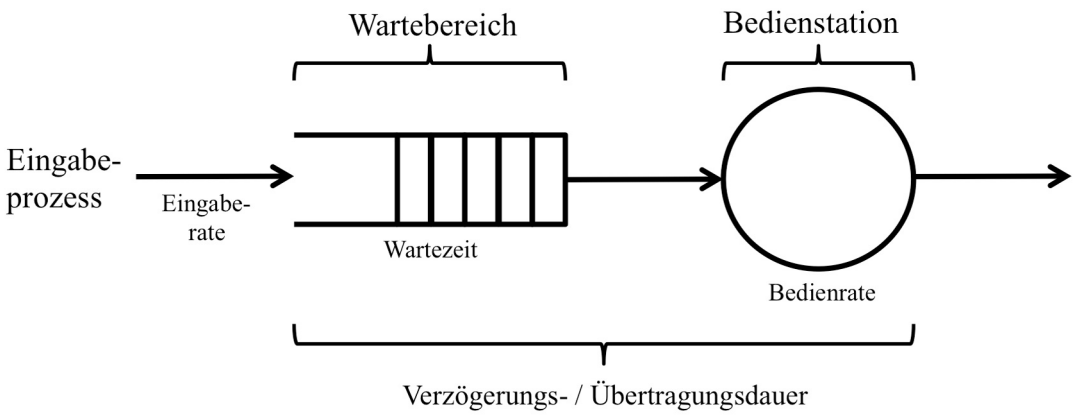

Abbildung 3.3: Modell einer Warteschlange nach [Bossert & Breitbach, 1999]

Im Internet werden folgende Mechanismen mithilfe von Warteschlangen eingesetzt:

- *Datenverkehrsformung (Shaping)*
- *Datenpaket-Scheduling (Queueing)*
- *Zwischenspeicherung (Buffering)*

Diese Mechanismen bilden die Voraussetzungen für die Bereitstellung eines Best-Effort-Netzwerks. Im Folgenden werden die Mechanismen Datenverkehrsformung und Datenpaket-Scheduling vorgestellt. Die Zwischenspeicherung erfolgt in Kapitel 3.3.6.1.

3.3.1.1 Datenverkehrsformung und -regulierung (Shaping & Policing)

Ein Nachteil eines paketvermittelnden Netzwerks ist die unstete Inanspruchnahme der möglichen Kapazität, die, je nach Datenform und Anzahl aktiver Netzwerkentitäten, in Form von *Bursts* (Stöße, Häufungen) geschehen kann [Tanenbaum & Wetherall, 2012]. Die Zahl der abgesandten Pakete ändert sich innerhalb von Bruchteilen einer Millisekunde [Bossert & Breitbach, 1999]. Dies erhöht die Gefahr einer Datenverkehrsstaubildung [Aggarwal, 2012].

Die **Datenverkehrsformung** wird verwendet, um eine durchschnittliche Übertragungsrate und das Burst-Aufkommen eines Datenflusses bei Netzwerkeintritt zu regulieren [Tanenbaum & Wetherall, 2012]. Dieser Mechanismus erlaubt es einem NSP ebenfalls, mehr Kapazitäten zu verkaufen als er eigentlich besitzt (*Überzeichnung, Oversubscription)* [Leon-Garcia & Widjaja, 2004].

Eine Überzeichnung ist möglich, wenn *Dienstgütevereinbarungen (Service-Level-Agreement, SLA)* zwischen dem NSP und dem Kunden, der das Netz in Anspruch nehmen möchte, geschlossen werden [Tanenbaum & Wetherall, 2012]. Hierbei einigen sich beide Parteien auf eine bestimmte vom Kunden erwünschte Dienstgüte (ebd.). Die einzuhaltenden Parameter werden durch den NSP durchgesetzt oder überprüft (ebd.).

Zweck dieser Technik ist es, den Datenfluss mithilfe einer Warteschlange vor Netzwerkeintritt auf ein bestimmtes Muster zu begrenzen und so dem NSP die Möglichkeit zu bieten, das Netz und seine Ressourcen nach seinen Vorstellungen zu regulieren und zu verwalten. Dieses Verfahren, die Überprüfung der eintreffenden Datenpakete auf die einzuhaltende Dienstgüte, wird *Policing* genannt [Tanenbaum & Wetherall, 2012].

Um Verzögerungen durch die hervorgerufene Speicherung bei der Datenverkehrsformung zu minimieren, muss der Netzwerkbetreiber ein Gleichgewicht zwischen der Größe der Warteschlange und der angeforderten Dienstgüte finden [Bossert & Breitbach, 1999]. Dies ergibt sich aus der Kapazität und der Auslastung des zu betreuenden Netzwerks. Für eine Datenverkehrsformung werden grundlegend zwei Arten von Algorithmen verwendet:

- *Leaky-Bucket*
- *Token-Bucket*

Beim *Leaky-Bucket*-Algorithmus wird der Wartebereich der Warteschlange durch ungleichmäßig hereinkommende Datenpakete gefüllt [Leon-Garcia & Widjaja, 2004]. Die Bedienstation der Warteschlange besitzt eine Weiterleitung mit konstanter Ausgabe (ebd.). Der Wartebereich hat eine endliche Größe, die bei Überfüllung weiter hereinkommende Datenpakete auslässt (ebd.) oder in eine weitere vorgeschaltete Warteschlange stellt [Tanenbaum & Wetherall, 2012]. Stoßweise ankommender Datenverkehr wird in eine konstante und begrenzte Ausgaberate verändert und ein durchschnittlicher Datenfluss wird erzeugt [Aggarwal, 2012]. Der Leaky-Bucket-Algorithmus erlaubt keine direkte Weiterleitung von stoßweise ankommendem Datenverkehr (ebd.).

Eine andere Herangehensweise erlaubt der *Token-Bucket*-Algorithmus als eine Erweiterung des Leaky-Bucket-Algorithmus [Leon-Garcia & Widjaja, 2004]. Hier werden nur Daten weitergeleitet, die in eine festgelegte Anzahl und Größe von Bursts passen [Aidarous & Plevyak, 2003]. Diese sogenannten Tokens werden bis zu einer bestimmten Menge in einer festgelegten Rate und Größe erzeugt und legen die Menge an Daten fest, die weitergeleitet werden können (ebd.). Zu einem Zeitpunkt, an dem viele Tokens vorhanden sind, können hohe Datenübertragungsraten gewährleistet werden. Sind keine Tokens mehr vorhanden, müssen die Daten in der Warteschlange verweilen, bis neue Tokens erzeugt wurden (ebd.). Ein Vorteil ist, dass Hosts in der Lage sind, unregelmäßige Datensendungen in das Netzwerk zu senden, ohne dass die Datenübertragungsrate davon beeinflusst wird [Aggarwal, 2012]. Sie wird nur dann begrenzt, wenn die Datensendungen zu häufig erfolgen und dann z.B. über dem SLA liegen.

3.3.1.2 Datenpaket- und Warteschlangen-Scheduling (Queueing)

Das **Datenpaket-Scheduling** wird für die Regulierung der Daten im Netz verwendet und innerhalb von Netzwerkentitäten implementiert, die Datenverkehr empfangen und weiterleiten. In den meisten Fällen ist es das Hauptziel des Datenpaket-Scheduling die Systemkapazität zu maximieren, effizient den Dienstgüteansprüchen der Benutzer zu genügen und ein Mindestlevel an Fairness bereitzustellen [Tsai et al., 2010].

Dabei können mehrere Warteschlangen zum Einsatz kommen. Die Warteschlangen sind abhängig von den hereinkommenden Datenströmen bzw. den logischen Datenleitungen. Scheduling-Algorithmen folgen den nachstehenden Eigenschaften (ebd.):

- Effizienz
- Absicherung
- Flexibilität
- geringe Komplexität

Die Algorithmen müssen die verschiedenen Datenströme gegeneinander so absichern, dass sie sich nicht gegenseitig stören können und die Auslastung nicht einseitig gegeben ist [Tsai et al., 2010]. Verschiedene Dienstgüte-Ansprüche sollten möglichst flexibel und mit großer Vielfalt unterstützt werden (ebd.). Des Weiteren muss ein Scheduling-Algorithmus eine geringe Komplexität besitzen, um den Rechenaufwand in der Netzwerkentität möglichst klein zu halten (ebd.).

Scheduling-Mechanismen werden eingesetzt, um hereinkommende Datenströme auf die vorhandenen Ressourcen gerecht oder nach bestimmten Voraussetzungen zu verteilen. Zu den möglichen gemeinsam verwendeten Ressourcen einer Netzwerkentität zählen [Tanenbaum & Wetherall, 2012]:

- Datenübertragungsrate
- Pufferspeicher
- CPU-Zyklen

Die *Datenübertragungsrate* ist der maximal mögliche Durchsatz von Daten. Der *Pufferspeicher* definiert die Größe der Warteschlange, also wie viele Daten zwischengespeichert werden können, bevor diese weitergeleitet werden. *CPU-Zyklen* definieren den benötigten Rechenaufwand für die Bearbeitungszeit.

Es gibt mehrere grundlegende Algorithmen für das Datenpaket-Scheduling. Die wichtigsten sind die folgenden:

1. *First-in first-out (FIFO)* oder auch *First Come First Serve* ist einer der einfachsten und am meisten verwendeten Warteschlangen-Algorithmen aufgrund der geringen Rechenkosten [Aidarous & Plevyak, 2003]. Alle Datenpakete besitzen dieselbe Priorität, unabhängig von ihrer Größe, des Typs, Inhalts oder anderen Paketcharakteristiken (ebd.). Sie werden in der gleichen Reihenfolge bearbeitet, wie sie eintreffen [Ibe, 2002]. Durch die Gleichbehandlung aller Datenpakete folgt der Algorithmus einem Best-Effort-Verhalten. Probleme sind mögliche Beeinträchtigungen zwischen Datenflüssen mit unterschiedlichen Eigenschaften [Tanenbaum & Wetherall, 2012].
2. *Fair-Queueing (FQ)* ist eine Verbesserung des FIFO-Algorithmus, um Ressourcen gerechter zu verteilen: FQ arbeitet mit je einer Warteschlange pro logischer Datenleitung, die nach dem *Round-Robin-Prinzip (RR)* abgearbeitet werden

[Tanenbaum & Wetherall, 2012]: Eingehende Datenströme werden abwechselnd bearbeitet. Das Fair-Queueing besitzt ebenfalls Best-Effort-Verhalten.

3. Das *Priority Queueing (PQ)* verbessert die Dienstgüte des FIFO-Algorithmus für individuelle Datenpakete, indem es anstatt nur einer Warteschlange, mehrere mit unterschiedlichen Prioritäten bietet: Eintreffende Datenpakete oder Datenflüsse werden klassifiziert [18] und den Prioritätswarteschlangen zugeteilt [Aidarous & Plevyak, 2003]. Ein Abarbeiten der Warteschlangen erfolgt mithilfe RR, wobei die höchste Priorität stets zuerst abgearbeitet wird [Tanenbaum & Wetherall, 2012]. Ein Nachteil besteht darin, dass Datenpakete mit kleinerer Priorität nicht abgearbeitet werden können, solange es Datenpakete in der höheren Priorität gibt.
4. Eine Verbesserung der Dienstgüte kann mit dem *Weighted-Fair-Queueing (WFQ)-Algorithmus* erreicht werden. Hierbei werden die einzelnen Datenflüsse klassifiziert und mehreren, den Klassifizierungen entsprechenden, Warteschlangen zugeteilt [Ibe, 2002]. Diese werden dann mithilfe von unterschiedlichen Gewichten durch den RR-Scheduler abgearbeitet (ebd.). Gegeben durch das anteilige Gewicht wird jede Warteschlange in endlicher Zeit bearbeitet.

Während es bei Warteschlangen-Schedulern darum geht, unterschiedlichen Datenverkehr möglichst effizient zu integrieren und auf die Ressourcen zu verteilen, können Überlastsituationen auftreten, an die sich der Scheduler anpassen muss. Bei knapper Ressourcensituation am Ausgang und gleichzeitigem Ressourcenbegehren durch ankommende Datenströme kann der Datenverkehr nicht wie gewünscht weitergeleitet werden.

Eine Überlastsituation kann zu erhöhtem Paketverlust und damit zu Verzögerungen und Jitter führen [Aidarous & Plevyak, 2003]. Liegt mehr Datenverkehr an den Warteschlangen der eingehenden Datenströme vor als weitergeleitet werden kann, so erhöht sich die Menge der Daten in der Warteschlange, die auf eine Weiterleitung warten. In diesem Fall kann die Netzwerkentität erkennen, dass eine Stausituation vorliegt. Ankommende Datenpakete müssen in dieser Situation mithilfe von Verwerfungsmethoden bzw. durch Lastabwurf gelöscht werden, um den Verkehr an die Netzwerksituation an der Ausgabe anzupassen. Die Techniken, die als Verwerfungsmethoden innerhalb des Datenverkehrs-Scheduling eingesetzt werden, zählen zu den Hop-zu-Hop-Techniken für Überlast-Management und -Vermeidung [Leon-Garcia & Widjaja, 2004].

Die folgenden Algorithmen sind die gängigsten, um bei Überlastsituationen Datenpakete zu verwerfen:

1. Beim *Tail Drop-Algorithmus* werden alle Pakete verworfen, die nicht mehr in die Warteschlange passen [Tanenbaum & Wetherall, 2012]. Bei diesem Algorithmus können unfaire Ergebnisse entstehen, wenn verschiedene un-

[18] siehe Kapitel 3.3.2

gleich schnell gesendete Datenströme auf die Warteschlange treffen [Aidarous & Plevyak, 2003]. Hinzu kommen Synchronisationsprobleme durch das Sendeverhalten der Überlastkontrollmechanismen in der Transportschicht [Aidarous & Plevyak, 2003].

2. Der *Random Early Detection-Algorithmus (RED)* verwirft Pakete, bevor eine Überlastsituation eintritt [Aidarous & Plevyak, 2003]. Datenpakete, die sich in der Warteschlange befinden, werden nach zufälligem Muster verworfen – noch bevor der Pufferspeicher der Netzwerkentität ausgeht (ebd.). Diese Methode wird umso stärker angewendet, desto größer die Überlast ist (ebd.). Durch die zufällig verworfenen Datenpakete ist die Wahrscheinlichkeit für Probleme mit Überlastkontrollmechanismen in der Transportschicht geringer und die Fairness bei unterschiedlich großen Datenströmen ist höher (ebd.).

Eine vollgelaufene Warteschlange kann erst dann Datenpakete wieder annehmen, wenn ein Teil der in der Warteschlange enthaltenen Datenpakete abgearbeitet ist [Bolot, 1993]. In dieser Zeit werden weiterhin eintreffende Datenpakete nach den Algorithmen gelöscht. Bei einer höheren Sendefrequenz von konsekutiven Datenpaketen ist somit der Datenpaketverlust ebenfalls höher. Datenpaketverluste treten daher oftmals in Stößen auf (ebd.).

Andere Probleme an Netzwerkknoten, die bei Überlastungen auftreten, sind die Auswirkungen verschiedener Datenverkehrstypen aufeinander. Hierbei können große Datenpakete, die z.B. von Massendaten herstammen, und kleine Datenpakete, die durch interaktive Datenströme hervorgerufen werden, unterschiedlich weitergeleitet werden. Wenn ein Knotenpunkt keine Priorisierung beherrscht, werden die Datenpakete gleichberechtigt behandelt. Das Resultat ist eine vermehrte Blockierung der Warteschlange durch große Datenpakete. [Järvinen et al., 2013] beschreiben ein Phänomen, bei dem eine gegenseitige Störung von Datenströmen durch die parallele Nutzung eines Flaschenhalses auftritt. Typischer Web-Datenverkehr, der beim Aufrufen von Webseiten entsteht, wird dabei vermehrt als aggressiv gegenüber interaktiven Video- und Audioverbindungen eingestuft. Das Resultat sind eine höhere Latenz oder vermehrter Datenpaketverlust an Engpässen.

3.3.2 Zugangskontrolle und Paketklassifizierung

Zusammen mit den oben erläuterten Mechanismen wurden bereits priorisierte Verfahren erläutert, die Datenflüsse mit bestimmten Eigenschaften anders behandeln als andere. Dies entspricht bereits nicht mehr dem Best-Effort-Ansatz. Ein NSP besitzt verschiedene Gründe, um über den Best-Effort-Ansatz hinauszugehen:

1. Ein NSP ist daran interessiert, dass sein Netzwerk möglichst gleichmäßig ausgelastet wird und nicht nur bestimmte Bereiche, die über ein autonomes Routingverfahren als kürzesten Weg bestimmt wurden.

2. Ein NSP will so viel Leistung wie möglich verkaufen können, auch wenn es die Leistungsfähigkeit des Netzwerks überschreitet (*Oversubscription*).
3. Ein NSP möchte für bestimmte Teilnehmer stabilere oder leistungsfähigere Kommunikationsmöglichkeiten bereitstellen als für andere Teilnehmer.

Diesen Absichten liegen Verfahren zugrunde, die die oben beschriebenen Mechanismen nutzen, um erweiterte Dienstgüten für bestimmte Datenflüsse anzubieten. Bei der *Zugangssteuerung* wird festgelegt, wie der Datenverkehr angenommen werden kann und welche Ressourcen für ihn reserviert werden [Leon-Garcia & Widjaja, 2004]. Existieren zu einem bestimmten Datenstrom Dienstgütezusagen, muss überprüft werden, ob diese innerhalb der Netzwerkentitäten eingehalten werden können. Es besteht dann die Möglichkeit, den Datenstrom anzunehmen oder abzulehnen (ebd.). Eine Zugangskontrolle muss für alle Netzwerkentitäten der gesamten Übertragungsstrecke angenommen werden, um zu funktionieren (ebd.).

Die Zugangskontrollentität berechnet die Ressourcenanforderungen eines Datenflusses und die freien Ressourcen innerhalb des Netzwerkpfades (ebd.). Die Ressourcenanforderungen beziehen sich auf bestimmte Parameter, die andere Netzwerkentitäten innerhalb des Netzwerks verstehen können. Diese Parameter können Dienstgüteanforderungen sein, wie z.B. die maximale Datenübertragungsrate, aber auch Datenflussspezifikationen, wie die minimale Paketgröße, die benötigte Token-Bucket-Rate oder die maximale Dauer eines Bursts [Tanenbaum & Wetherall, 2012].

Eine *Paketklassifizierung* ist notwendig, um zwischen verschiedenen Datenflüssen in einem Netzwerk unterscheiden zu können und sie dann für den weiteren Gebrauch zu markieren [Aidarous & Plevyak, 2003]. Die Markierung kann dann für eine Priorisierung genutzt werden. Eine Paketklassifizierung kann mithilfe unterschiedlicher Kriterien durchgeführt werden. Beim Eintreffen des Pakets wird der Paket-Header, je nach Dienstgüte-Strategie, auf bestimmte Flusseigenschaften hin untersucht [Aidarous & Plevyak, 2003]. Diese Eigenschaften können die folgenden Kriterien betreffen (ebd.):

- Quelladresse
- Zieladresse
- Genutztes Transportprotokoll (UDP/TCP)
- Port
- weitere Kriterien innerhalb des Protokoll-Headers (Paketgröße, Type of Service etc.)

Ein zur Dienstgüteverbesserung oft genutzter Dienst, der mithilfe dieser Technik arbeitet, betrifft die sogenannten *differenzierten Dienste* [Aidarous & Plevyak, 2003].[19] Die Paketklassifizierung kann durch die oben aufgelisteten Kriterien sowohl benutzerabhängig als auch anwendungsabhängig gestaltet werden (ebd.). Die Komplexität bei einer Klassenzuordnung kann stark variieren und hängt von der gewünschten Dienstgü-

[19] Vgl. Kapitel 3.3.3

tequalität ab. Netzwerkeinstiegspunkte oder Router müssen für die entsprechende Komplexität gerüstet sein.

Nach einer durchgeführten Klassifizierung werden die Pakete entsprechend markiert. Hierdurch ist es möglich, die folgenden Strategien zur Dienstgüte durchzuführen:

- SLA Traffic-Shaping Policy
- Priorisierung innerhalb des Schedulers
- Verwerfungspriorität innerhalb der Warteschlange
- Labeling[20]

3.3.3 Dienstgüte mit Differenzierten Diensten (DiffServ)

Differenzierte Dienste [Nichols et al., 12/1998] [Blake et al., 12/1998] sind eine Möglichkeit, die Dienstgüte für bestimmte Datenflüsse klassenbasiert im Internet zu verbessern [Tanenbaum & Wetherall, 2012]. Diese Möglichkeit kann innerhalb eines Netzwerks durch den NSP angeboten werden. Ein NSP kann damit eine Differenzierung in Hinblick auf den Dienst und den Benutzer separat vornehmen [Ibe, 2002].

Die grundlegende Funktionsweise von **differenzierten Diensten** ist es, dass in das Netzwerk eintretende Datenpakete klassifiziert und für die entsprechende Dienstklasse markiert werden. Eine zusätzliche Signalisierung für den gesendeten Datenstrom ist nicht notwendig (ebd.).

Differenzierte Dienste bieten keine Dienstgütegarantien, da die Dienstgüteklassen an jeder Netzwerkentität gesondert behandelt werden (ebd.). Markierte Datenpakete werden per Teilstrecke, je nach Dienstgüteklasse, priorisiert weitergeleitet [Tanenbaum & Wetherall, 2012]. Die Festlegung der Dienstgüteklassen obliegt dem NSP. Eine mögliche Priorisierung von Datenpaketen, bietet die Klassifizierung nach dem ITU-T Y.1541-Standard in Tabelle 3.1 [ITU, 12/2011]. Diese Dienstgüte-Klassifizierung unterscheidet zwischen Paketverlustrate, Ende-zu-Ende-Verzögerung und Jitter.

QoS-Klasse	mögl. Anwendungen	Packet-verlustrate	Laufzeit	Jitter
0	Interaktiver Echtzeitverkehr, VoIP, Videokonferenzen	10^{-3}	100 ms	50 ms
1	Interaktiver Echtzeitverkehr, VoIP, Videokonferenzen	10^{-3}	400 ms	50 ms
2	Interaktiver, transaktionsorientierter Datenverkehr	10^{-3}	100 ms	Nicht spezifiziert
3	Interaktiver, transaktionsorientierter Datenverkehr	10^{-3}	400 ms	Nicht spezifiziert
4	Datenverkehr mit geringen Paket-verlusten, Massendaten, Streaming	10^{-3}	Nicht spezifiziert	Nicht spezifiziert
5	Best-Effort-Datenverkehr, IP	Nicht spezifiziert	Nicht spezifiziert	Nicht spezifiziert

Tabelle 3.1: Dienstgüte-Klassifizierung nach ITU-T Y.1541

[20] Vgl. Kapitel 3.3.5

Ein Vorteil bei der Verwendung von differenzierten Diensten ist, dass keine komplexe Implementierung für die Netzwerkentitäten erforderlich ist [Tanenbaum & Wetherall, 2012]. Die Integration der Klassifizierung wird mithilfe des IP-Protokolls vorgenommen. Mithilfe des Type-of-Service-Feldes im IP-Header, einem Headerfeld bestehend aus 8 Bit, kann das Paket nach möglichen Klassen markiert werden [Ibe, 2002]. Ein Router ist damit in der Lage das Datenpaket sofort nach Klassen differenziert zu behandeln. Andere IP-Operationen, die nicht durch differenzierte Dienste klassifiziert wurden, werden durch das Verfahren nicht beeinträchtigt. Zu unterscheiden sind drei verschiedene Modi, die durch die Differenzierung geschaltet werden können [Ibe, 2002]:

- Beim *Assured Forwarding* [Heinanen et al., 06/1999] werden vier Prioritätsklassen und drei Paketverwerfungsklassen definiert, wodurch zwölf unterschiedliche Behandlungsarten zur Verfügung stehen. Zu sendende oder am Router eintreffende Datenpakete werden zunächst einer der vier Prioritätsklassen zugewiesen. Nachdem die durchzuführende Dienstgüte überprüft wurde, wird den Datenpaketen eine der drei Verwerfungsklassen zugewiesen. Die Datenpakete werden dann, priorisiert von einem Scheduler (z.B. nach dem PQ oder WFQ Verfahren), verarbeitet und weitergeleitet.
- Das *Expedited Forwarding* [Davie et al., 03/2002] ist eine einfachere Form des erstgenannten Verfahrens. Hier werden Datenpakete lediglich als bevorzugt markiert, um z.B. eine verbesserte Verzögerungszeit, einen niedrigeren Jitter oder hohe Datenübertragungsraten zu ermöglichen. Markierte Datenpakete werden über eine priorisierte Warteschlange weitergeleitet und sollen damit ein „quasi"-verkehrsfreies Netzwerk durchlaufen. Aufgrund der Einfachheit der Priorisierung können Markierungen für Datenpakete netzwerkübergreifend verwendet werden. Netzwerke, die keine Priorisierung anbieten, werden dann nach Best-Effort-Verfahren durchlaufen.
- Beim *Default per-hop behaviour* werden die Datenpakete ohne Klassifizierung nach dem Best-Effort-Verfahren weiterverarbeitet.

3.3.4 Dienstgüte mit Integrierten Diensten (IntServ)

Integrierte Dienste bieten Garantien bei der Dienstgüte und wurden in Hinblick auf Echtzeitanwendungen sowie Multimedia-Streaming entwickelt [Ibe, 2002]. Dies wird üblicherweise durch die Reservierung von Ressourcen an Netzwerkknoten erreicht (ebd.). Das vorwiegend verwendete Protokoll der integrierten Dienste ist das *Resource Reservation Protocol (RSVP)* [Braden et al., 09/1997].

Beim RSVP wird zunächst eine Anfrage durch den Empfänger eines zukünftigen Datenstromes bei einem Sender gestellt. Der Sender sendet eine *Path Message*, die auf dem Weg zum Empfänger Informationen über den Verlauf des Pfades sammelt. Der

Empfänger antwortet mit einer *Reservierungsnachricht*, die an den Sender zurückgeschickt wird. Die Reservierungsnachricht nutzt dabei die vorher gesammelten Informationen, um den Routern den nächsten Pfadabschnitt mitzuteilen, an die die Reservierungsnachricht gesendet werden muss.

Auf dem Weg zum Sender wird an den Netzwerkknoten eine Dienstgüte-Reservierung durchführt. Wird der Reservierung stattgegeben und die Reservierungsnachricht gelangt bis zum Sender, ist eine Dienstgütereservierung auf dem zu sendenden Pfad durchgeführt worden. Abbildung 3.4 zeigt den grundsätzlichen Kommunikationsablauf von RSVP. Zu beachten ist hier, dass zunächst die erste Anfrage vom Empfänger einen anderen Routing-Pfad nehmen kann, als es die Path Message und die Reservierungsnachricht später tun.

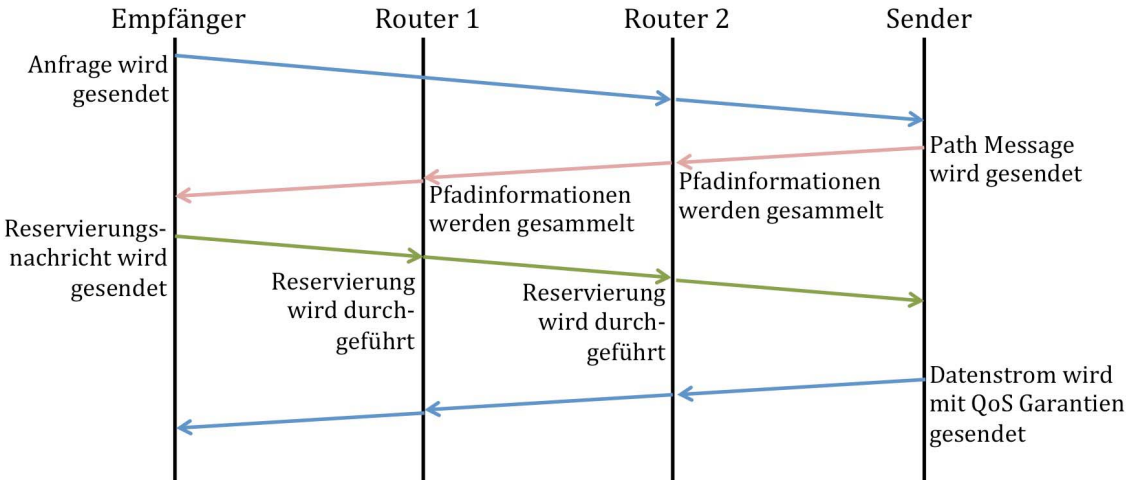

Abbildung 3.4: RSVP Kommunikationsablauf

Integrierte Dienste unterstützen drei Dienstklassen, die an den Knotenpunkten reserviert werden können [Ibe, 2002]:

1. Garantierter Dienst: Eine Dienstgütereservierung mit festen Verzögerungs-, Datendurchsatz- und Datenpaketverlustgarantierten.
2. Controlled Load-Dienst: Datensendungen werden nicht durch starken Datenverkehr beeinflusst. Verschiedene Verzögerungen, begrenzter Datendurchsatz sowie Datenpaketverluste existieren, werden aber geglättet.
3. Best-Effort-Dienst: Standardeinstellung ohne Reservierung nach Best-Effort Verfahren.

Die Verwendung der integrierten Dienste ist nicht weit verbreitet, da sie erhebliche Skalierungsprobleme mit sich bringen [Ibe, 2002]. Durch die grundsätzliche Natur von RSVP wird ein individueller Datenpfad für jeden Benutzer bzw. jede Benutzergruppe angelegt sowie Dienstgüte reserviert und verwaltet. Zusätzlich werden beständige Updates des Pfades mithilfe der RSVP-Signalisierungen benötigt. Speziell große Netzwerke leiden unter dieser Herangehensweise [Ibe, 2002]. Andere Verfahren, wie die bereits diskutierten differenzierten Dienste, oder das in Kapitel 3.3.5 beschriebene Multi Proto-

col Label Switching haben diese Technik daher bereits 2002 weitestgehend ersetzt (ebd.).

3.3.5 Traffic Engineering auf Datenflussaggregierter Ebene

Die obengenannten Techniken benennen Mittel, um ein möglichst gerechtes Netzwerk nach dem Best-Effort-Prinzip aufzubauen und eine individuelle Dienstgüte für ausgewählte Datenströme innerhalb eines solchen Netzwerks zu verbessern. Eine effiziente Lastverteilung oder auch Umlegung des Datenaufkommens innerhalb des Netzwerks auf Datenpfade mit besserer Dienstgüte lässt sich durch diese Methoden nicht realisieren [Fortz et al., 2002].

Die Analyse, Planung und Optimierung des Datenverkehrs innerhalb des Netzwerks ist das Ziel von *Traffic Engineering (TE)* (ebd.). TE bildet daher ebenfalls eine Perspektive zur Verbesserung der Dienstgüte, allgemein oder auch in individualisierter Form.

Eine Strategie ist die Umgestaltung der Netzwerkregulierung, d.h. die Entwicklung weg von einem autonomen System hin zu einem durch den NSP gesteuerten Netzwerk. Damit kann ein NSP nicht nur den Datenverkehr unabhängig von autonom agierenden Regelungen verwalten, sondern auch Dienstgüteklassifizierungen definieren und innerhalb des Netzwerkkonzepts umsetzen.

Multi Protocol Label Switching (MPLS) [Rosen et al., 01/2001] ist eine Technik, mit der ein NSP Dienstgüten und Übertragungspfade innerhalb des Netzwerks für einzelne Datenströme definieren kann. Dazu wird innerhalb der Vermittlungsschicht ergänzend zum IP-Paket ein zusätzlicher Header eingefügt, der ein sogenanntes MPLS-Label beinhaltet. Dieses Label wird an einem Knotenpunkt für den Index einer Tabelle verwendet, die ausführlichere Funktionen für die Dienstgüte und Weiterleitungsinformationen beinhaltet, als dies bei einer Routingtabelle der Fall wäre [Davie & Rekhter, 2000]. Der Zugriff auf diese Tabelle ist um einiges schneller als mit herkömmlichen Routingmethoden, da lediglich das MPLS-Label in der Tabelle nachgeschlagen werden muss und die Informationen ohne weitere Berechnungen zur Weiterleitung bereitstehen [Davie & Rekhter, 2000].

MPLS wird von NSPs zum Traffic Engineering verwendet, um die Lastverteilung im Netzwerk verwalten zu können und vorhersehbar zu machen. MPLS bietet die Basis für eine heterogene Applikationsstruktur, in der Echtzeitdienste wie IP-TV oder VoIP-Telefonie zusammen mit herkömmlichem Internetverkehr innerhalb eines Netzwerks zusammengebracht werden können [Tanenbaum & Wetherall, 2012]. Datenströme sind durch die Zuweisung von Netzwerkpfaden und vordefinierbarer Dienstgüte unter Kontrolle zu bringen und steuerbar.

Ein weiteres Verfahren, um den Netzwerkverkehr auf datenflussaggregierter Ebene zu steuern, sind Software Defined Networks (SDN) [Open Networking Foundation, 2012]. Hierbei wird eine Trennung zwischen den eigentlichen Datenflüssen und den für das Netzwerk nötigen Steuerungsdaten vorgenommen. Eine direkte Einflussnahme auf die Netzwerkgeräte innerhalb eines Netzwerks ist somit von einem beliebigen Standort

aus realisierbar. Dies eröffnet Möglichkeiten, Dienstgüten netzwerkweit zu integrieren und für bestimmte Datenflüsse einzurichten (ebd.).

3.3.6 Dienstgüteverbesserungen im Ende-zu-Ende Betrieb

Die oben vorgestellten Maßnahmen betreffen direkt die im Netzwerk eingestellten Geräte. Ein NSP kann so eine optimale Lastenverteilung erreichen und ist in der Lage, hohe Kapazitäten zu verkaufen. Höhere Dienstgüten können erreicht werden, indem ein Kunde beim NSP diese explizit reservieren lässt. Differenzierte Dienste erlauben dem Kunden ohne explizite Reservierungen mit Mehrkostenaufwand, höhere Dienstgüten zu erreichen. Dazu muss aber der NSP diese Techniken innerhalb seines Netzes unterstützen. Durch Verträge zwischen den NSPs können übergreifende Dienstgüteregeln vereinbart werden. Eine weltweite Einflussnahme über NSP-Grenzen hinweg ist jedoch nicht zwingend gegeben und für einen Endkunden nur mit hohem Aufwand zu bewerkstelligen, wenn die entsprechenden Verträge zwischen den NSPs bestehen.

Alternative Vorgehensweisen für Dienstgüteverbesserungen bieten sich in höheren Netzwerkschichten an, die auf den Eigenschaften eines Best-Effort-Netzes aufbauen und Mechanismen anbieten, die einen höheren Dienstgütegrad bereitstellen können. Diese Ende-zu-Ende-Techniken zielen auf:

- Verbesserung der Sendeeigenschaften in Hinblick auf Überlast-Vermeidung und -Kontrolle
- Verbesserung der Empfangseigenschaften in Hinblick auf die Beibehaltung des Sendemusters
- Wiederherstellung von verlorenen Datensegmenten durch wiederholtes Senden (ebd.)
- Verlust-Verschleierung durch Kodierung [Briscoe et al., 2014]
- Fehlervermeidung durch Redundanz (ebd.)

Überlastmechanismen funktionieren in der Regel durch Sondierung des Übertragungskanals und Anpassung des Sendeverhaltens. Diese werden innerhalb der Transportprotokolle realisiert. Eine Verbesserung hinsichtlich eines getakteten Sendemusters auf Empfangsseite wird mithilfe eines Zwischenspeichers erreicht, der ankommende Pakete kurzzeitig speichert und dann nach dem erwarteten Taktmuster an eine Anwendung weiterleitet. Zwischenspeicher werden ebenfalls in ARQ-Mechanismen eingesetzt, um eine Wiederherstellung von verlorengegangenen Datensegmenten durch wiederholtes Senden zu erreichen.

Dienstgüteverbesserungen bei der Datenübertragungsrate, die eine Verringerung der Daten zur Folge haben, werden in der Regel durch Kodierung erreicht. Spezialisierte Kodierungsverfahren werden bei Echtzeitübertragungen mit einer bestimmten Fehlertoleranz, wie z.B. Video oder Audio, angewandt. Diese können die Datenübertragungsrate dynamisch hinsichtlich der Qualität justieren. Verluste werden in dieser Form „verschleiert“. Kodierung wird ebenfalls zur Fehlervermeidung angewandt, indem zusätzliche redundante Elemente den zu sendenden Informationen hinzugefügt werden.

3.3.6.1 Zwischenspeicherung (Buffering)

Zwischenspeicher können allgemein dafür eingesetzt werden, unregelmäßig eintreffende Daten in eine regelmäßige Anordnung zu bringen. Applikationen, die eine regelmäßige Sendung von Daten als Dienstgüte-Anforderung (Verzögerungsvarianz, Jitter) voraussetzen, können mithilfe von Zwischenspeichern in die Lage versetzt werden, trotz unstetiger Situationen innerhalb des Netzwerks die Daten in geglätteter und in gewünschter Form an die Applikation weiterzuleiten.

Jitter-Kompensation wird mithilfe eines Speichers an der Empfangsseite durchgeführt. Eine Weiterleitung an die Applikation geschieht entweder explizit durch Informationen der Applikation selbst, wie z.B. mithilfe von Zeitstempeln innerhalb der Datenpakete [Van, 2004], oder implizit mithilfe eines berechneten Erwartungswerts, der aus den vorher empfangenen Datenpaketen resultiert [Schulzrinne et al., 7/2003]. Abhängig von der Größe des Zwischenspeichers kann Jitter nur bis zu einem gewissen Grad geglättet werden. Je höher der Jitter, desto größer muss auch der Zwischenspeicher sein. Eine Zwischenspeicherung und die damit erfolgte Vorhaltezeit vergrößert die allgemeine Verzögerung, bis das Datenpaket an die Applikation weitergeleitet wird.

3.3.6.2 Kodierung im Ende-zu-Ende Betrieb

Daten können mit Kompressionsalgorithmen der *Quellenkodierung* oder der *Entropiekodierung* verringert werden [Fluckiger, 1995]. Durch die Verringerung der Datensendung kann der Datendurchsatz maximiert werden. Eine Quellenkodierung fokussiert auf die Quelle, bzw. die Art der zu verwendenden Daten und nutzt die Eigenschaften der Applikation, um die Datenmenge zu reduzieren (ebd.). Sie kann verlustlos oder verlustbehaftet sein (ebd.). Eine verlustbehaftete Kompression zur Datenreduktion kommt vor allem in Video- oder Audioübermittlung vor sowie bei der Kompression von Bildern. Dabei werden z.B. unhörbare Frequenzen, Stille im Signal oder einfarbige Bereiche in Bildern herausgefiltert und zusammengefasst.

Bei der Entropiekodierung werden Abfolgen von Informationen durch kleinere Codefragmente ersetzt. Eine einfache Variante ist das Ersetzen von sich wiederholenden Symbolen oder Datenblöcken durch einfachere Sequenzen oder durch spezielle Markierungssymbole [Fluckiger, 1995]. Statistische Methoden werden dazu genutzt, eine verlustfreie Kompression mithilfe von Substitution zu erreichen. Datensätze werden hierbei nach ersetzbaren Sequenzen mit dem Kriterium einer bestimmten Häufigkeitswahrscheinlichkeit durchsucht und durch kürzere Datenblöcke ersetzt. Diese Art der Kodierung wird häufig zur Datenübermittlung in der Bitübertragungsschicht verwendet, findet aber auch durch z.B. den Lempel-Ziv-Algorithmus [Ziv & Lempel, 1977] Anwendung zur Komprimierung von Dateien durch den Benutzer selbst.

Formen der Kodierung zur Fehlerkontrolle werden vornehmlich zur Fehlererkennung und -korrektur innerhalb der Bitfolge bzw. Rahmen oder Datenpakete in den Kommunikationsprotokollen der unteren Netzwerkschichten verwendet. Kodierung kann auch dazu genutzt werden, um komplett verlorengegangene Datensegmente auf

Ende-zu-Ende-Ebene auszugleichen. Nach [Bossert & Breitbach, 1999] wird die *Kanalkodierung* definiert als „die zuverlässige Übertragung von Daten durch Fehlervorwärtskorrektur [*Forward Error Correction, FEC*] bzw. Fehlererkennung“ und eine daraus resultierende Wiederholungsanfrage (ARQ). Fehlererkennung wird hauptsächlich in der Sicherungsschicht durchgeführt, um fehlerhafte Pakete zu verwerfen oder bis zu einem bestimmten Grad zu korrigieren (ebd.). Sie kann aber auch in einfacher Form in allen anderen Netzwerkschichten eingesetzt werden. Dabei werden redundante Informationen hinzugefügt, damit Fehler erkannt werden können (ebd.). Bei der Kodierung wird der physikalische Kanal in einen Kanal mit besserer BER transformiert (ebd.).

Auf Ende-zu-Ende-Ebene kann Kodierung dazu verwendet werden, mithilfe von *Loss-Recovery* Methoden verlorene Datensegmente auszugleichen. Ein Datensegment gilt als verloren, wenn es:

a) beim Empfänger mit fehlerhaften Daten ankommt und verworfen werden muss;
b) nie beim Empfänger ankommt [Kurose & Ross, 2014];
c) nach seinem vorgesehenen Nutzungszeitpunkt ankommt (ebd.).

Ein verlorenes Datensegment kann durch die Verwendung eines zusätzlichen Paritätssegments wiederhergestellt werden. Ein solches Segment kann nach [Shacham & McKenney, 1990] mithilfe einer Exklusiv-Oder-Verknüpfung zwischen n Originaldatensegmenten erstellt werden. Eine Datensendung beinhaltet damit $n+1$ Datensegmente (ebd.). Durch die Verknüpfung kann ein einzelner Verlust eines Datensegments ausgeglichen werden, indem auf Empfängerseite wiederum eine Exklusiv-Oder-Verknüpfung zwischen den verbleibenden n Datensegmenten durchgeführt wird (ebd.). Kommen alle Datensegmente an, können die Paritätssegmente verworfen werden. Ähnliche Verfahren bieten die Möglichkeit, durch weitere Paritätssegmente in anderen Anordnungen mehr als ein Datensegment wiederherzustellen [Shacham & McKenney, 1990].

Durch die Nutzung einer FEC kommt es zu einer erhöhten Datenübertragungsrate, die durch die zusätzliche Redundanz benötigt wird [Bolot & Garcia, 1996]. FEC kann dazu dienen, auf fehleranfälligen Übertragungsstrecken fehlende Datensegmente auszugleichen. Dies kann die Latenzzeit verbessern, da nicht auf eine wiederholte Sendung des fehlenden Datensegments gewartet werden muss (ebd.). Gleichzeitig wird durch den Wiederherstellungsprozess mehr Zeit in Anspruch genommen, da auf alle Daten der gesamten Gruppe gewartet werden muss bis sie eingetroffen sind (ebd.).

Probleme ergeben sich, wenn mehrere Datensegmente hintereinander unter Verlust zu leiden haben und die Daten somit nicht wiederhergestellt werden können. Datensegmentverluste sind aber häufig miteinander korreliert, da sie durch vollgelaufene Warteschlangen entstehen. Daher sinkt die Effektivität einer FEC bei größer werdender Anzahl von Datensegmentverlusten [Bolot, 1993]. Daneben kann eine Datensendung mit FEC wegen der zusätzlichen Datensegmente, die den Datenburst vergrößern, zu zusätzlichem, stoßweisem Verhalten führen, das sich negativ auf die Sendeeigenschaften im Netzwerk auswirkt [Fluckiger, 1995].

[Smithwick, 1995] beschreibt die positiven Aspekte einer FEC für Anwendungen der Telemedizin, wenn bei niedrigen Übertragungsraten für die Übertragung von z.B. Roboterkontrolldaten Kommandos mehr als einmal gesendet werden. Diese Strategie ist eine progressive redundante Möglichkeit, die Fehlerwahrscheinlichkeit zu verringern, steht jedoch im Kontrast zur Problematik eines korrelierten Datenpaketverlusts. Hinzu kommt der Einfluss von Frequenz und Datenpaketgröße auf die im Netzwerk bestehenden Warteschlangen. Ein progressives vermehrtes Senden von Daten erhöht zudem die Überlast. Um einer möglichen Korrelation von konsekutiven Datenpaketverlusten zu entgehen, müssten redundante Pakete mit größeren Abständen zueinander gesendet werden, was wiederum die Verzögerung bei Datenpaketverlust erhöht [Liang et al., 2001].

3.3.6.3 Grundlagen des TCP-Protokolls

Das Transmission Control Protocol (TCP) ist ein zuverlässiges, verbindungsorientiertes Protokoll der Transportschicht. Die erste Methodik wurde durch [Cerf & Kahn, 1974] veröffentlicht und 1981 durch die Internet Engineering Task Force (IETF) standardisiert [Postel, 09/1981]. Sie ist seitdem bis heute Gegenstand vieler Forschungen und Verbesserungen.

Daten, die von den anwendungsorientierten Schichten an die Transportschicht weitergegeben werden, werden in Datensegmente eingeteilt und mithilfe einer Ende-zu-Ende-Verbindung zu einem Empfangs-Endpunkt gesendet. TCP ist verbindungsorientiert, da Sender und Empfänger zunächst durch einen Verbindungsprozess gehen müssen, der einen logischen Kanal aufbaut, ihn aufrecht erhält und schließlich wieder abbaut [Tanenbaum & Wetherall, 2012]. Eine TCP-Verbindung wird mithilfe eines Ports an den Endpunkt bzw. an die Applikation gebunden [Postel, 09/1981]. In Kombination mit dem IP-Protokoll wird eine eindeutige TCP-Verbindung zwischen zwei Endpunkten durch das folgende Quadrupel identifiziert [Fall & Stevens, 2012]:

Quell IP-Adresse, Quell Port, Ziel IP-Adresse, Ziel Port

Die Zuverlässigkeit des Protokolls wird mithilfe von ARQ-Mechanismen, Fehlererkennung und Sequenznummern gewährleistet. Eine Fehlererkennung von fehlerhaft übertragenen Symbolen wird durch eine im Header mitgesendete Checksumme realisiert, die bei Empfang der Daten mit einer durch den Empfänger berechneten Checksumme überprüft wird. Die Checksumme errechnet sich unter TCP, indem ein Segment in 16-Bit-Worte unterteilt wird und diese als Einer-Komplement aufsummiert werden [Kurose & Ross, 2014]. Zusätzlich zum TCP-Header und den angehängten Daten wird bei dieser Rechnung ein Pseudo-Header miteinbezogen, der aus der Quell-IP, der Ziel-IP, der Protokollversion und der Länge des TCP-Segments generiert wird.

Sequenznummern werden zur Synchronisierung der Daten zwischen Sender und Empfänger verwendet. Jedes gesendete Byte an Nutzdaten wird aufsteigend nummeriert und in Datensegmente zusammengefügt [Fall & Stevens, 2012]. Ein Datensegment erhält eine 32-Bit-Sequenznummer, die durch die Nummerierung des ersten im Segment

enthaltenen Datenbytes angegeben wird (ebd.). Hierdurch ist es dem Empfänger möglich, die Daten in der richtigen Reihenfolge zusammenzusetzen sowie Lücken und Duplikate zu erkennen.

TCP arbeitet mit einem ARQ-Protokoll, welches sicherstellt, dass die Daten beim Empfänger ankommen. Ein gesendetes Datensegment mit Nutzdaten enthält die Sequenznummer des ersten Datenbytes. Ein Empfänger antwortet darauf mit einem Bestätigungssegment durch ein gesetztes ACK-Bit (Acknowledge). Bei dem Bestätigungssegment wird die Sequenznummer des nächsten zu erwarteten Datenbytes mitgesendet. Wurde ein Segment nicht empfangen und es entsteht eine Lücke innerhalb der empfangenen Daten, werden ACK-Segmente stets mit derselben Sequenznummer versendet (duplicate ACK), damit der Empfänger weiß, ab wann Daten nicht angekommen sind.

Das ACK-Bestätigungssegment besitzt zusätzlich eine eigene Sequenzierung, damit auch diese in der richtigen Reihenfolge empfangen werden kann. Wird ein Datensegment gesendet, wird ein Timer gestartet, der erst durch den Empfang des entsprechenden ACK-Segments gelöscht wird. Bei Ablauf des Timers (Timeout) wird von einem Datensegmentverlust ausgegangen und eine Neusendung des Datensegments veranlasst.

Ein Timeout errechnet sich nach [Jacobson, 1988] durch die zu erwartende Zeit, die es benötigt, bis die Daten beim Empfänger angekommen sind und eine Bestätigung über den Erhalt wieder zurückkommt – inklusive einer möglichen Variabilität. Dies wird durch die aktuelle geglättete RTT (smoothed RTT, SRTT) und zuzüglich dem vierfachen der mittleren Abweichung der RTT (RTTVAR) errechnet [Jacobson, 1988] [Paxson & Allman, 11/2000]:

$$Timeout = SRTT + 4 * RTTVAR \tag{3.1}$$

SRTT und *RTTVAR* werden mithilfe eines Tiefpassfilters berechnet, um nicht allzu empfindlich auf plötzliche Änderungen des Netzwerks zu reagieren [Jacobson, 1988]:

$$SRTT = alpha * SRTT + (1 - alpha) * RTT \tag{3.2}$$

$$RTTVAR = beta * RTTVAR + (1 - beta) * |SRTT - RTT| \tag{3.3}$$

wobei *RTT* der aktuelle Messwert der momentanen RTT ist. Der Wert *alpha* wird in den meisten TCP-Implementierungen mit *alpha* = 7/8 angegeben. Der Wert *beta* hingegen beträgt *beta* = 3/4 [Fall & Stevens, 2012]. In der weiteren Entwicklung der in dieser Arbeit beschriebenen Protokollmodifikation wird dieses Verfahren für verschiedene Berechnungen benutzt.[21]

Gesendete Daten müssen innerhalb eines Pufferspeichers vorgehalten werden, bis ein entsprechendes ACK-Segment eingetroffen ist, da sie im Falle eines Fehlers erneut gesendet werden müssen. Bei diesem Schiebefensterverfahren wird die Länge des *Sendefensters* zunächst mithilfe der *Flusskontrolle* bestimmt [Fall & Stevens, 2012]. Hierbei ist es nötig, den momentan möglichen maximalen Dateninput – die Größe des *Empfangsfensters* und die *maximale Segmentgröße* – des Empfängers zu kennen. Das Emp-

[21] Vgl. hierzu Kapitel 6

fangsfenster bildet die oberste Schranke des Sendefensters und schränkt damit die maximal hintereinander zu sendenden Daten ein. Die maximale Größe des Empfangsfensters entspricht der Größe der vom Netzwerk aufzunehmenden Daten für die Zeit bis Segmentverluste ausgeglichen werden können und das Empfangsfenster geleert werden kann. Dies entspricht dem zweifachen des *Übertragungsrate-Verzögerungs-Produkts* (Bandwidth-Delay-Product, BDP) [Ford et al., 03/2011]:

$$BDP = Datenübertragungsrate * OWD \tag{3.4}$$

$$Max\ Empfangsfenster = 2 * BDP \tag{3.5}$$

Die Größe des momentanen verbleibenden Speicherbereichs im Empfangsfenster wird mit jedem ACK-Segment mitgesendet. Läuft der Speicher beim Empfänger voll, weil die Daten momentan nicht weiterbearbeitet werden können, so ist dieser Wert Null und der Sender kann nicht weitersenden. Wird im Empfangsfenster Speicherplatz frei, teilt dies der Empfänger dem Sender über ein unbestätigtes ACK-Segment mit. Zugleich fragt der Sender in regelmäßigen Abständen nach, ob das Empfangsfenster wieder aufnahmebereit ist. Ein ACK-Segment des Empfängers mit der benötigten Fenstergröße beantwortet diese Anfrage [Fall & Stevens, 2012].

ARQ-Verfahren können im Fehlerfall die Latenz durch das wiederholte Senden von Datenpaketen um ein Mehrfaches erhöhen. Dies ist kritisch bei der Übertragung von Echtzeitdaten. Die hieraus entstehende Verzögerung ist einer der dominierenden Latenzfaktoren bei über das Internet genutzten Anwendungen [Flach et al., 2013]. Speziell bei zuverlässigen Echtzeitübertragungen sind weggefallene Datenpakete von besonderer Tragweite, da sie die Ausgabe der bereits korrekt empfangenen Daten blockieren (Head-of-Line Blocking). Durch das ARQ-Verfahren wird nach anschließendem Erhalt der gesamte Datenblock plötzlich freigeben. Dies führt nicht nur zu vermehrter Latenz, sondern auch zu erhöhtem Jitter. Mögliche Methoden, um diese Verzögerungen unter TCP zu minimieren, sind die Verringerung der Erkennungszeit eines Datenpaketverlusts sowie die initiative Sendung vermeintlich verlorener Datenpakete [Briscoe et al., 2014] [Stevens, 01/1997].

Der Aufbau einer Verbindung geschieht über einen Drei-Wege-Handshake. Hierbei werden drei Nachrichten hintereinander zwischen einem Client und einem Server ausgetauscht [Postel, 09/1981]. Der Client, der die Verbindung eröffnen möchte, sendet ein Datensegment mit einem gesetzten SYN-Bit – für die Initiierung einer Verbindung – zusammen mit einer anfänglichen Initialisierungs-Sequenznummer (ebd.). Der Server empfängt das Datensegment. Ist der Server bereit für eine Verbindung, antwortet er mit einem Bestätigungssegment in Form eines gesetzten SYN/ACK-Bits, um zu signalisieren, dass die erste Nachricht angekommen ist (ebd.). Ist der Server nicht bereit, wird die Verbindung mit einem Antwortsegment mit RST-Bit beendet. Das SYN/ACK-Segment enthält die vorher empfangene Sequenznummer +1 – also das nächste zu erwartende Datenbyte – sowie eine eigene Initialisierungs-Sequenznummer für die Bestätigungsnachricht selbst (ebd.). Der Client bestätigt diese Nachricht wiederum mit einem gesetzten ACK-Bit und den beiden aufgestockten Sequenznummern. Die Verbindung ist damit

hergestellt und Daten können gesendet werden. Beide Instanzen sind nun gleichberechtigt, d.h., es wird nicht mehr zwischen Client und Server unterschieden. Abbildung 3.5 veranschaulicht den Vorgang.

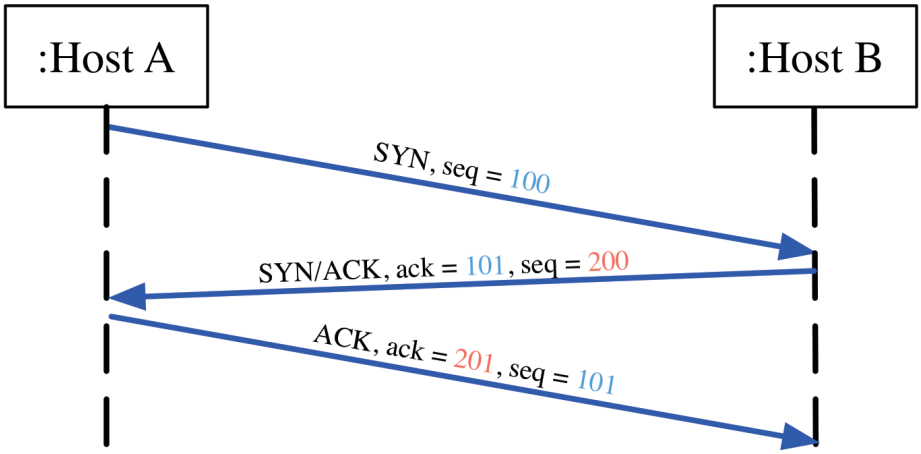

Abbildung 3.5: Drei-Wege-Handshake bei TCP

Der Verbindungsabbau gestaltet sich ähnlich. Ein Endpunkt sendet ein Segment mit einem gesetzten FIN-Bit. Daraufhin sendet die Gegenstelle ein ACK-Segment, ist aber noch in der Lage, weiterhin Daten zu übertragen bzw. zu empfangen. Diese halb geschlossene Verbindung wird dann geschlossen, wenn ebenfalls von der Gegenstelle ein FIN-Segment gesendet wird. Dieses wird wiederum von der anderen Seite mit einem ACK-Segment bestätigt. Die Verbindung ist geschlossen. [Fall & Stevens, 2012]

Die Vorgehensweise beim Auf- und Abbau einer TCP-Verbindung wird am besten über das Zustandsdiagramm beschrieben (siehe Abbildung 3.6) [Wehrle et al., 2002]. Der anfängliche Zustand eines Endpunkts ist *Closed*. Erwartet der Endpunkt eine Verbindung, befindet er sich im passiven Zustand *Listen*, bei dem er auf hereinkommende Verbindungsanfragen wartet. Wenn der Endpunkt eine Verbindung öffnen möchte, wechselt er in den Zustand *SYN_sent*, nachdem ein SYN-Segment gesendet wurde. Bei Empfang dieses Segments wird stattdessen in den *SYN_recv*-Zustand gewechselt und ein SYN/ACK-Segment gesendet. Nach dem Senden bzw. Empfang eines ACK-Segments ist die Verbindung hergestellt und es wird in den Zustand *Established*, dem Zustand für die Datenübertragung, gewechselt.

Um die Verbindung aktiv zu schließen, wird ein FIN-Segment gesendet und in den Zustand *FIN_wait_1* gewechselt. Die Gegenstelle empfängt das FIN-Segment und wechselt in den *Close_wait* – den Zustand für ein passives Schließen der Verbindung. Sie sendet ein ACK-Segment und schließt die Verbindung, wenn keine weiteren Daten mehr anstehen, indem sie unter Senden eines eigenen FIN-Segments in den Zustand *Last_ACK* wechselt. Der Endpunkt, der das Schließen initiiert hat, wechselt durch Empfang des ACK-Segments in den Zustand *FIN_wait_2*, um gegebenenfalls noch weitere Daten zu empfangen. Erhält er das FIN-Segment, wird in den *Time_wait*-Zustand unter Senden eines ACK-Segments gewechselt, wo eine Standardzeit gewartet wird, um sicherzustellen, dass die Verbindung nicht wieder verwendet wird, bevor das ACK-

Segment die Gegenstelle erreicht. Eine weitere Möglichkeit ist hier, dass die Gegenstelle ebenfalls ein FIN-Segment sendet und somit zur selben Zeit die Intention eines Verbindungsabbaus besteht, was sich in dem Zustand *Closing* ausdrückt. Der Zustand *Last_ACK* der Gegenstelle wird geschlossen, wenn das ACK-Segment empfangen wird.

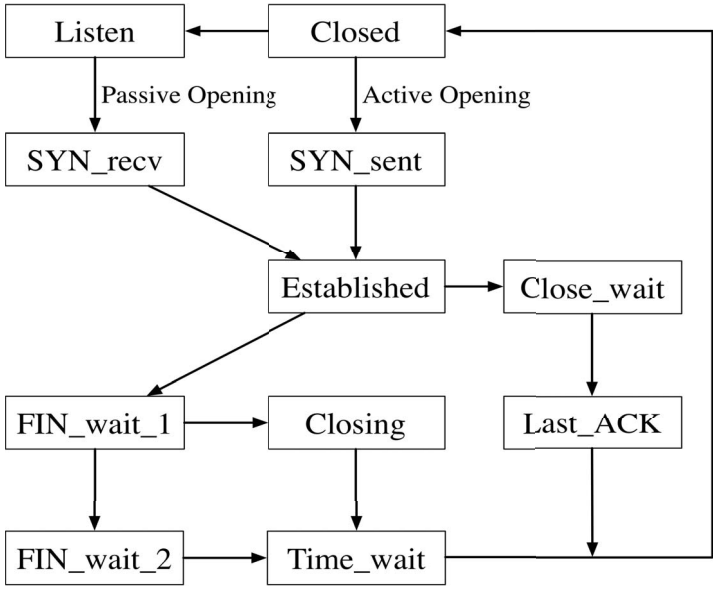

Abbildung 3.6: Vereinfachtes TCP Zustandsdiagramm für Verbindungsaufbau und Abbau nach [Wehrle et al., 2002]

Bei der Übertragung von Daten geht das grundlegende Bestreben dahin, dass die Daten so schnell wie möglich gesendet werden (maximale Verbindungsausnutzung) [Leon-Garcia & Widjaja, 2004]. Dies ist in erster Linie von dem mit dem NSP ausgehandelten SLA für den Netzzugang abhängig. In einem AS wird durch die Knotenpunkte autonom entschieden, welches die kostengünstigsten Routen zu einem Ziel sind. Wenn innerhalb des Systems zu viel Netzwerkverkehr – unabhängig vom ausgehandelten SLA – über eine bestimmte Route geleitet wird, kommt es zu Stausituationen, bei denen Datensegmente verworfen werden müssen.

Die Überlastkontrolle von TCP ist eine Funktion auf Datenfluss-Ebene, um den gesendeten Datenstrom mithilfe von Feedbackinformationen auf überlastungsbedingte Situationen einzustellen und die Datenflussrate zu regulieren [Leon-Garcia & Widjaja, 2004]. Sie besteht im Ganzen aus vier Algorithmen, die in [Jacobson, 1988] und [Jacobson, 1990] entwickelt wurden:

1. Slow Start
2. Congestion Avoidance
3. Fast Retransmit
4. Fast Recovery

Eine aktualisierte Standardisierung wird in [Allman et al., 9/2009] beschrieben. Die grundsätzliche Funktionsweise wird durch das Messen von Segmentverlusten erreicht. Läuft der bereits erläuterte Timer ab, wird von einem Segmentverlust ausgegangen. Je mehr Segmente fehlerfrei gesendet werden können, desto geringer ist die Überlastung.

Die Anzahl der gesendeten Segmente wird durch die Größe des Sendefensters bestimmt. In diesem Fall wird dieses Fenster Überlastungsfenster (Congestion Window, cwnd) genannt [Tanenbaum & Wetherall, 2012]. Die Größe des Überlastungsfensters entspricht der Anzahl der möglichen Bytes, die das Netzwerk aufnehmen kann, bevor eine Bestätigung durch den Empfänger erfolgen muss [Allman et al., 9/2009].

Zu Beginn einer Übertragung mit unbekannten Netzwerkeigenschaften oder nachdem eine Überlastung durch einen Timeout entdeckt wurde, werden beim *Slow Start-Algorithmus* Datensegmente zunächst langsam gesendet, um die verfügbare Kapazität des Netzwerks herauszufinden und das Netzwerk nicht mit einer großen Datenmenge zu überfluten (ebd.).

Die anfängliche Größe des Überlastungsfensters wird durch die *maximale Segmentgröße (Maximum Segment Size, MSS)* bestimmt. Sie bildet hiervon das zwei bis vierfache (je nach Größe der MSS) [Allman et al., 9/2009]. Die MSS bestimmt sich durch die größte Segmentgröße, die ein Endpunkt übertragen kann. Der Wert wird mithilfe des *Maximum Transmission Unit (MTU)* Algorithmus berechnet [Mogul & Deering, 11/1990] [Mathis & Heffner, 03/2007], der hier nicht weiter ausgeführt werden soll. Der Wert der MSS liegt bei Benutzung von Ethernet zwischen *64 Byte* und *1460 Byte* [Postel, 11/1983].

Während der Benutzung des Slow Start-Algorithmus wird das Überlastungsfenster maximal immer um so viele weitere MSS vergrößert, wie Daten kumulativ bestätigt werden, sobald ein ACK-Segment eintrifft [Allman et al., 9/2009]. Der Slow Start-Algorithmus endet, wenn die Größe des Überlastungsfensters die maximale Größe eines Schwellwerts (d.h. des Sendefensters) erreicht oder wenn eine Überlastung beobachtet wird (ebd.). Erreicht die Größe des Überlastungsfensters die Größe des Schwellwerts, so beginnt die Phase des *Congestion Avoidance Algorithmus* (ebd.). Hierbei wird das Überlastungsfenster linear um eine Segmentgröße pro RTT vergrößert (ebd.). Der Algorithmus wird beendet, wenn eine Überlastung auftritt oder wenn das vorher ausgehandelte maximale Empfangsfenster erreicht wird (ebd.).

Wird eine Überlastung festgestellt, wird der Schwellwert verkleinert und Slow Start startet erneut. Die neue Größe des Schwellwerts wird mit der folgenden Formel errechnet (ebd.):

$$Slow\ Start\ Schwellwert = max\ (FlightSize\ /\ 2,\ 2*MSS) \tag{3.6}$$

wobei *FlightSize* die Größe aller noch unbestätigten Segmente innerhalb des Netzwerks darstellt. Abbildung 3.7 zeigt eine Darstellung des Vorgangs als Liniendiagramm. Darin ist zu erkennen, dass der Slow Start nach einem Segmentverlust verkürzt wird und daher mehr Zeit während des Congestion Avoidance-Algorithmus verbracht wird.

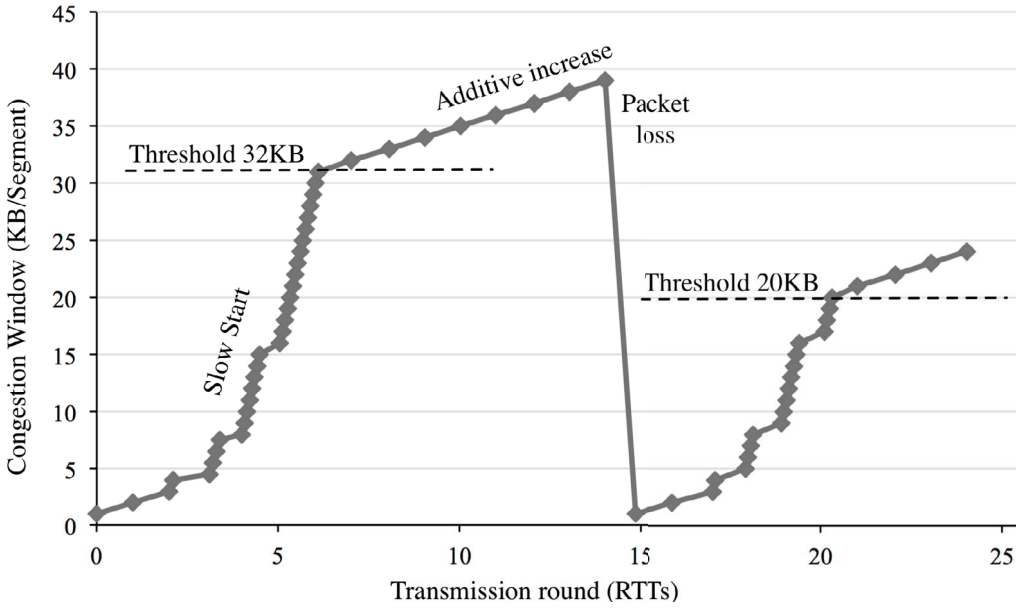

Abbildung 3.7: Slow Start und Congestion Avoidance Algorithmus nach [Tanenbaum & Wetherall, 2012]

Die Algorithmen *Fast Retransmit* und *Fast Recovery* werden dazu verwendet, um Fehler schneller beim Sender anzuzeigen und auf sie einzugehen. Sie werden gestartet, wenn beim Sender ein dupliziertes ACK-Segment mit gleicher Sequenznummer ankommt [Allman et al., 9/2009]. Diese duplizierten ACK-Segmente weisen auf Datensegmente hin, die entweder außerhalb der Reihe (out-of-order) beim Empfänger angekommen sind oder gar nicht empfangen wurden. Nachdem drei dieser ACK-Duplikate angekommen sind, wird das verlorene Segment direkt gesendet, ohne einen Timeout abzuwarten, was die Zeit einer Fehlerkorrektur verringert (ebd.). Da ansonsten die meisten Segmente ankommen, wird bei diesem Fehler kein Slow Start-Algorithmus in Gang gesetzt. Das Überlastungsfenster wird auf den Schwellwert zurückgesetzt plus die drei noch ausstehenden Segmente, die noch nicht gesendet wurden. Der Congestion Avoidance-Algorithmus wird fortgesetzt (ebd.).

Abbildung 3.8 zeigt ein Zustandsdiagramm der TCP-Überlastkontrolle. Die Überlastkontrolle kennt drei Zustände, die jeweils die einzelnen Algorithmen darstellen, die den oben beschriebenen Abläufen folgen.

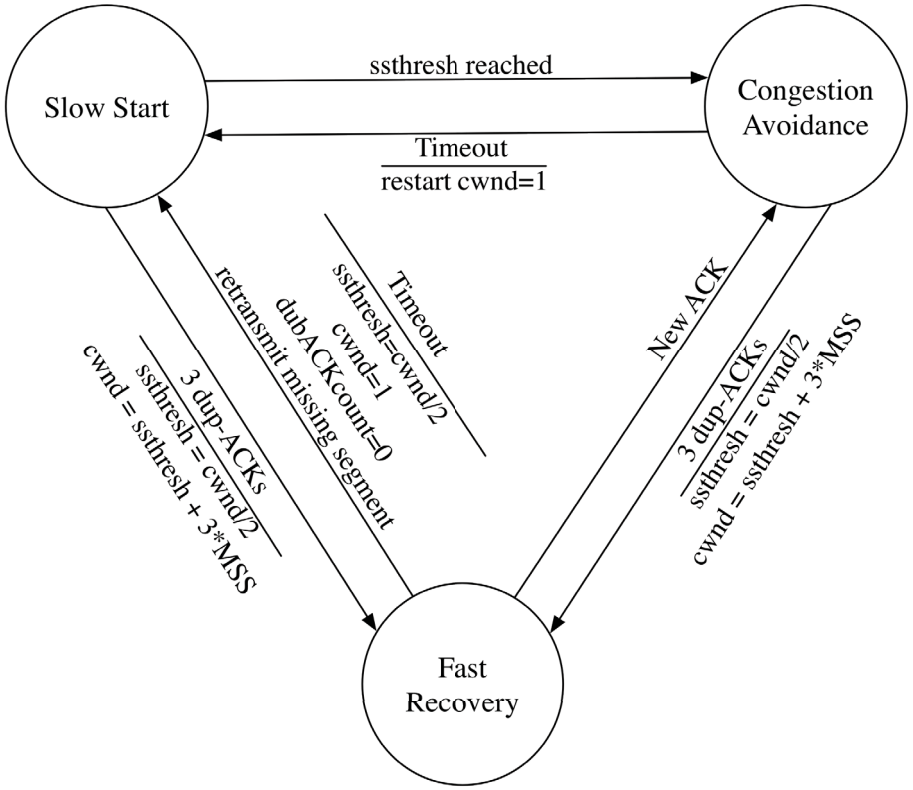

Abbildung 3.8: Zustände der TCP Überlastkontrolle

Für die Implementierung der TCP-Überlastkontrolle wurden Leitungszustände definiert, die als Kernel-Parameter auslesbar sind. Diese Leitungszustände bilden die Basis für die Implementierung der Zustandsübergänge des oben beschriebenen Algorithmus [Seth & Venkatesulu, 2008]. Diese werden in den späteren Kapiteln genutzt, um die in dieser Arbeit entwickelten adaptiven Algorithmen für eine Verbindungsanpassung zu realisieren. Die folgenden Kernel-Parameter definieren den Zustand der Verbindung (ebd.):

- *TCP_CA_OPEN:* Es gibt zurzeit keine Überlastsituation;
- *TCP_CA_RECOVERY*: Es kann in einen Recovery-Zustand gewechselt werden, wenn eine Überlastsituation erkannt wurde aufgrund von drei Duplicate-ACKs oder eines Timeouts;
- *TCP_CA_DISORDER:* Eine gestörte Übertragung der Segmente wurde durch den Empfang von einschlägigen ACK-Segmenten (Überlastung anzeigende) detektiert;
- *TCP_CA_CWR:* Das Überlastungsfenster wurde verringert, da eine Überlast detektiert wurde;
- *TCP_CA_LOSS:* Ein Timeout wurde ausgelöst und ein Segment wurde verloren.

Die Überlastkontrolle von TCP ist bis heute Objekt intensiver Forschung. Untersuchungen zielen auf bessere Erkennung von Überlastungen sowie Verkürzung der Erkennungszeiten [Petlund et al., 2008] [Opstad et al., 2015].

Das TCP-Protokoll besitzt einen Header, welcher es erlaubt, die beschriebenen Funktionen umzusetzen. Jedes Segment besitzt den in Abbildung 3.9 dargestellten Aufbau. Charakteristisch für das TCP-Protokoll ist die Adressierung über Ports, die zusammen mit der IP-Adresse aus dem IP-Protokoll einen Endpunkt bilden. *Quell-Port* und *Ziel-Port* bilden die ersten *32 Bit* des Segments. Es folgt die *Sequenznummer* für die Kennzeichnung des ersten Datenbytes innerhalb des Segments sowie die *Bestätigungsnummer*, die die Sequenznummer für das nächste erwartete Byte angibt. Die Bestätigungsnummer ist nur mit gesetztem ACK-Bit gültig, wenn das Segment eine Bestätigung für ein erhaltenes Datensegment ist.

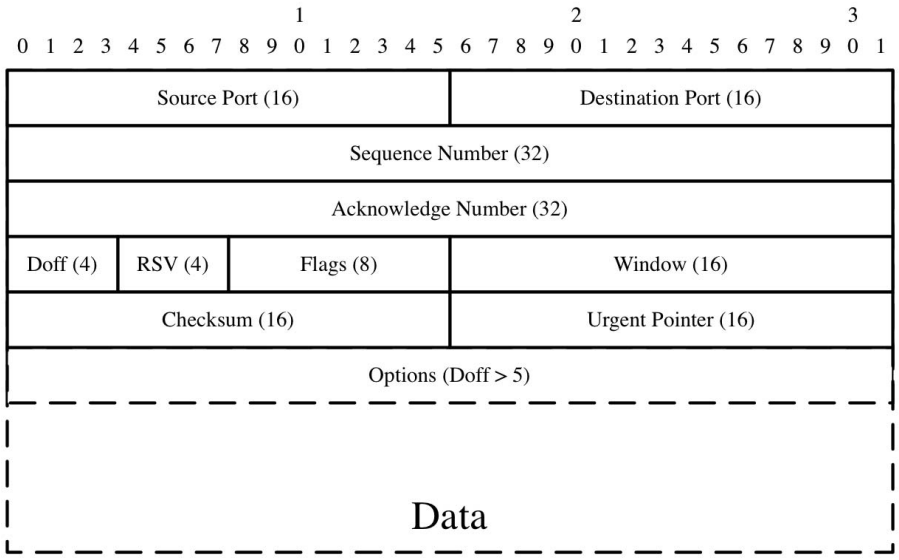

Abbildung 3.9: TCP-Header

Um dem Empfänger mitzuteilen, an welcher Stelle die eigentlichen Nutzdaten beginnen, wird das *Data-Offset Feld (DOff)* verwendet. Da das Options-Feld im Header variabel ist, werden diese Informationen benötigt, um die Größe des Headers weiterzugeben [Fall & Stevens, 2012]. Das DOff-Feld besitzt eine Breite von *4 Bit*. Damit kann eine maximale Headergröße von insgesamt *60 Byte* in 32-Bitwörtern umgesetzt werden [Postel, 09/1981]. Das *4 Bit* große *RSV-Feld* ist für experimentelle Implementierungen reserviert und muss bei normaler Benutzung Null betragen (ebd.). Für die nächsten *Flags-Felder* sind *8 Bit* reserviert. Darunter fallen die folgenden *Flags (ebd.)*:

- CWR – Anzeige, dass das Überlastfenster verringert wurde, da ein ECE-Flag empfangen wurde;
- ECE – ein Router hat eine Netzwerk-Überlast erkannt;
- URG – Urgent Pointer, zur Nutzung eines Unterbrecher-Mechanismus für die sofortige Bearbeitung der beinhaltenden Daten;
- ACK – Bestätigungsbit (siehe oben);
- PSH – Push Funktion, Senden von Daten, die die Applikation ohne Zwischenspeicherung sofort erreichen sollen;
- RST – Reset Connection, um dem anderen Endpunkt einen Fehlerfall mitzuteilen;

- SYN – Synchronisation der Sequenznummer (siehe oben) und Initiierung einer Verbindung;
- FIN – Verbindung abschließen – keine Daten mehr (siehe oben).

Diese Flags werden bei Verwendung mit einer Eins aktiviert. Das Header-Feld *Window* ist für die Flusskontrolle bestimmt. Hier wird die verbleibende Fenstergröße in Byte ausgetauscht. Durch die 16-Bit-Größe des Feldes kann eine maximale Empfangsfenstergröße von *64 KByte* angegeben werden [Fall & Stevens, 2012]. Die *Checksum* enthält die 16-Bit-Prüfsumme, die für die Erkennung von Übertragungsfehlern verwendet wird. Der 16-Bit *Urgent Pointer* indiziert die Daten, die zur sofortigen Bearbeitung an die Applikation geschickt werden sollen, als Offset zur Sequenznummer in Oktetten [Postel, 09/1981]. Es wird nur interpretiert, wenn auch das URG-Bit aktiviert wurde. Das *Options-Feld* besitzt eine variable Größe und kann für verschiedene Dinge wie das Bekanntgeben der MSS, Senden eines Timestamps oder andere experimentelle Optionen[22] verwendet werden [Fall & Stevens, 2012].

Eine wichtige Funktion unter TCP ist der *Nagle-Algorithmus*, um das Senden effizienter zu gestalten [Nagle, 01/1984]. Dieser ist dafür zuständig, die Anzahl der Datensegmente möglichst zu reduzieren und auf ihre MSS hin zu optimieren. Für viele Netzwerke ist dies von Vorteil, weil weniger Segmente übertragen werden und der Overhead des TCP-Headers geringer ist. Der Nagle-Algorithmus puffert Daten, bis die MSS ausgenutzt ist, eine ACK-Bestätigung des vorherigen Datensegments empfangen wird oder eine bestimmte Zeit abgelaufen ist [Seth & Venkatesulu, 2008]. Im Falle von Massendaten kann die Effizienz hierdurch erhöht werden. Bei Benutzung von Anwendungen mit Echtzeitansprüchen, bei denen Daten so schnell wie möglich gesendet werden müssen, erhöht der Nagle-Algorithmus die Verzögerungen unnötig und sollte daher desaktiviert werden.

Die Verwendung des TCP-Protokolls wird über sogenannte Sockets realisiert. Ein Socket ist eine Schnittstelle zu protokollspezifischen Datenstrukturen, welche innerhalb des Betriebssystemkerns oder innerhalb von Betriebssystemmodulen implementiert sind. Der Socket nutzt die darunterliegende Vermittlungsschicht und bildet einen logischen Kommunikationsendpunkt für die Applikation und den Kommunikationsteilnehmer [Leffler et al., 1991] [Seth & Venkatesulu, 2008]. Für die Applikation ist lediglich der Socket sichtbar, jedoch nicht die darunterliegende Datenstruktur. Die Datenstruktur bildet einen Speicher in Form einer verketteten Liste für Datensegmente und Header sowie die Möglichkeit verschiedene Steueroperationen in Form von *System Calls* auszuführen.

3.4 Zusammenfassung der Dienstgüte-Methodiken

Das Internet besteht aus einer Vielzahl heterogener Netzwerke, deren gemeinsame Arbeitsbasis die Bereitstellung von Best-Effort-Diensten ist. Die in Kapitel 3.3 diskutierten Mechanismen dienen der Bereitstellung einer fundierten Grundlage des Best-

[22] Siehe Kapitel 5.1.3

Efforts, d.h. eine faire und gleichberechtigte Behandlung von Datenverkehr nach bestmöglichem Bemühen. Zudem bilden sie die Basis für erweiterte Möglichkeiten, die Dienstgüte für „Spezialdienste“ über die von Best-Effort heben.

Die folgende Auflistung bietet eine Übersicht über die oben angesprochenen Methoden zur Bereitstellung von Best-Effort-Diensten sowie Diensten, die ein NSP über den Ansatz von Best-Effort hinaus anbietet:

- Überdimensionierung eines Netzwerks
- Datenverkehrsformung und -regulierung
- Datenpaket-Scheduling
- Paketklassifizierung
- Differenzierte Dienste
- Integrierte Dienste
- Traffic Engineering

wobei die Überdimensionierung und die Integrierten Dienste vor allem in kleineren Netzwerken vorkommen, da sie schlecht auf große Benutzergruppen skalierbar sind. Bei allen Methoden sind vor allem netzwerkübergreifende Szenarien problematisch, bei denen ein bestimmter Dienstgütestandard durchgängig von Bedeutung ist.

Eine faire und gleichberechtigte Behandlung des Datenverkehrs ohne Nutzung von übergeordneten Diensten bietet zwar für alle das Gleiche, jedoch kommt es an Engpässen in Netzwerken zu einer Datenverkehrsüberlast, die durch Warteschlangen und Datenpaketverlust aufgelöst werden muss. Dies hat zur Folge, dass sich Latenzzeiten erhöhen und der Datendurchsatz sinkt. Speziell für Echtzeitanwendungen stellen diese Faktoren ein Problem dar. Eine Netzwerkevaluation der Strecke zwischen UDE und UKM soll diese Probleme identifizieren.

4 Vernetzung im Rahmen von internationalen Partnerschaften

Bei größer werdender Distanz vergrößern sich Probleme bei der Dienstgüte einer Übertragung. Ein Extremfall ist die Steuerung einer telemedizinischen Apparatur über Landesgrenzen hinweg, die mit einer großen Entfernung einhergeht. Hierbei werden mehrere internationale Netzwerke mit womöglich unterschiedlichen Routing- und Warteschlangen-Regelwerken genutzt. Dabei variieren die Dienstgüte-Leistungsdaten zu verschiedenen Zeiten und Standorten. Routing, Warteschlangen an Netzwerkknoten und damit verbundene Verluste besitzen den größten Einfluss auf zusätzliche Verzögerungen [Singla et al., 2014].

Weitere Probleme können durch Engpässe an Netzwerksegmenten mit unzureichend entwickelter Infrastruktur entstehen [Ford, 2014]. Bei einer telemedizinischen Verbindung zwischen einer Klinik in einem Entwicklungsland und mehreren Spezialisten einer fachbezogenen Klinik in einem höher entwickelten Land sind Probleme vor allem auf den letzten Meilen der Übertragung zu erwarten – also beim Anschluss der Klinik an den lokalen Internet Service Provider (ISP) und die Anbindung des ISPs an den internationalen Backbone. Diese stellen einen Flaschenhals dar und bestimmen damit den maximalen Datendurchsatz sowie Schwankungen bei Verzögerungszeiten. Untersuchungen müssen zeigen, inwieweit eine Vernetzung auf globaler Ebene Einfluss auf die Dienstgüte-Leistungsdaten nimmt.

Eine Zusammenarbeit zwischen der UDE und der UKM besteht seit 1997 und hat seit dieser Zeit zu mehreren gemeinsamen Bildungs- und Forschungsprojekten geführt. Weitere gemeinsame Forschungsvorhaben sind angedacht, für die eine höher entwickelte Vernetzung zwischen den beiden Universitäten angestrebt wird. Mit einer Länge von ca. 10.000 km führt eine Verbindung zwischen den beiden Standorten über mehrere Staatsgebiete und globale Basisnetzwerke (Tier-1 und Tier-2) [Winther, 2006] hinweg. Eine Herausforderung ist die Stabilisierung sowie Minimierung der Netzwerklatenz, wie sie für spezialisierte Anwendungen im Bereich der Telemedizin nötig sind.

Das Ziel dieses Kapitels ist die Untersuchung der räumlichen Strukturen des durch die Verbindung verwendeten Netzwerks und der verkehrsdominierenden Pfade zwischen den beiden Endpunkten. Um sich ein Bild von der derzeitigen Netzwerksituation zwischen UDE und UKM zu machen, werden mehrere Messungen durchgeführt. Diese Messungen geben Aufschluss über mögliche Schwachstellen einer Verbindung für telemedizinische Zwecke über diese weite Distanz.

Zu den Messungen zwischen den beiden Endpunkten gehören:

- Routenbestimmung
- Latenzmessungen
- Datenübertragungsratenmessungen

Das Kapitel beginnt mit einer Beschreibung der unterschiedlichen Netzwerke und der möglichen Verbindungen zwischen den beiden Endpunkten. Die verschiedenen Netzwerke werden evaluiert und es wird eine mögliche Problemverbesserung mithilfe multipler Pfade diskutiert. Die verschiedenen Messungen und die dazu verwendeten Werkzeuge werden erläutert.

4.1 (Inter-)Nationale Netzwerke und ihre Infrastruktur

Zwischen den beiden Universitäten UDE und UKM sind unterschiedliche Netzwerkverbindungen möglich. Sie werden durch mehrere Internetknoten verknüpft, die die Hauptknoten der interkontinentalen Vernetzung bilden. Jede der beiden Universitäten besitzt einen Breitbandanschluss an das Internet bzw. an ein lokales nationales Netzwerk. In Deutschland ist die UDE an das deutsche Forschungsnetzwerk (DFN) X-WIN angebunden. Die UKM besitzt dagegen mehrere mehr oder wenige konventionelle Verbindungen, die je nach Anschlusskonzept genutzt werden. Eine Anbindung der UKM an das Malaysische Forschungsnetzwerk MyREN (Malaysian Research and Education Network) [Ministry of Higher Education, 2017] besteht derzeit nicht. Alle Verbindungen werden daher jenseits des X-WIN-Netzwerks über konventionelle Internetknotenpunkte geroutet.

4.1.1 Internetanbindung der UDE

Die UDE besitzt einen Breitbandanschluss an das deutsche Forschungsnetz (DFN) [DFN, 2017]. Das DFN wird von einem Verein betrieben, der von wissenschaftlichen Organisationen und Institutionen getragen wird [DFN, 2017]. Das Netzwerk verbindet Institutionen in Wissenschaft und Bildung miteinander, stellt netzbezogene Dienstleistungen zur Verfügung und ermöglicht Forschungsinstitutionen die Koppelung mit anderen Netzwerken wie dem Internet. Der Verein wurde 1984 mit Unterstützung durch das Bundesministerium für Forschung und Technologie (BMFT) mit dem Ziel gegründet, ein deutsches Forschungsnetzwerk zu entwickeln und zu fördern [Trueöl, 1984]. Mitglieder waren anfangs Hochschulen und außeruniversitäre Forschungseinrichtungen (ebd.).

Nach mehrfacher Aufrüstung, Umrüstung und Umbenennung des Netzwerks wurde im Jahre 2006 das heutige X-WiN-Netzwerk eingerichtet [DFN, 2006]. Es ist ein glasfasergestütztes Netzwerk für ca. 700 Forschungsinstitutionen und Hochschulen in Deutschland mit 65 sogenannten *Core-Netzwerk*-Standorten, die den Backbone des Netzwerks bilden (ebd.). Es besitzt eine Einbettung über mehrere direkte Verbindungen des Glasfasernetzes in das Europäische Forschungsnetzwerk GÉANT [GÉANT, 2016] und andere angrenzende nationale Forschungsnetzwerke.

Das Core-Netzwerk des X-WiN setzt sich aus ca. 10.000 km Glasfaser zusammen [DFN, 2017]. Das Netzwerk ist redundant aufgebaut, sodass ein Knotenpunkt mindesten über zwei unabhängige (redundante) Wege erreichbar ist (ebd.). Das Netzwerk bietet Teilnehmern für jeden Weg eine Datenübertragungsrate von *10 Gigabit/s* [DFN, 2016]. Das sogenannte Super-Core-Netzwerk ist eine redundante Kreisverbindung zwischen den Knotenpunkten Hannover, Frankfurt, Erlangen und Berlin mit einer Datenübertragungsrate von *200 Gigabit/s* pro Doppelanbindung. Von diesen Netzwerkknoten aus sind die Verbindungen in die angrenzenden Netzwerke geschaltet. Frankfurt bildet den wichtigsten Verbindungspunkt zum europäischen Forschungsnetzwerk GÉANT und dem DE-CIX [DE-CIX, 2017] Knoten, der für Deutschland die Hauptverbindung ins Internet darstellt. Abbildung 4.1 zeigt die Verbindungen des Core- und des Super-Core-Netzwerks.

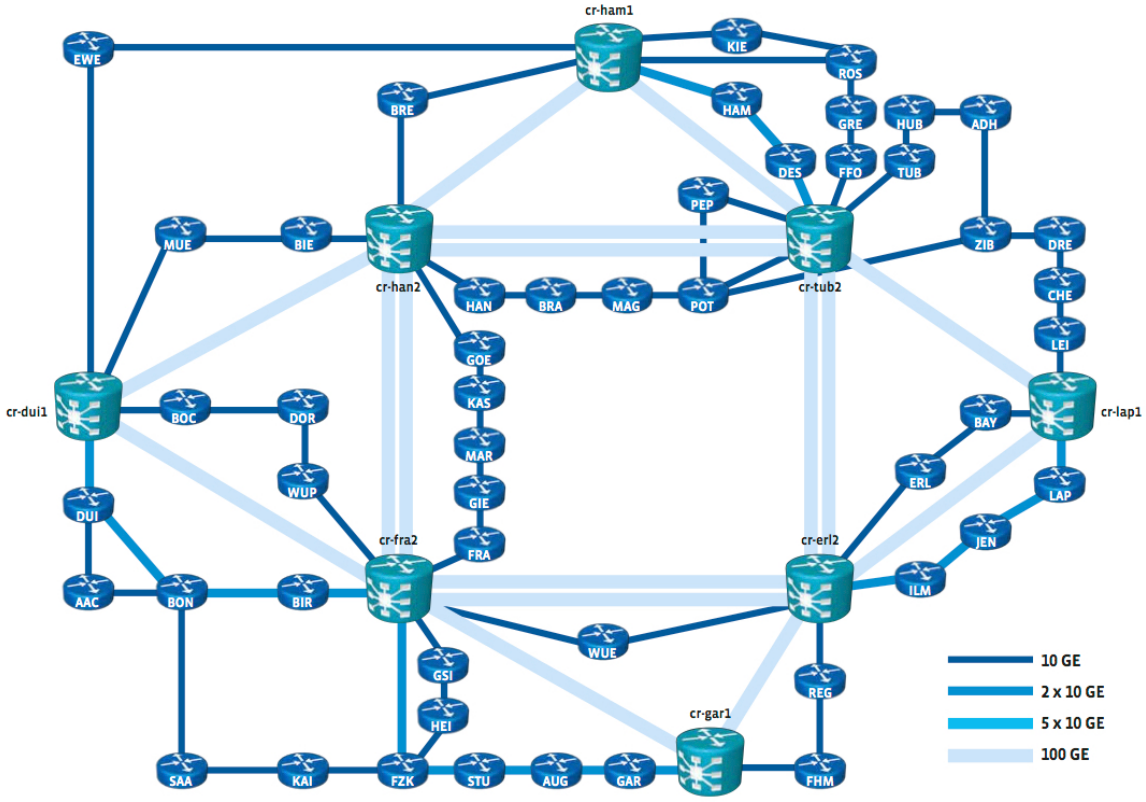

Abbildung 4.1: X-WiN Topologie und Core-Netzwerk[DFN, 2016]

Die Dienstgüte wird beim X-WiN durch selbstauferlegte Regeln und Ansprüche angegeben. Das X-WiN gilt als eines der stabilsten und leistungsstärksten Forschungsnetzwerke der Welt [DFN, 2016]. Neben den oben genannten Datenübertragungsraten liegen weitere Parameter des selbst auferlegten Dienstgüte-Ziels in den folgenden Bereichen [DFN, 2017]:

- das OWD von Paketen soll kleiner als 1 ms pro 100 km Leitungslänge betragen;
- Schwankungen der Paketlaufzeit sollen weniger als 0,1 ms pro 100 km Leitungslänge betragen;
- eine Paketverlustrate von Null soll erreicht werden.

Die Universität Duisburg-Essen bildet einen Knoten im Core-Netzwerk mit einem direkten Anschluss an das Super-Core-Netzwerk. Eine Anbindung an das Internet erfolgt über Hannover nach Frankfurt zum DE-CIX Internetknoten. Eine aktuelle Latenzzeit zwischen der UDE und Frankfurt beträgt nach [DFN, 2016] in etwa *6 ms*.[23]

4.1.2 Internetanbindungen der UKM

Die UKM verfügt derzeit nicht über einen Zugang zum Malaysischen Forschungsnetzwerk MyREN. Daher wird jeglicher Datentransfer zwischen UDE und UKM ab dem Frankfurter Netzwerkknoten des X-WiN über den DE-CIX Kotenpunkt in das Internet geroutet und umgekehrt. Die UKM verfügt über mehrere Zugänge zum Internet über konventionelle Internet Service Provider (ISP). Im Rahmen der Kooperation sind derzeit die folgenden Zugänge verfügbar:

1. UKM Standard-Netz
2. Polycom-Netzwerkverbindung
3. 3G-Mobilfunkverbindung

Über das UKM Standard-Netz erfolgt die Anbindung der gesamten UKM an den ortsansässigen ISP *Maxis* [Maxis, 2017]. Diese Verbindung wird standardmäßig als Internetverbindung für das gesamte Netzwerk der ingenieurwissenschaftlichen Fakultät der UKM verwendet. Alle Geräte der Fakultät, die intern eine Verbindung an das Internet benötigen, nutzen diese Leitung. Die Verbindung ist starken Einschränkungen unterworfen, da die Firewall-Regelungen der Fakultät sehr restriktiv sind. Rechner, die von außen angesprochen werden sollen, benötigen eine explizite Freischaltung für IP-Adressen und Portnummern. Freischaltungen erfolgen über einen relativ aufwendigen, bürokratischen und von hierarchischen Strukturen geprägten Prozess. Zudem sind die Router zum Teil überlastet oder durch die extremen Umwelteinflüsse in Malaysia, wie Temperaturen (~27-35°C) und Luftfeuchtigkeit (~80 % - 90 %), beeinträchtigt.

Die zweite Anbindung der UKM erfolgt ebenfalls über eine Verbindung des Standard ISPs der Fakultät. Diese besitzt allerdings ein priorisiertes Routing innerhalb des UKM-Netzes. Diese Verbindung war ursprünglich für die Nutzung einer *Polycom*-Konferenzanlage [Polycom, 2017] gedacht, um eine verbesserte Dienstgüte in Hinblick auf Echtzeitübertragungen für Video- und Audiokonferenzen zu ermöglichen. Für diese Verbindung sind die Firewall-Regeln weniger restriktiv, da für *Polycom*-Anlagen unterschiedlichste Ports benötigt werden.

Die dritte Verbindung nutzt ein ortsansässiges 3G-Netzwerk, das durch den Provider *uMobile* [uMobile, 2017] über Universal Mobil Telecommunications System (UMTS) bzw. High Speed Downlink Packet Access (HSDPA) betrieben wird. Die Verbindung wird mithilfe einer Prepaid-Karte und eines 3G-Routers aufgebaut, an den der verwendete Testrechner angeschlossen ist. In der Regel ist die Anbindung stabil, unterliegt aber ebenfalls dem Einfluss der extremen Witterungslage in Malaysia. Zugänge

[23] Eine „Weathermap“ mit einer Momentaufnahme des Status des X-WiN und seinen Dienstgüte-Kennzahlen befindet sich im Anhang dieser Arbeit.

von außen auf einen Rechner sind nicht direkt möglich, da der 3G-Provider keine Serveraktivitäten über das 3G-Netz zulässt und keine offenen Ports bereitstellt.

4.1.3 Internationale Netzwerkinfrastruktur durch Routenbestimmung zwischen UDE und UKM

Der Datentransfer zwischen UDE und UKM wird ab dem Frankfurter Netzwerkknoten des X-WiN über den DE-CIX Kotenpunkt in das Internet geroutet und umgekehrt. Von dort aus verläuft er über mehrere mögliche Wege, die durch die angeschlossenen Betreiber (Tier-1 und Tier-2) bestimmt werden. Um einen Einblick in die nachfolgenden Knotenpunkte und ihre Verzögerungszeiten zu bekommen, wird eine Routenbestimmung (*Traceroute*) mithilfe des ICMP-Protokolls durchgeführt. Hierdurch werden die einzelnen IP-Teilstrecken erfasst, die Laufzeiten der Segmente für jede Teilstrecke bestimmt und damit die Länge des Pfades deutlich gemacht.

Das ICMP-Protokoll wird zum Austausch von Informations- und Fehlermeldungen von Netzwerken verwendet [Postel, 9/1981]. Beim Generieren einer Routenbestimmung mit ICMP wird bei jedem einzelnen Knotenpunkt eine Antwortnachricht produziert: Eine Anfrage (Ping) wird an das Netzwerkgerät gesendet und es wird eine Echo-Antwort (Pong) erzeugt, die wieder an den Quellrechner zurückgesendet wird. Am Quellrechner wird die Zeit gemessen, die von der Anfrage bis zur eintreffenden Antwort benötigt wird. Das ursprüngliche Programm *Ping*, das diese Technik als eines der Ersten nutzte, wurde 1983 entwickelt, um eine Netzwerkkonnektivität diagnostizieren zu können [Muuss, o.D.].

Für ein Abbild der Teilstrecken werden mithilfe des Werkzeugs *mtr* [MTR, 2013] über einen Zeitraum von *100 s*, jeweils einmal pro Sekunde, die Verzögerungszeiten zu den einzelnen Knotenpunkten mithilfe von ICMP gemessen. Eine Statistik wird generiert, die Aufschluss gibt über:

- die Paketverlustrate
- die durchschnittliche, beste und schlechteste Umlaufzeit (RTT)
- die Standardabweichung der gemessenen Werte

Die Paketverlustrate wird in Prozent angegeben. Die RTT ist die Zeit in Millisekunden, die ein Paket benötigt, um von der Quelle zu einem Ziel hin und wieder zurück zu gelangen. Die Standardabweichung wird gleichfalls in Millisekunden angegeben.

Als Zieladresse werden für die Messungen folgende IP-Adressen verwendet:

UKM Verbindung	IP-Adresse
UKM Standard Netz	210.187.26.2
Polycom-Verbindung	210.187.26.113
3G-Mobilfunkverbindung	123.136.107.155

Tabelle 4.1: IP-Adressen der UKM Internetverbindungen

Aufgrund der oben genannten Netzwerk-Limitierungen innerhalb des UKM-Netzes bzw. 3G-Netzes ist es je nach Verbindung nicht möglich, das Ziel bis zum Ende zu er-

reichen. Es können aber Teilstrecken zu einem großen Teil beschrieben werden. Für die IP-Adressen werden aufgrund netzwerkbedingter Restriktionen nicht für jede Teilstrecke Namensauflösungen durchgeführt.

Angaben der Laufzeiten sind davon abhängig, inwiefern die Netzwerkgeräte auf ICMP-Nachrichten antworten. So können Antwortzeiten der ICMP-Pakete höher ausfallen, als dies für reguläre Datensegmente der Fall ist. Ausgegebene Laufzeiten können bereits dem gesamten Pfad entsprechen, da die durchlaufenen Teilstrecken Router eines MPLS Netzwerks sind.[24] In diesem Fall werden die Router im Rahmen eines TE von zentraler Stelle verwaltet und geben keine eigenständigen Antworten auf ICMP-Nachrichten. Andere Netzwerkgeräte wiederum geben je nach Konfigurationsart keine Antwort auf ICMP-Nachrichten. Dies führt zu einer Verlustrate von 100 %, und es kann keine Umlaufzeit gemessen werden.

Die folgenden Messungen stellen eine Momentaufnahme (Mai, 2016) der Route dar. Die Route wird durch die Routing-Strategie des NSP bestimmt. Sie kann auf unterschiedlichen Werten hinsichtlich der Datenübertragungsrate, der Paketverlustrate, der Latenz oder einer voreingestellten Weiterleitungsstrategie bzw. Gewichtung durch den NSP basieren [Savage et al., 1999]. Voraussichtlich existiert ein primärer Routingpfad durch die einzelnen AS. Kleine Änderungen innerhalb der AS können sich abhängig von der Tageszeit oder dem Netzwerkverkehr ergeben. Größere Änderungen, d.h. dass die verwendeten AS variieren, sind eher auf längere Sicht zu erwarten, z.B. bei einer Änderung in den Routing-Regeln des NSP.

4.1.3.1 Routenbestimmung zum UKM Standard Netz

Tabelle 4.2 zeigt eine Routenbestimmung vom Quellrechner der UDE bis zum Zielrechner an der UKM (durchgeführt am 02.03.2016, 11-16 Uhr).

Nr.	Name	IP	Loss %	Avg [ms]	Best [ms]	Worst [ms]	SDev [ms]
1.	cat35gbb.uni-duisburg.de	134.91.90.1	0.0%	0.5	0.5	1.1	0.0
2.	cat29gba.uni-duisburg.de	134.91.254.26	0.0%	0.5	0.5	3.7	0.3
3.	cat35gleb.uni-duisburg.de	134.91.254.34	0.0%	0.6	0.5	4.2	0.5
4.		134.91.254.226	0.0%	2.7	0.4	10.7	3.2
5.	bb-sh0.netz.uni-duisburg-essen.de	132.252.0.45	0.0%	3.3	0.7	11.5	3.4
6.		132.252.254.94	0.0%	0.9	0.8	1.3	0.0
7.	cr-dui1-te0-0-0-2.x-win.dfn.de	188.1.232.109	0.0%	1.3	1.1	2.5	0.1
8.	cr-han2-hundredgige0-9-0-5.x-win.dfn.de	188.1.144.190	0.0%	5.9	5.6	8.2	0.2
9.		80.156.160.137	0.0%	9.7	5.7	58.9	11.6
10.		217.239.47.218	0.0%	27.1	24.0	30.4	1.3
11.		87.190.233.202	0.0%	281.2	277.3	336.8	11.0
12.		???	100.%	0.0	0.0	0.0	0.0
13.		202.188.129.114	2.0%	282.3	275.1	285.9	2.8
14.		???	100.%	0.0	0.0	0.0	0.0

Tabelle 4.2: Traceroute der UKM Standard Netzwerkverbindung

Die 134er und 132er IP-Adressen (Teilstrecke 1-6) sind Teil des UDE-Netzwerkes. Nummer 7 stellt das erste Netzwerkgerät im X-WiN-Forschungsnetzwerk mit dem Standort Duisburg dar. Darauf folgt Hannover (Teilstrecke 8). Die nächsten Abschnitte

[24] Vgl. Kapitel 3.3.5

sind nicht genau lokalisierbar (Teilstrecke 9-11). Da sie durch die Telekom von Deutschland aus verwaltet werden, sind IP-Adresslokatoren[25] nicht in der Lage, eine korrekte Standortlokalisierung durchzuführen. Teilstrecke 9 besitzt eine durchschnittliche Laufzeit von *9,7 ms* und kann daher bereits außerhalb Deutschlands liegen. Teilstrecke 11 besitzt eine große durchschnittliche Latenzzeit von *281,2 m*s. Die Lokalisierung der Teilstrecke 13 verweist auf einen Standort in Malaysia.

Die anschwellenden Latenzzeiten bei Teilstrecke 8, 9 und 10 zeigen die größer werdende Distanz der Messstrecke. Die höchsten Latenzzeiten werden in Teilstrecke 11 und 13 gemessen. Diese können als repräsentativ für die ICMP-Gesamtlatenzzeiten des Weges angenommen werden.[26]

4.1.3.2 Routenbestimmung zur UKM Polycom Verbindung

Die nachstehende Tabelle 4.3 zeigt die Netzwerkstrecke der UDE zur UKM über die Polycom-Verbindung (durchgeführt am 02.03.2016, 11-16 Uhr). Wie aus der Tabelle ersichtlich wird, sind im Gegensatz zu Tabelle 4.2 die Antwortzeiten der letzten Teilstrecke erkennbar. Dies liegt daran, dass über die Polycom-Verbindung die benötigten Ports für ICMP-Nachrichten geöffnet sind. Der Zielrechner ist in der Lage, auf ICMP-Anfragen zu antworten.

Nr.	Name	IP	Loss %	Avg [ms]	Best [ms]	Worst [ms]	SDev [ms]
1.	cat35gbb.uni-duisburg.de	134.91.90.1	1.0%	0.8	0.5	28.7	2.8
2.	cat29gba.uni-duisburg.de	134.91.254.26	0.0%	0.5	0.5	1.0	0.0
3.	cat35gleb.uni-duisburg.de	134.91.254.34	0.0%	0.6	0.5	2.6	0.2
4.		134.91.254.226	0.0%	3.2	0.4	11.2	3.4
5.	bb-sh0.netz.uni-duisburg-essen.de	132.252.0.45)	0.0%	3.7	0.8	11.6	3.5
6.		132.252.254.94	0.0%	1.2	0.8	6.0	0.6
7.	cr-dui1-te0-0-0-2.x-win.dfn.de	188.1.232.109)	0.0%	1.6	1.2	9.0	0.8
8.	cr-han2-hundredgige0-9-0-5.x-win.dfn.de	188.1.144.190)	0.0%	6.4	5.6	11.0	0.7
9.		80.156.160.137	0.0%	11.2	5.7	84.2	14.3
10.		217.239.53.82	0.0%	26.1	22.4	35.7	2.4
11.		87.190.233.202	0.0%	278.8	275.6	337.6	8.1
12.		???	100.%	0.0	0.0	0.0	0.0
13.		202.188.129.114	5.0%	281.1	273.8	284.8	2.3
14.		210.187.26.113	7.0%	292.9	286.1	310.0	3.2

Tabelle 4.3: Traceroute der UKM Polycom Netzwerkverbindung

Die Teilstrecken der Polycom-Verbindung gleichen den Routenabschnitten der UKM-Standard-Anbindung bis auf wenige Teilabschnitte. Eine Abweichung ist bei Teilstrecke 10 auszumachen. Hier unterscheiden sich aber lediglich die letzten zwei Oktette, was andeutet, dass sich der Abschnitt im selben Providernetzwerk befindet. Die nachfolgenden Teilstrecken sind wiederum identisch, mit dem Unterschied, dass Antworten des Zielrechners innerhalb der UKM möglich sind.

Die Latenzzeiten der UKM Standard-Verbindung und der Polycom-Verbindung sind nahezu identisch. Der letzte Teilabschnitt (14) innerhalb der UKM befindet sich ca. *12 ms* von dem vorherigen Standort in Malaysia entfernt.

[25] wie z.B. http://www.utrace.de
[26] Eine Statistik der gemessenen Umlaufzeiten befindet sich im Anhang

4.1.3.3 Routenbestimmung zur UKM 3G-Verbindung

Die Route der 3G-Anbindung der UKM unterscheidet sich stark von den anderen beiden Anbindungen. Die nachfolgende Tabelle 4.4 zeigt die Routenbestimmung von der UDE bis zur UKM über die 3G-Anbindung (durchgeführt am 03.03.2016, 11-16 Uhr).

Darin beschreiben die ersten Teilstrecken 1-7 erneut die Route durch das Netzwerk der UDE und den ersten X-WiN-Knoten in Duisburg. Teilstrecke 8 ist hier der X-WiN-Hauptknoten in Frankfurt, gefolgt durch einen weiteren Frankfurter Knoten. Teilstrecke 10 ist nicht nachvollziehbar, dafür können die weiteren Knoten 11 und 12 unter Verwendung eines IP-Adresslokators als Netzwerkknoten identifiziert werden, die in Großbritannien gemeldet sind. Eine wahrscheinliche Anbindung von London aus ist hier Peking.

Nr.	Name	IP	Loss %	Avg [ms]	Best [ms]	Worst [ms]	SDev [ms]
1.	cat35gbb.uni-duisburg.de	134.91.90.1	1.0%	0.5	0.5	1.7	0.1
2.	cat29gba.uni-duisburg.de	134.91.254.26	0.0%	0.5	0.5	3.0	0.2
3.	cat35gleb.uni-duisburg.de	134.91.254.34	0.0%	0.7	0.5	6.3	0.6
4.		134.91.254.226	1.0%	3.0	0.4	10.8	3.3
5.	bb-sh0.netz.uni-duisburg-essen.de	132.252.0.45	0.0%	2.9	0.7	11.3	3.2
6.		132.252.254.94	0.0%	0.9	0.8	2.2	0.1
7.	cr-dui1-te0-0-0-2.x-win.dfn.de	188.1.232.109	0.0%	1.4	1.1	2.4	0.1
8.	cr-fra2-hundredgige0-9-0-3.x-win.dfn.de	188.1.144.178	0.0%	8.4	8.1	9.0	0.0
9.	100ge7-2.core1.fra1.he.net	80.81.192.172	0.0%	21.6	8.0	70.4	15.0
10.		???	100.%	0.0	0.0	0.0	0.0
11.	xe-8-0-1.0.pjr04.ldn001.flagtel.com	85.95.25.221	0.0%	20.8	20.2	26.3	0.9
12.		85.95.27.98	0.0%	215.4	213.8	267.5	7.8
13.		80.77.1.98	0.0%	290.6	289.2	322.5	5.3
14.	d1-114-224-143-118-on-nets.com	118.143.224.114	0.0%	284.2	283.7	297.5	1.5
15.	global.hgc.com.hk	218.189.23.162	0.0%	299.3	296.8	338.9	6.5
16.		123.136.99.42	0.0%	290.1	289.9	293.7	0.3
17.		123.136.101.156	0.0%	298.9	298.6	300.4	0.1
18.		???	100.%	0.0	0.0	0.0	0.0

Tabelle 4.4: Traceroute der UKM 3G Netzwerkverbindung

Die weiteren Teilstrecken 13-15 befinden sich bereits in Hong Kong, wo sich einer der wichtigsten Internetknoten des asiatischen Raumes befindet [TEIN3 (Hg.), 2013]. Dies lässt darauf schließen, dass die Teilstrecke 12 bereits in Peking lokalisiert ist, worauf auch die große durchschnittliche Latenzzeit von *215,4 ms* hinweist. Von Hong-Kong (15) geht es dann über die Teilstrecke 16 nach Malaysia. Der letzte Teilabschnitt 18 ist wiederum nicht in der Lage, auf ICMP-Nachrichten zu antworten, da das 3G-Netzwerk dieses Protokoll nicht erlaubt. Die durchschnittliche Gesamtlatenz ist etwas höher im Vergleich zu den anderen beiden Verbindungen.

4.1.3.4 Karte der Teilstrecken der bekannten interkontinentalen Route UDE – UKM

Für eine geografische Routendarstellung können die identifizierten Netzwerkknoten auf einer Weltkarte eingetragen werden. Die nachfolgende Abbildung 4.2 zeigt die verschiedenen Routen der oben genannten Verbindungen zwischen UDE und UKM:

- die Route in rot ist die Verbindung der UKM Standard- und der UKM Polycom-Anbindung
- die blaue Route beschreibt die Verbindung über die 3G-Anbindung der UKM

In der nachfolgenden Abbildung 4.2 sind nur die Knotenpunkte eingezeichnet, die tatsächlich identifiziert werden konnten.

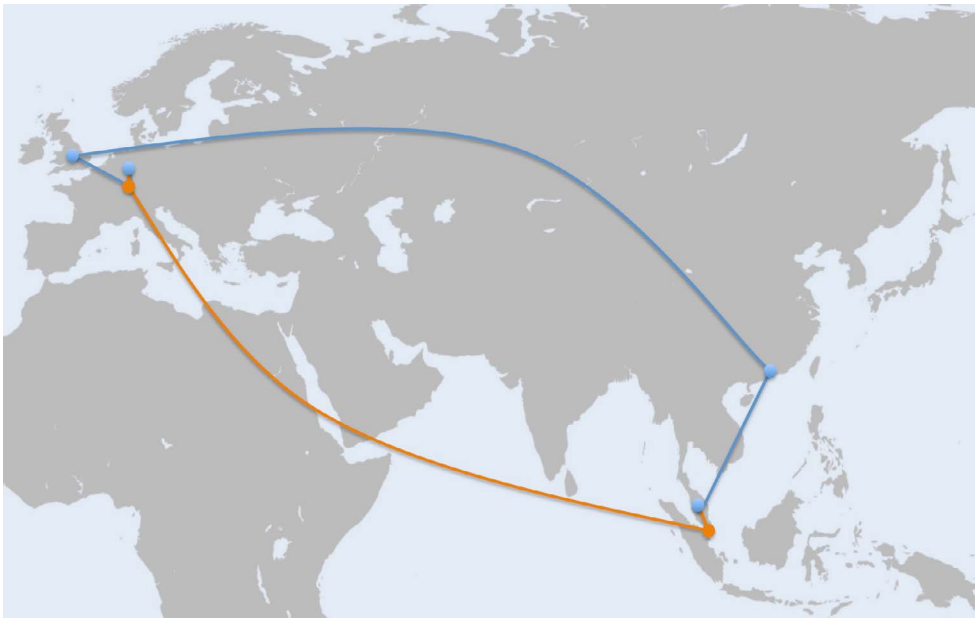

Abbildung 4.2: Hauptrouten der Verbindung UDE - UKM

4.2 Methoden und Werkzeuge für die Messung der Verbindung UDE – UKM

Zwischen UDE und UKM soll eine stabile Verbindung aufgebaut werden. Welche Metriken diese Verbindung besitzt, muss mithilfe von Messungen bestimmt werden. Wie in Kapitel 2.4 beschrieben, werden für telemedizinische Anwendungen bestimmte Dienstgüten benötigt. Die dafür notwendigen Parameter wurden dort bereits diskutiert. Die bereitgestellte Dienstgüte der Gesamtstrecke zwischen UDE und UKM soll im Rahmen dieses Kapitels bestimmt werden. Messungen der Netzwerkstrecken sollen die folgenden Dienstgüte-Parameter ermitteln:

- Umlaufzeiten (RTT) bzw. Latenzzeiten der Strecke
- Schwankungen (Varianzen) der Latenzzeiten (Jitter)
- Paketverluste / Segmentverluste
- Datenübertragungsrate

Diese Parameter sollen in Abhängigkeit von der Tageszeit und der verschiedenen Anbindungen, die an der UKM möglich sind, untersucht werden. Für die Messungen werden sowohl Protokolle der anwendungsorientierten Schichten als auch der Transportschicht eingesetzt. Zur Vereinfachung werden im weiteren Verlauf dieses Kapitels allgemein die Begriffe Datenpaket bzw. Paket verwendet.

Für Messungen an beiden Standorten wird jeweils ein Rechner verwendet, der mit dem Internet verbunden ist. An der UDE ist dies ein Rechner, der über eine Anbindung an das Universitätsnetz verfügt und damit Zugriff auf das Forschungsnetz besitzt. Der Rechner an der UKM wird über die entsprechende Anbindung des jeweiligen ISPs an das Internet angeschlossen. Die beiden Rechner und genutzten Netzwerkgeräte besitzen jeweils folgende Eigenschaften:

UDE Messrechner, Duisburg, Deutschland	Marke und Typ	Dell OptiPlex GX620
	Prozessor	Intel Pentium 4 CPU 2,8 GHz *2
	Arbeitsspeicher	2 GiB
	Betriebssystem	Linux Mint 17.1 32-bit
	Kernel	3.13.0-27-generic #64
	Netzwerkkarte	Broadcom Corporation NetXtreme BCM5751
UKM Messrechner, Kuala Lumpur, Malaysia	Marke und Typ	Dell OptiPlex 7010
	Prozessor	Intel Core i7-3770 CPU @ 3,4 GHz *4
	Arbeitsspeicher	4 GiB
	Betriebssystem	Ubuntu 14.04 32-bit
	Kernel	Linux 3.18.34-mptcp.v0.90 i686
	Netzwerkkarte 1	Intel Corporation 82579LM
	Netzwerkkarte 2	VIA Technologies, Inc. VT6105/VT6106S

Tabelle 4.5: Technische Daten der Messrechner

4.2.1 Durchführung der Messungen

Bei einer Umlaufzeitmessung wird ein Paket gesendet, welches auf Empfangsseite eine Echo-Nachricht erzeugt, die wiederum an den anfänglichen Sender zurückgeschickt wird. Dies ist vergleichbar mit dem oben genannten Verfahren zur Routenbestimmung. Eine Umlaufzeitmessung kann mithilfe eines aktiven Messverfahrens durchgeführt werden, d.h. mithilfe von zusätzlichem Netzwerkverkehr wird eine Messung durchgeführt. Das Verfahren wird nach [Wang et al., 2004] durch die folgende Funktion ausgedrückt:

$$RTT(i) = min\ \{R_T(p(i)) - S_T(p(i)),\ t_0\}, \quad t_0 > 0, \tag{4.1}$$

wobei S_T den Zeitstempel markiert, der beim Absenden eines Pakets p mit der Nummer i beim Sender erzeugt wird. R_T definiert den Zeitstempel, der beim Eintreffen des zurückgesendeten Pakets auf der ursprünglichen Senderseite erzeugt wird. Die Zeit t_0 ist

eine frei definierbare Zeitspanne, die zur Bestimmung eines Paketverlusts verwendet wird. Liegt die Rücksendung der Echo-Nachricht über t_0, so wird diese als Paketverlust eingestuft.

Mit diesem Messverfahren lässt sich die Umlaufzeit bestimmen. Eine genaue OWD kann mit diesem Verfahren dagegen nur annäherungsweise ermittelt werden [Wang et al., 2004]. Dies ist dem Umstand geschuldet, dass ein zurückkommendes Paket einen anderen Pfad verwenden kann und somit die beiden Wege vom Sender weg und wieder zurück asymmetrisch sein können (ebd.).

Der Vorteil bei der Umlaufzeitmessung liegt in der Selbstsynchronisation der Zeitstempel, da beide Zeitmessungen auf demselben Endgerät durchgeführt werden (ebd.). Ein Verfahren für die Messung des OWD ist weitaus aufwendiger, da eine Messinfrastruktur innerhalb des Netzwerks benötigt wird, bei der mehrere synchronisierte Systeme an unterschiedlichen Standorten eine Messung durchführen [Wang et al., 2004]. Vorzugsweise wird daher eine vereinfachte Berechnung des OWD wird mithilfe von Formel 4.2 unter der Annahme von symmetrischen Pfaden berechnet [Postel, 09/1981]:

$$OWD(i) = \frac{RTT(i)}{2} \tag{4.2}$$

Die Umlaufzeitmessungen werden über mehrere Tage für die verschiedenen Internet-Verbindungen der UKM durchgeführt. Sie werden mithilfe einer Hypertext-Transfer-Protokoll-Anfrage (HTTP) von einem der Rechner an den anderen ermittelt. Diese Anfrage wird einmal pro Sekunde durchgeführt. Der Datenverkehr wird mithilfe von *Wireshark* [Wireshark, 2017] erfasst und für die weitere Analyse gespeichert. Die Verzögerungsvarianz wird durch Formel 2.3 berechnet.[27]

Das HTTP-Protokoll ist ein Protokoll der anwendungsorientierten Schichten zur Übertragung von Daten im Internet [W3C, 1991]. HTTP wird in der Regel für das Senden von Daten beim Zugriff auf Webseiten verwendet (ebd.). Hierüber ist im Falle der Messungen ein Datenaustausch sehr einfach zu realisieren und die verschiedenen Parameter sind durch das Senden von vorbestimmten Paketen gut messbar.

Für das Senden der HTTP-Anfrage wird auf dem Client unter der Linux-Kommandozeile das Programm *wget* [wget, 2017] verwendet. Auf dem Server läuft ein Apache-Daemon [Apache, 2016], der die Anfrage beantwortet und die gewünschten Daten verschickt. Der Apache-Daemon läuft in der Standard-Konfiguration mit einer minimalen HTML-Seite mit einer Größe von *176 Bytes*. Da der UKM-Rechner von außen nur bedingt erreichbar ist, wird dieser Rechner als Client verwendet und der Rechner an der UDE als Server. Die Messungen werden also von der UKM aus vorgenommen.

Eine einzelne HTTP-Anfrage produziert den in Tabelle 4.6 dargestellten Datenverkehr im Falle einer fehlerfreien Verbindung – ohne erneutes Übertragen. Die „Protokoll"-Spalte zeigt das verwendete Protokoll des Datenpakets, die Spalte „Type" zeigt

[27] Vgl. Kapitel 2.4.2

den Nachrichtentyp hinsichtlich des Protokolls und die letzte Spalte die Länge des Pakets in Bytes. Teil der HTTP-Anfrage ist zunächst der Handshake des Transport Control Protokolls (TCP)[28] (Zeile 1-3). Darauf folgen die eigentliche HTTP-Anfrage (Zeile 4) und eine TCP-Bestätigung, dass das Datenpaket angekommen ist (Zeile 5). Die HTTP-Anfrage wird durch das angeforderte Datenpaket beantwortet (Zeile 6). Die angekommene Sendung wird wiederum durch TCP bestätigt und zugleich der Befehl für das Beenden der Verbindung gesendet (Zeile 7), was wiederum von beiden Seiten bestätigt wird (Zeile 8-9). Die Verbindung wird beendet. Insgesamt erzeugt eine HTTP-Anfrage neun zu sendende Pakete mit einer Komplettlänge von *1140 Bytes*, was einem Durchschnitt von *126,7 Bytes pro Paket* entspricht.

	Protokoll	**Type**	**Length [Byte]**
1	TCP	SYN	66
2	TCP	SYN/ACK	66
3	TCP	ACK	54
4	HTTP	GET	190
5	TCP	ACK	60
6	HTTP	200 OK	536
7	TCP	FIN/ACK	54
8	TCP	FIN/ACK	60
9	TCP	ACK	54

Tabelle 4.6: Erzeugter Datenverkehr durch HTTP Anfrage

Abbildung 4.3 zeigt den Ablauf der HTTP-Anfrage als Sequenzdiagramm. Die zeitlichen Abläufe sind fortschreitend von oben nach unten eingetragen und mit durchschnittlichen Prozesszeiten für die Verarbeitung der Pakete in kursiv versehen.

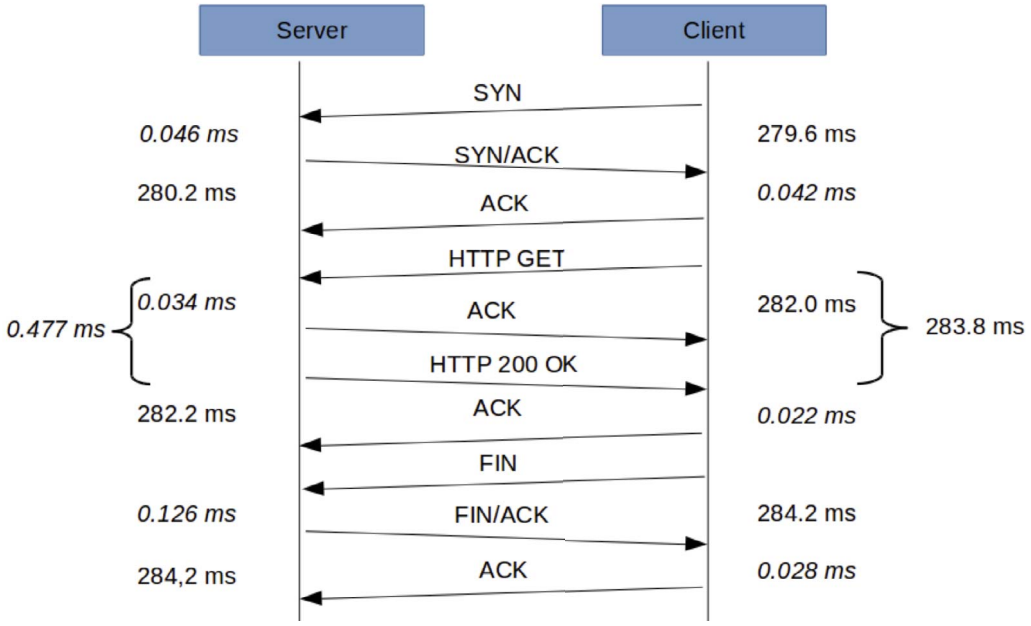

Abbildung 4.3: Zeitlicher Ablauf eines Messvorgangs

[28] Vgl. Kapitel 3.3.6.3

Abgesehen von der tatsächlichen Apache-Datenübergabe sind die Prozesszeiten der Versuchsrechner für die HTTP-Anfrage relativ stabil und statisch. Sie liegen im Mikrosekundenbereich bei einem Durchschnitt von *50 µs*. Beispielhaft sind außerdem Umlaufzeiten zu den gegebenen Messzeitpunkten eingetragen, die im Durchschnitt *282 ms* betragen. An jedem Endpunkt werden bei einem Durchlauf der HTTP-Anfrage jeweils drei Messungen vorgenommen.

Für die Analyse der abgesetzten und empfangenen Daten wird *Wireshark* verwendet, welches die benötigten Messungen für Latenzzeiten an der Netzwerkschnittstelle durchführen kann. *Wireshark* ist ein Werkzeug zur Protokoll- und Datenanalyse. Die *Wireshark*-Messungen werden am Client durchgeführt, damit der gesamte Datenstrom (hin und zurück) innerhalb der Messung berücksichtigt werden kann. Durch das Setzen von Filtern kann *Wireshark* dazu gebracht werden, jeweils nur bestimmte Daten aufzuzeichnen. Der Filter wird auf die IP-Adresse des Servers eingestellt, um anderen lokalen Netzwerkverkehr zu filtern.

Für die Messungen der Datenübertragungsrate wird die Software *iPerf* [iPerf, 2016] verwendet. *iPerf* ist ein Kommandozeilenwerkzeug, welches die Datenübertragungsrate zwischen zwei Endpunkten misst. Der Mess-Rechner an der UDE wird hierbei wieder als Server verwendet und der Rechner an der UKM als Client.

Alle aufgezeichneten Daten werden in *50 MiB* großen Dateien vom Datenformat *.pcap* gespeichert. Diese Dateien werden in das *.csv*-Format konvertiert, damit sie mithilfe von Tabellenkalkulationen bzw. Statistikprogrammen weiterverarbeitet werden können. Hierzu wird das Programm *tshark* verwendet, eine kommandozeilenbasierte Version von *Wireshark*, die Dateien per Stapelauftrag konvertieren kann.

4.2.2 Diagrammdarstellungen für die Messungen

Um die Daten als Grafik darzustellen, werden *Microsoft Excel* und das Statistikprogramm *R* [R, o.D.] genutzt. Damit können verschiedene Graphen und Diagramme für die Visualisierung der Daten erstellt werden. Folgende Diagramme wurden durch *Microsoft Excel* und *R* produziert:

- Punktdiagramm
- Dichtefunktion
- Boxplot

Das *Punktdiagramm* zeigt die Verteilung verschiedener Messwerte eines Wertepaares, eingetragen in ein kartesisches Koordinatensystem. Die X-Achse stellt die Zeitachse des Zeitpunkts der Messung dar bzw. den Zeitpunkt des gesendeten Pakets. Im Falle der Umlaufzeit-Messungen zeigt das Diagramm die Länge der gemessenen Umlaufzeit. Die Verzögerungsvarianz zwischen zwei konsekutiven Umlaufzeiten wird ebenfalls als Punktdiagramm dargestellt. Y-Werte sind hier die berechnete Verzögerungsvarianz. Bei der Darstellung der Datenübertragungsraten befindet sich auf der Y-Achse die Übertragungsgeschwindigkeit.

Die *Dichtefunktion* beschreibt die Auftrittswahrscheinlichkeit bestimmter Werte in einem gegebenen Zeitraum. Diese wird für die Anzahl der verschiedenen Umlaufzeiten berechnet. Die Dichtefunktion der Übertragungswiederholungen (pro Stunde) wird als rote Linie innerhalb des Punktdiagramms dargestellt. Für jede identifizierte Übertragungswiederholung wird die Anzahl aller Übertragungswiederholungen innerhalb der letzten 30 Minuten davor und der nächsten 30 Minuten danach aufgerechnet.

Ein *Boxplot* ist ein Diagramm, das die Verteilung der Werte in übersichtlicher Form darstellt. Hierbei werden die Werte in verschiedene Stufen eingeteilt:

- Median für die höchste Auftrittswahrscheinlichkeit,
- Quartile (oberes/unteres) für 50 % aller Auftrittswahrscheinlichkeiten – der Interquartilsabstand,
- Whisker (oberes/unteres) für Werte bis zum 1,5-fachen Wert des Interquartilsabstands,
- sowie Ausreißer für sehr seltene Auftrittswahrscheinlichkeiten höher als das 1,5-fache des Interquartilsabstands.

Im Gegensatz zur Dichtefunktion können auftretende Werte besser identifiziert werden. Im Falle der Messungen zeigt der Boxplot die aufgetretenen Umlaufzeiten, Datenübertragungsraten und Verzögerungsvarianzen.

4.3 Evaluation der Netzwerkverbindungen

Um einen Eindruck von der Leitungsqualität in Abhängigkeit von den Wochentagen sowie der Tageszeit zu bekommen, wurden die jeweiligen Messungen über mehrere Tage durchgeführt. Daraus wurden die folgenden Daten extrahiert:

- Umlaufzeiten
- Verzögerungsvarianzen
- Datenpaketverluste
- Datenübertragungsrate

4.3.1 Umlaufzeitmessungen der Netzwerkverbindungen

Die Umlaufzeit wurde jeweils über einen zweiwöchigen Zeitraum mithilfe der oben vorgestellten HTTP-Methode durchgeführt. Die nachfolgende Tabelle 4.7 zeigt für alle zur Verfügung stehenden Verbindungen den Zeitraum der Messungen.

Verbindung	Start (UTC+1)	Ende (UTC+1)	Test-laufzeit [s]
UKM-Standard	30.03.2016, 11:26	13.04.2016, 21:25	1.245.540
Polycom	08.03.2016, 09:35	25.03.2016, 01:03	1.438.680
3G	19.04.2016, 06:47	02.05.2016, 12:12	1.142.700

Tabelle 4.7: Erfassungsdetails der Leitungsmessungen für Verzögerungen

Aus den Langzeitmessungen ergaben sich deutliche Unterschiede zwischen den drei Verbindungen. Die UKM-Standardverbindung und die Polycom-Verbindung verhalten sich sehr ähnlich. Weitaus schlechter verhält sich die 3G-Verbindung, die große Unter-

schiede zu den anderen besitzt. Abbildung 4.4 zeigt zwei typische Tagesverläufe der drei UKM-Standardverbindung im Vergleich mit der 3G-Verbindung. Ein vollständiges Diagramm über die gesamte Messung befindet sich im Anhang.

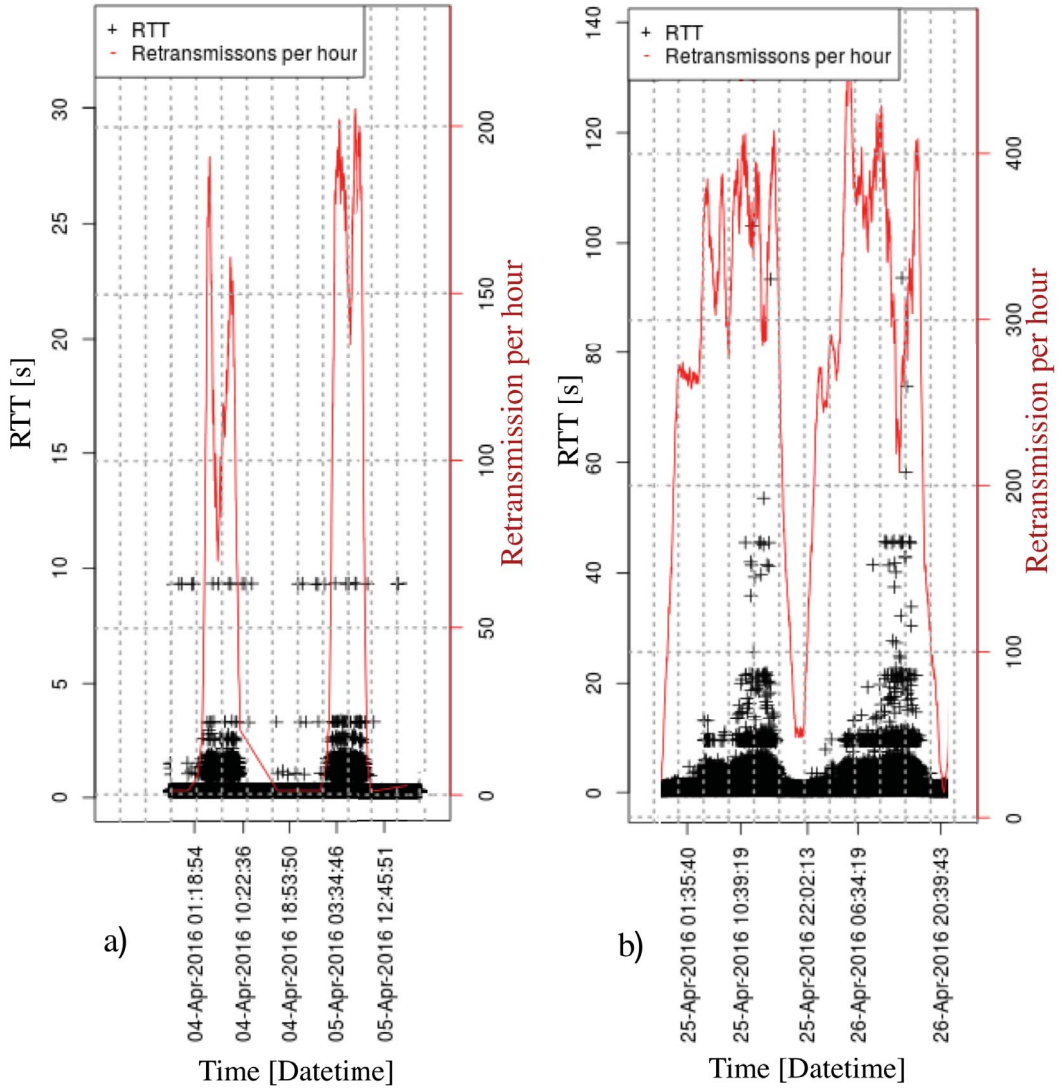

Abbildung 4.4: Umlaufzeiten mit Übertragungswiederholungen, a) UKM-Standard-Netzanbindung, b) 3G-Netzanbindung

Gemessene Umlaufzeiten werden durch ein schwarzes Kreuz markiert. Wie zu erkennen ist, liegen diese hauptsächlich nahe der Nulllinie, da die Skala zu grob ist, um den genauen Wert ablesen zu können. Ausreißer existieren in bestimmten Abständen und bei Spitzenwerten von mehr als *30 Sekunden* bei der UKM-Standardverbindung. Bei der 3G-Netzanbindung betragen Spitzenwerte bis zu *140 Sekunden*. Deutlich zu erkennen sind die Abstufungen der kleineren Ausreißer, die durch Timeouts und Übertragungswiederholungen aufgrund von Datenpaketverlusten entstehen.

Rot markiert ist der Dichtewert der Übertragungswiederholungen pro Stunde. Dieser tritt erhöht bei Ausreißern der Umlaufzeiten auf. Zu diesen Zeiten existieren erhöhte Datenpaketverluste, die zu Übertragungswiederholungen führen. Diese Ballungen treten

vor allem zu Stoßzeiten (08-18 Uhr) während der regulären Tageszeit in Malaysia auf.[29] Die im Anhang befindliche vollständige Darstellung zeigt außerdem, dass erhöhte Datenpaketverluste vor allem an Wochentagen existieren – im Gegensatz zu den Wochenenden, an denen nur wenig Datenpaketverluste auftreten.

Diese Ballungen von Übertragungswiederholungen weisen auf Stausituationen auf der Übertragungsstrecke hin. Tabelle 4.8 stellt die durchschnittliche Anzahl der Übertragungswiederholungen der drei Anbindungen innerhalb eines stündlichen Aufnahmefensters dar.

Verbindung	Übertragungs-wiederholungen, Messzeit	Übertragungs-wiederholungen, 1/h
UKM-Standard	4933	14,25
Polycom	8628	12,74
3G	85288	268,7

Tabelle 4.8: Übertragungswiederholungen der Verbindung UDE - UKM

Aus der Tabelle geht hervor, dass sich die Übertragungswiederholungen bei der Polycom-Verbindung und der Standard-Verbindung sehr ähneln – im starken Kontrast zur 3G-Verbindung, die unter bis zu 20-fach höheren Übertragungsproblemen leidet.

Einen besseren Aufschluss über die am häufigsten auftretenden Umlaufzeiten liefern die Dichtefunktionen in Abbildung 4.5. Die Darstellung zeigt eine detaillierte Ansicht der UKM-Standardverbindung sowie der 3G-Verbindung.[30] Es wird nur die höchste Dichte dargestellt.

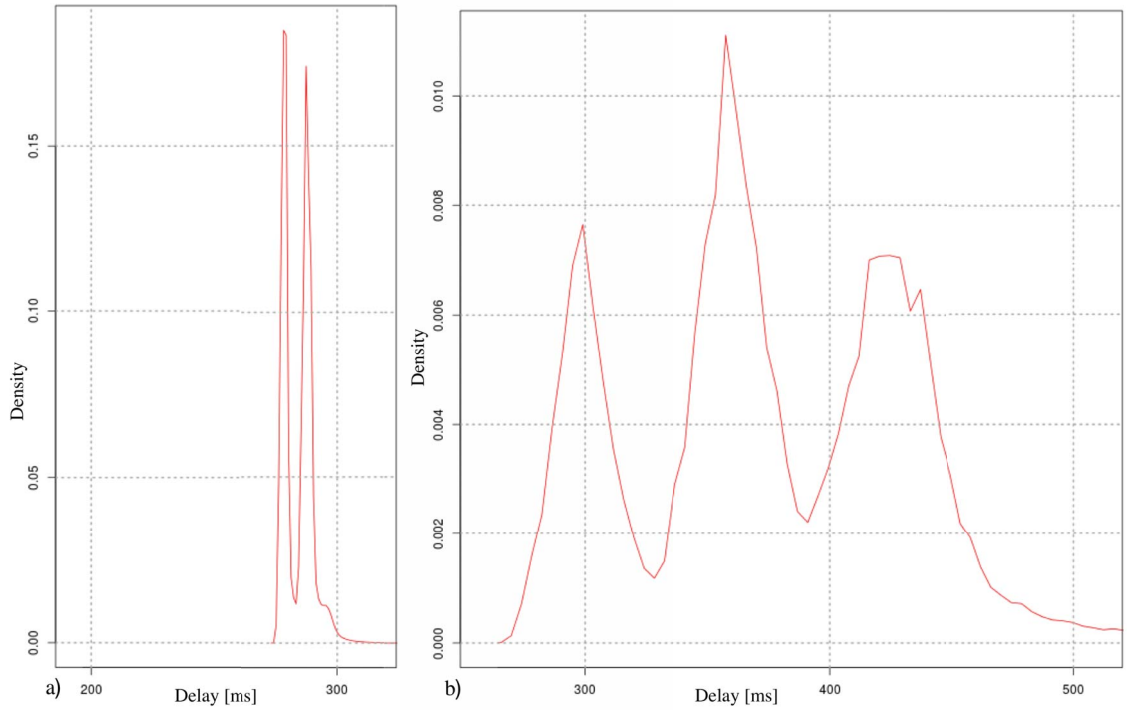

Abbildung 4.5: Dichtefunktionen der Umlaufzeitmessungen, a) UKM-Standard-Netzanbindung, b) 3G-Netzanbindung

[29] Die angegebenen Zeiten in der Grafik entsprechen den Messzeitpunkten der deutscher Zeit GMT+1
[30] eine Dichtefunktion der UKM-Polycomverbindung befindet sich im Anhang

Wie aus der Abbildung hervorgeht, existieren in den Grafiken mehrere Spitzenwerte, die auf unterschiedliche Routen hinweisen, die hauptsächlich während der Messungen geschaltet waren. Ein Vergleich der Dichtefunktionen zeigt die niedrigsten Werte bei der UKM-Standardverbindung, die den stabilsten Verlauf besitzt. Eine Routenänderung kommt vor, unterscheidet sich aber lediglich um *10 ms*. Die kleinsten Werte bei der 3G-Verbindung sind im Vergleich mit der anderen nur geringfügig höher, jedoch mit weitaus geringerer Auftrittswahrscheinlichkeit.

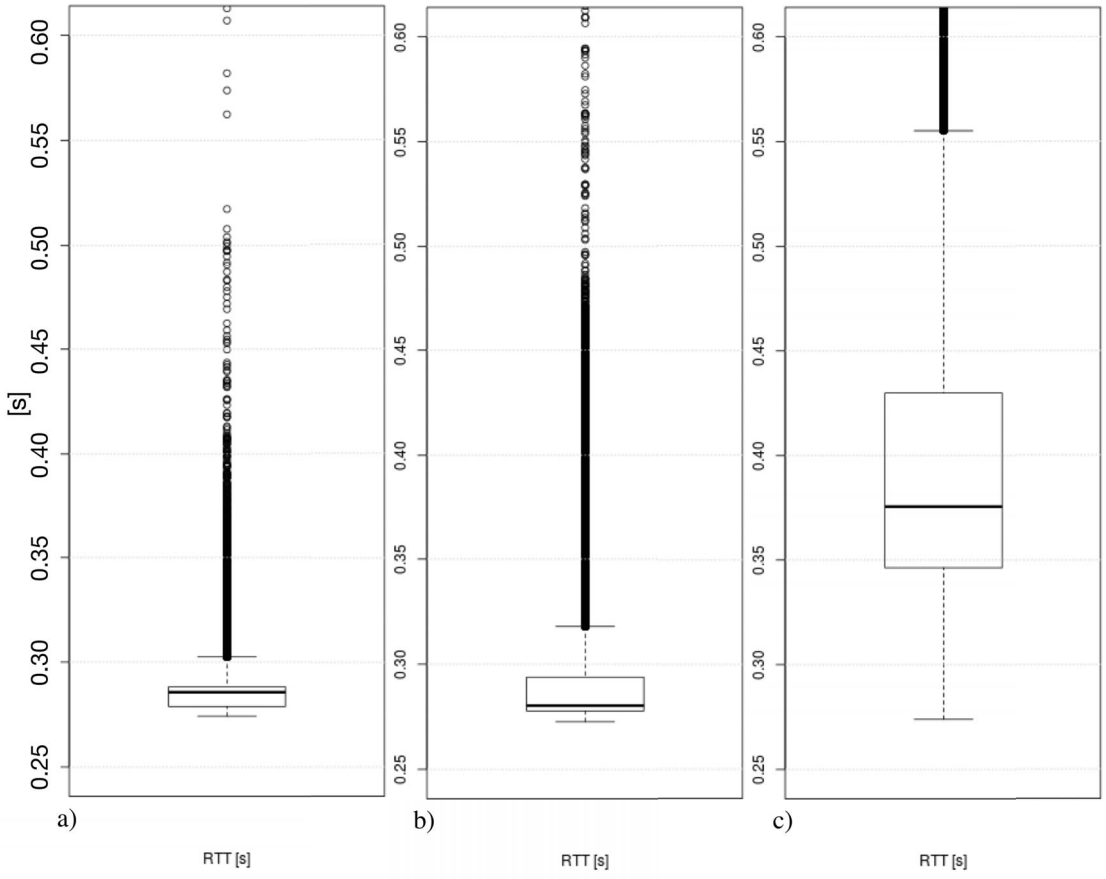

Abbildung 4.6: Boxplots der Umlaufzeiten der Verbindungen, (a) UKM-Standard-Netzanbindung, b) Polycom-Netzanbindung, c) 3G-Netzanbindung

Die Boxplots in Abbildung 4.6 zeigen die gemessenen Umlaufzeiten als Boxendarstellung. Die UKM-Standardverbindung und die Polycom-Verbindung ähneln sich weitestgehend. Die Standardverbindung besitzt etwas weniger Ausreißer und einen kleineren Median als die Polycom-Verbindung. Im Kontrast hierzu besitzt die Polycom-Verbindung einen etwas geringeren Median. Deutlich sichtbar ist in dieser Darstellung der Unterschied zur 3G-Verbindung. Der Median liegt deutlich höher und die Quartile sind sehr viel breiter. Die niedrigsten Werte der 3G-Verbindung liegen dennoch mit den kleinsten Werten der anderen Verbindungen gleichauf.

Insgesamt weisen die Grafiken in Bezug auf die Umlaufzeit auf ein zum Teil stark schwankendes Verhalten der Verbindungen. Diese sind abhängig von der Tages- und Nachtzeit und besitzen an Wochentagen die höchsten Werte für Datenpaketverluste. Durch hohe Verlustwerte werden die Verzögerungen stark erhöht und liegen zum Teil

im Sekundenbereich. Auf der anderen Seite liegen die durchschnittlichen Verzögerungswerte in Bereichen, die unter den in Tabelle 2.11 in Kapitel 2.5.2 spezifizierten Anforderungen liegen. Das Basisszenario ist also im Bereich der Umlaufzeit grundsätzlich umsetzbar. Probleme bereiten die starken Datenpaketverluste und damit einhergehenden Schwankungen. Die folgende Tabelle 4.9 zeigt eine Gesamtübersicht der erfassten Charakteristiken:

Verbindung	Mittelw. RTT [ms]	Min. RTT [ms]	Max. RTT [s]
UKM-Standard	286,3 ms	274,3 ms	33,1 s
Polycom	299,5 ms	272,6 ms	21,6 s
3G	474,3 ms	273,6 ms	136,9 s

Tabelle 4.9: Charakteristiken der Messung UDE-UKM

4.3.2 Verzögerungsvarianz der Netzwerkverbindungen (Jitter)

Die Verzögerungsvarianzen können ebenfalls aus den oben diskutierten Langzeitmessungen für die Umlaufzeiten unter Verwendung von Formel 2.3 extrahiert werden. Die nachfolgenden vergrößerten Boxplots in Abbildung 4.7 zeigen die Verzögerungsvarianzen der drei Verbindungen.

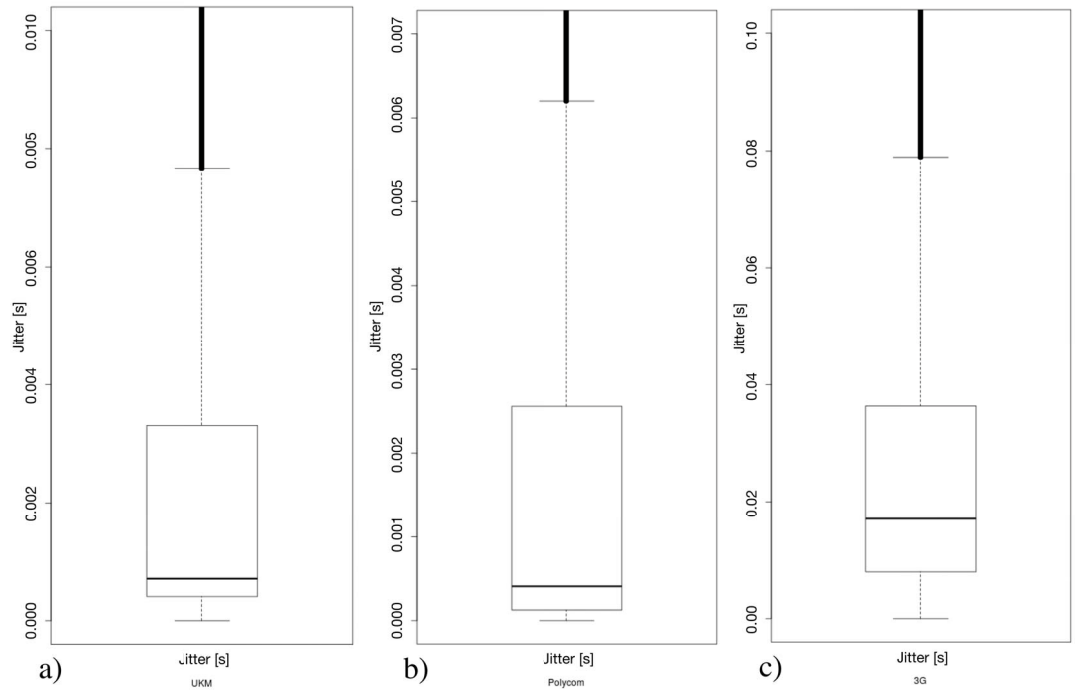

Abbildung 4.7: Boxplots der Verzögerungsvarianz (Jitter) der Verbindungen, (a) UKM Standard-Netzanbindung, b) Polycom-Netzanbindung, c) 3G-Netzanbindung

Deutlich sichtbar sind die starken Schwankungen. Insgesamt sind die Werte der UKM-Standard-Verbindung am stabilsten, wobei der Median ein wenig höher liegt als bei der Polycom-Verbindung. Die 3G-Verbindung besitzt die größten Schwankungen in den Latenzzeiten.

Insgesamt liegt der Median der Verzögerungsvarianzen der UKM-Standardverbindung und der UKM-Polycom-Verbindung in einem niedrigen Bereich. Schwankungen

übersteigen jedoch die in Tabelle 2.11 spezifizierten Anforderungen an das Basisszenario: Vor allem die optimalen Jitterwerte für die ARTP-Maschinensteuerung liegen bedenklich nahe oberhalb der Whisker. Der Median der 3G-Verbindung liegt bereits oberhalb dieses Wertes. Die nachfolgende Tabelle 4.10 zeigt die durchschnittlichen Werte für Jitter. Daraus geht hervor, dass die 3G-Verbindung deutlichen Schwankungen unterliegt: Der Mittelwert besitzt einen kritischen Wert für viele der angedachten Anwendungen. Im Gegensatz hierzu besitzen die anderen beiden Verbindungen einen relativ niedrigen Jitter.

Verbindung	Median Jitter [ms]	Mittelw. Jitter [ms]	Min. Jitter [ms]	Max. Jitter [s]
UKM-Standard	0,71 ms	6,33 ms	0 ms	32,82 s
Polycom	0,41 ms	6,96 ms	0 ms	21,4 s
3G	17,02 ms	152,12 ms	0 ms	136,6 s

Tabelle 4.10: Jitter-Charakteristiken der Messung UDE-UKM

4.3.3 Datenübertragungsraten der Netzwerkverbindungen

Für die Evaluation der Datenübertragungsrate der drei Verbindungen wurde das Programm *iPerf* verwendet. Zu diesem Zweck wurden Messreihen entworfen, die jeweils eine Woche durchgeführt wurden. Eine einzelne Messung wurde für *20 Sekunden* jede halbe Stunde vorgenommen. Die nachfolgende Tabelle 4.11 gibt Aufschluss über die durchgeführten Messungen.

Verbindung	Start (UTC+1)	Ende (UTC+1)	Test-laufzeit [s]	Anzahl der Messungen (n)	Bytes übertragen gesamt [MByte]
UKM-Standard	30.05.2016 01:29	07.06.2016 02:06	7.700 s	385	260,41 MB
Polycom	13.06.2016 00:41	21.06.2016 01:32	7.720 s	386	390,98 MB
3G	27.06.2016 21:15	04.07.2016 21:26	6.640 s	332	274,66 MB

Tabelle 4.11: Erfassungsdetails der Leitungsmessungen für Datenübertragungsrate

Bei den Messungen für die Datenübertragungsrate stellt sich ein etwas anderes Bild dar, als es bei den vorherigen Messungen der Fall gewesen ist. Die nachfolgende Abbildung 4.8 und Abbildung 4.9 zeigen Darstellungen der gemessenen Datenübertragungsraten über einen Zeitraum von 24 Stunden. Ein vollständiges Diagramm für den gesamten Messzeitraum kann im Anhang eingesehen werden.

Sichtbar sind vor allem die zeitlichen Schwankungen der 3G-Verbindung. Am Tag sinken die Datenübertragungsraten auf ein Minimum, in der Nacht jedoch sind sie weitaus größer: Spitzenwerte der 3G-Verbindung mit über *800 Kbit/s* überragen die anderen Verbindungen deutlich. Die höchsten Datenübertragungsraten bestehen hauptsächlich in der Nacht bis vormittags zwischen 0-11 Uhr. Die beiden anderen Verbindungen besitzen

eine relativ stabile Datenübertragungsrate, wobei die UKM-Standardverbindung ein wenig schlechter abschneidet und ebenfalls leichten Schwankungen in Abhängigkeit von der Tag- und Nachtzeit unterliegt. Beide Verbindungen besitzen einen Verlauf um etwa *300 Kbit/s*.

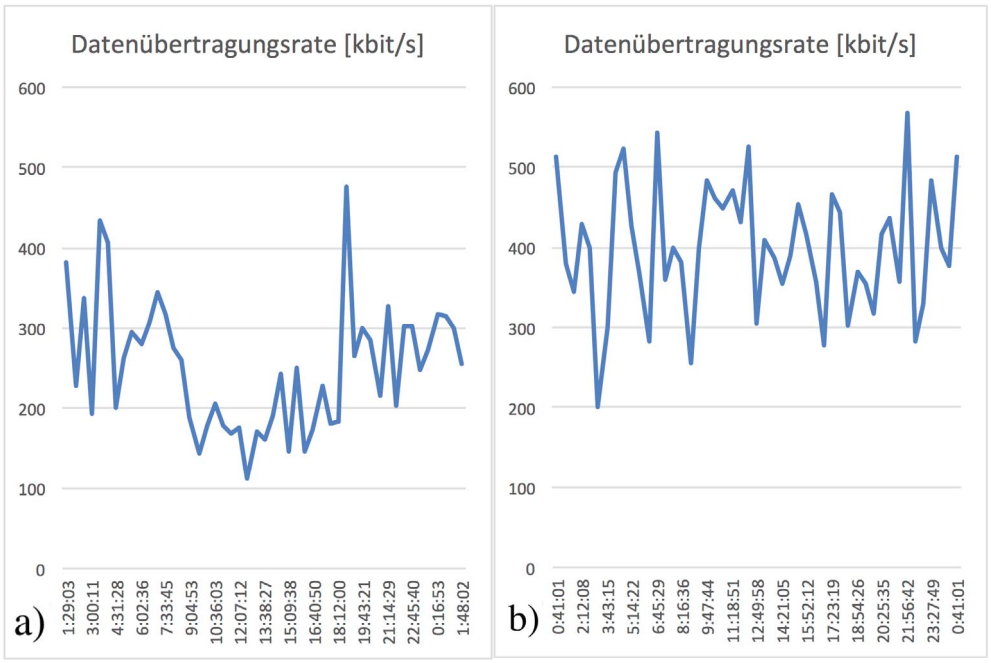

Abbildung 4.8: UDE/UKM Datenübertragungsraten eines Tages, a) UKM Standard-Netzwerkanbindung, b) Polycom-Netzanbindung

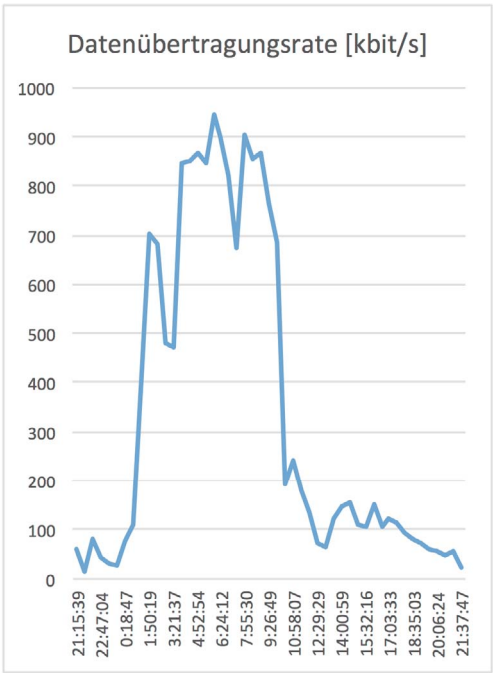

Abbildung 4.9: Datenübertragungsraten eines Tages, 3G-Verbindung

Die nachfolgenden Boxplots in Abbildung 4.10 zeigen die durchschnittlichen Werte der Datenübertragungsrate. Hierbei ist zu erkennen, dass die Polycom-Verbindung den stabilsten Verlauf bietet. Die breiteste Streuung ist bei der 3G-Verbindung auszumachen.

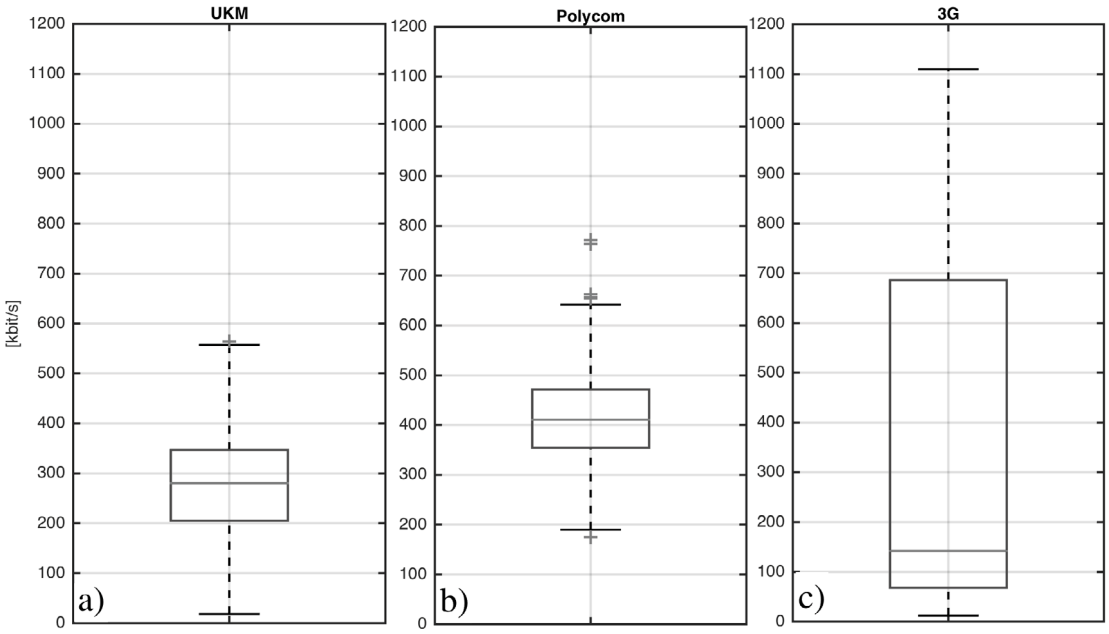

Abbildung 4.10: Boxplots der Datenübertragungsrate der Verbindungen [kbit/s], a) UKM Standard-Netzanbindung, b) Polycom-Netzanbindung, c) 3G-Netzanbindung

Die nachfolgende Tabelle 4.12 zeigt die Mittelwerte der durchgeführten Messungen. Mit einem deutlichen Unterschied besitzt die Polycom-Verbindung die höchste durchschnittliche Datenübertragungsrate und die stabilsten Werte.

Verbindung	Mittelwert Datenübertr. [Kbit/s]	Min. Datenübertr. [Kbit/s]	Max. Datenübertr. [ms]
UKM-Standard	278,015 Kbit/s	18,5 Kbit/s	564 Kbit/s
Polycom	426,152 Kbit/s	175 Kbit/s	2.590 Kbit/s
3G	345,767 Kbit/s	12,4 Kbit/s	1.110 Kbit/s

Tabelle 4.12: Charakteristiken der Datenübertragungs-Messung UDE-UKM

Die gemessenen Datenübertragungsraten zeigen, dass über die Polycomverbindung eine relativ stabile Verbindung möglich ist. Diese sinkt nur in Ausnahmefällen unter *350 kbit/s*. Grundsätzlich sind damit alle Dienstgüte-Anforderungen des Basisszenarios aus Tabelle 2.11 erfüllt. Eine qualitativ hochwertige Videoverbindung in High Definition ist jedoch unter diesen Umständen nur schwerlich möglich, da hierfür im besten Fall mehrere Mbit/s Datenübertragungsrate zur Verfügung stehen. Für eine normale Videokonversation ist die Verbindung jedoch ausreichend.

4.4 Verbesserung der Dienstgüte durch multiple Pfade

Die oben durchgeführten Messungen offenbaren deutliche Schwächen im Bereich der Verzögerungen und der Datenpaketverlustrate. Die größten Probleme wirft die mobile 3G-Verbindung auf, aber auch die anderen Verbindungen weisen Mängel aus. Vor allem ist die Dienstgüte stark von den Tages- und Wochenzeiten abhängig. Zu den regulären Arbeitszeiten steigen die Datenpaketverluste an.

Die 3G-Verbindung ist im Vergleich zu den anderen Verbindungen starken Schwankungen unterworfen. Die Verzögerungszeiten überschreiten zu Stoßzeiten die Zwanzigsekundenmarke, die bei den anderen Verbindungen nur selten überschritten wird. Solche hohen Verzögerungen entstehen nur durch starke Verbindungsprobleme, da normalerweise ein Datenpaketverlust durch das erneute Senden nur etwas mehr als die Verdopplung der Umlaufzeit bedeutet[31] – in diesem Fall also ca. *600 ms*. Die Basisverzögerung liegt jedoch bei allen Verbindungen in einem ähnlichen Bereich.

Die beiden vielversprechendsten Verbindungen, die UKM-Standard-Verbindung und die Polycom-Verbindung, besitzen in etwa dieselbe Route. Einen Unterschied macht hier die 3G-Verbindung, die über London und Hong-Kong geroutet wird. Sie unterscheidet sich von den anderen Verbindungen ab dem X-Win-Knotenpunkt der Universität Duisburg-Essen. Die Unabhängigkeit der verschiedenen Routen kann sich als Vorteil erweisen. Datenpaketverluste auf den verschiedenen Routen sind mit hoher Wahrscheinlichkeit unkorreliert, da die Teilstrecken nicht identisch sind. Eine Verbindung könnte im Störungsfall die andere Verbindung ersetzen und umgekehrt.

Eine neue Versuchsanordnung mit parallelen, gleichzeitig durchgeführten Messungen wurde nach der oben beschriebenen Methode der HTTP-Anfrage initiiert. Die Messergebnisse wurden an unterschiedlichen Wochentagen zu randomisierten Zeiten durchgeführt. Für die Auswertung wurde ein Schwellwert bei *400 ms* definiert. Dieser Schwellwert gilt als ein Datenpaketverlust. Abbildung 4.11 zeigt die Schwellwertüberschreitungen pro Sekunde der UKM-Standard-Netzwerkverbindung (rot) und der UKM-3G-Verbindung (blau) für einen Ausschnitt der Messung (Freitag).

Aus dem Diagramm hervorgehend sind die massiven Schwellwertüberschreitungen der 3G-Verbindung in blau dargestellt. Schwellwertüberschreitungen der Standard-Verbindung in rot sind zwar nicht so zahlreich, jedoch ebenfalls vorhanden. Eine kombinierte Schwellwertüberschreitung (orange) wird dann gezählt, wenn beide Verbindungen zur selben Zeit die festgelegte Schwelle überschreiten.

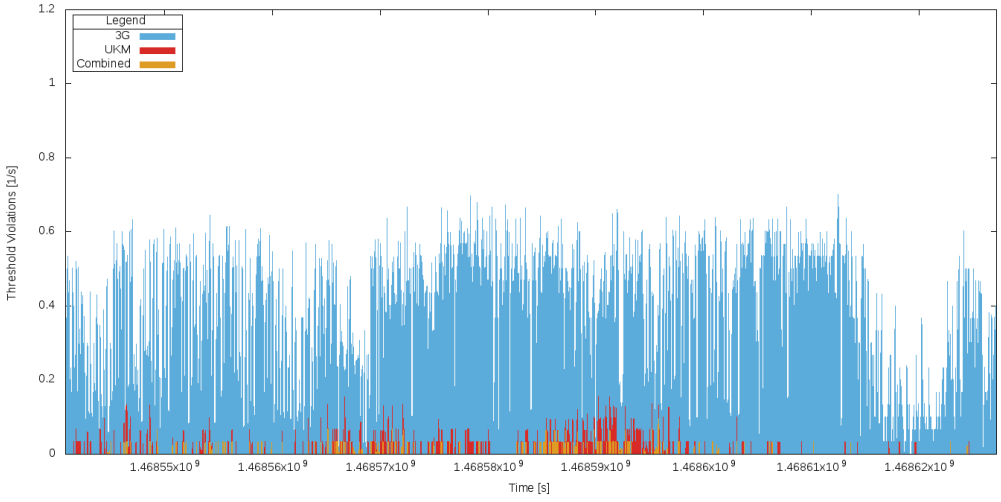

Abbildung 4.11: Schwellenwertdiagramm und Korrelation von UKM-Standard und 3G-Verbindung

[31] Vgl. Kapitel 3.3.6.3

Tabelle 4.13 zeigt die für die randomisierten Zeiten gemessenen durchschnittlichen Schwellwertüberschreitungen (innerhalb eines 30-Sekunden-Fensters) an unterschiedlichen Wochentagen. Die UKM-Standardverbindung besitzt weitaus weniger Schwellwertüberschreitungen als die 3G-Verbindung. Kombiniert können diese beiden Zahlen reduziert werden. Das Verhältnis für eine Verbesserung der UKM-Standardverbindung mithilfe der eigentlich schlechteren 3G-Verbindung befindet sich in der letzten Spalte. Das Ergebnis ist ein Durchschnittswert für eine Verbesserung der Datenpaketverlustrate von *43 %*. Demgegenüber stehen Spitzenverbesserungen von bis zu *73 %*.

Schwellwert-überschreitung	UKM Standard-Verb.	3G-Verb.	Kombiniert	Verhältnis Kombiniert/UKM
Freitag	606	23621	163	0,269
Samstag	582	31741	175	0,301
Sonntag	621	28887	187	0,301
Montag	726	31114	305	0,420
Dienstag	506	48388	383	0,757
Mittwoch	1314	52237	1079	0,821
Donnerstag	1289	70722	1083	0,840
Gesamt	**6180**	**301696**	**3506**	**0,567**

Tabelle 4.13: Anzahl der Schwellwertüberschreitungen zu unterschiedlichen Zeitpunkten für zwei kombinierte Verbindungen

Um diese Option weiter zu veranschaulichen, wurden die Übertragungswiederholungen aus den erhobenen Daten extrahiert und auf ein Liniendiagramm gebracht. Abbildung 4.12 zeigt die Darstellung der Übertragungswiederholungen der beiden Verbindungen. Das Diagramm zeigt die durchschnittlichen Übertragungswiederholungen für den gemessenen Zeitraum – berechnet für ein einstündiges Fenster.

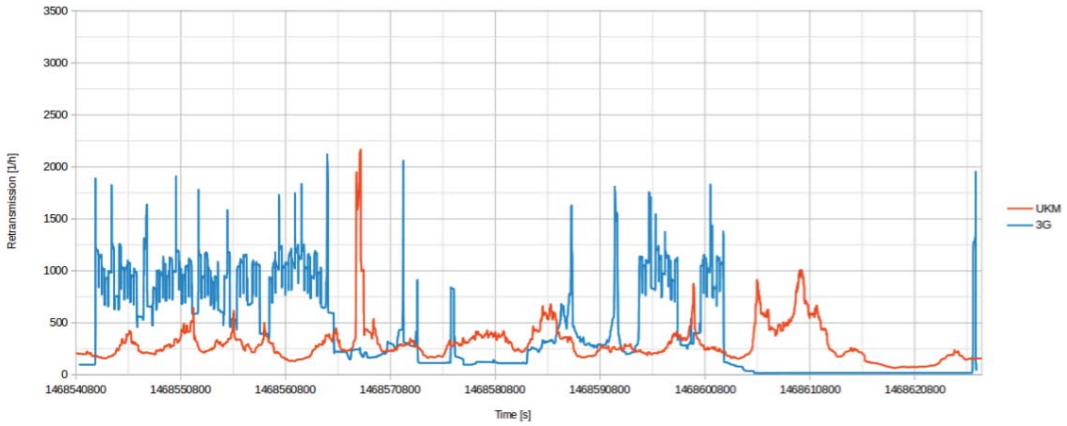

Abbildung 4.12: Übertragungswiederholungen der UKM-Standard Verbindung und der 3G-Verbindung [Verbunt, 2017]

Wie aus der Abbildung hervorgeht, leidet die 3G-Verbindung (blau) an den meisten Übertragungswiederholungen. Dies ist vor allem zu Beginn des Messzeitraums der Fall. Die UKM-Standardverbindung besitzt ebenfalls Übertragungswiederholungen. Beide erscheinen im Diagramm sehr unkorreliert und können bei einer zeitgleichen Nutzung der Verbindungen zu einer Minimierung der Wiederholungen führen.

4.5 Fazit der Verbindungsauswertung

In diesem Kapitel wurden Messungen vorgenommen, die die Leitungsqualität der Verbindung zwischen den beiden Standorten erfassen. Dabei hat sich die Unbeständigkeit der Verbindung gezeigt. Unter Verwendung von zwei Verbindungen kann eine erste Hypothese aufgestellt werden, um die Leitungsqualität zu verbessern:

Die Nutzung mehrerer Verbindungen ist eine Option, um die Dienstgüte zu verbessern und die Ressourcen der UKM-Anbindungen zum Vorteil auszunutzen: Eine gleichzeitige Nutzung der Verbindungen kann dazu führen, die Verlustwahrscheinlichkeit zu verringern.

Die Auswertung der Netzwerkverbindung zwischen UDE und UKM hat ergeben, dass die in Kapitel 2.5.2 erarbeiteten Dienstgüte-Metriken für ein telemedizinisches Szenario grundsätzlich ausreichend sind. Sowohl die Durchschnittswerte der Umlaufzeit als auch die Varianz und die Datenübertragungsrate liegen innerhalb der benötigten Werte. Starke Schwankungen, vor allem bei den Umlaufzeiten, führen jedoch schnell zu Situationen, in denen Verbindungen stark in Mitleidenschaft gezogen werden. Eine Verzögerung von mehreren Sekunden kommt einem Abbruch der Verbindung gleich. Vor allem bei der Steuerung eines ARTPs muss die Auftrittswahrscheinlichkeit von Datenpaketverlusten minimal sein und ein vorhandener Jitter möglichst niedrig.

In Tabelle 4.14 werden die in Kapitel 2.5.2 aufgestellten Dienstgüteanforderungen den gemessenen durchschnittlichen Bestwerten für eine Verbindung zwischen UKM und UDE gegenübergestellt. Die Werte geben Hinweise auf eine grundsätzlich mögliche Realisierung des Basisszenarios über eine Verbindung zwischen UDE und UKM. Eine Ende-zu-Ende-Verzögerung wird mithilfe der Umlaufzeit durch Formel 4.2 unter der Annahme von symmetrischen Pfaden berechnet. Eine tatsächliche Fehlerrate kann nur durch eine erschöpfende Messung der Datenübertragungsrate durchgeführt werden. Diese wurde bei den oben vorgenommen Messungen nicht angestrebt.

Anwendung	**Dienstgüte Anforderungen**			
	Datenüber-tragungsrate	**Ende-zu-Ende Verzögerung**	**Jitter**	**Fehlerrate**
Gemeinsamer Wert der Dienstgüte-Anforderungen	akkumuliert Min.: ca. 200 Kbit/s Max.: >1 Gbit/s	Min.: 165 ms Best.: < 55 ms	Min.: 40 ms Best.: < 5 ms	Min. abh. von E-zu-E und Jitter Best.: 0 Fehler/s
Messungen UDE/UKM (Avg, Best, Worst)	426 Kbit/s, 2,590 Kbit/s, 12,4 Kbit/s.	143,15 ms, 136,3 ms, 68,45 s.	6,33 ms, 0 ms, 136,6 s	Mindestens 12,74 1/h,

Tabelle 4.14: Charakteristiken der Übertragungsstrecke und Dienstgüte-Anforderungen des Szenarios

5 Erweiterte Dienstgüteverbesserung und aktueller Stand der Technik

Telemedizinische Anwendungen benötigen zuverlässige Verbindungen, die unter möglichst wenig bzw. keinem Datenpaketverlust leiden dürfen und zeitnah zugestellt werden müssen. Die bisher diskutierten Technologien bieten keine ausreichende Handhabe, auf Ende-zu-Ende-Ebene die Latenzzeiten und Datenpaketverluste auszugleichen.

Über die im vorherigen Kapitel beschriebenen Methoden hinaus werden in diesem Kapitel Techniken vorgestellt, die eine erweiterte Dienstgüteverbesserung zur Verfügung stellen. Es soll der aktuelle Stand der Technik dargestellt und Anknüpfungspunkte für die in dieser Arbeit durchgeführte Entwicklung veranschaulicht werden.

Es werden zunächst Mehrwegprotokolle vorgestellt und eine Übersicht ihrer Vorteile gegeben. Danach folgt eine Darstellung der Methoden, die unter Verwendung verschiedener Kommunikationswege genutzt werden können. Die grundlegenden Funktionen von MPTCP zur Verwendung als Protokollbasis, die für eine Modifikation genutzt werden sollen, werden erläutert und es wird die Vorgehensweise für die Entwicklung eines redundanten Mehrwegprotokolls dargelegt. Zum Schluss wird auf eine parallele Entwicklung eingegangen, die eine ähnliche Zielsetzung besitzt. Die Unterschiede zum geplanten Vorhaben werden herausgearbeitet.

5.1 Multihoming und Mehrwegprotokolle zur Verbesserung der Dienstgüte

Mehrwegtechniken können dafür eingesetzt werden, um mithilfe mehrerer Datenpfade die Dienstgüte zu verbessern. Dies führt zu [Qadir et al., 2015]:

- erhöhter Zuverlässigkeit
- verbessertem Datendurchsatz
- besserer Netzwerkeffizienz
- höherer Fehlertoleranz

Auf der untersten ISO/OSI-Schicht dienen sie dazu, mehrere physikalische Pfade redundant zu nutzen. Dieses Verfahren wird in speziellen Netzwerkanwendungen wie industriellen Netzwerken oder auf besonders gefährdeten Strecken angewandt.

Funktechniken besitzen natürlicherweise bereits eine Mehrwegausbreitung. Diese kann mithilfe von verschiedenen Modulations- bzw. Kodierungsverfahren verstärkt genutzt werden. Diese Technologien sind nach wie vor Teil von umfassender Forschung [Khasawneh et al., 2015] [Shuminoski & Janevski, 2016]. Funktechnologien besitzen

einen hohen Grad an Mobilität, bei der Mehrwegtechniken für das Daten-Roaming und -Routing angewandt werden können. Diese Techniken sind jedoch nicht Teil dieser Arbeit.

Mehrwegtechniken, die heterogen vermaschte Strukturen von Netzwerken wie dem Internet nutzen, setzen oberhalb der Sicherungsschicht des OSI-Referenzmodells auf. Hierfür müssen die Netzwerkentitäten Zugang zu mehreren anderen Entitäten oder Netzwerken erhalten. Herkömmliche Einzelwegtechniken nutzen eine Vermaschung nur ineffektiv aus, da eine Ende-zu-Ende-Verbindung durch die Datenübertragungsrate eines einzelnen Pfades limitiert ist [Qadir et al., 2015]: Durch kleine Veränderungen der Dienstgüteeigenschaften, etwa durch hinzukommenden Datenverkehr, können große Oszillationen innerhalb des Netzwerks entstehen (ebd.). Durch Mehrwegtechniken lässt sich das Netzwerk hingegen effizienter und gleichmäßiger nutzen. Dieses Prinzip wird auch *Ressource Pooling* [Wischik et al., 2008] genannt. Dabei geht es darum, dass mehrere Ressourcen gebündelt werden und sich so wie eine einzelne Ressource verhalten.

Beim *Multihoming* besitzt ein privates Netzwerk oder eine Netzwerkentität mehrere Anschlüsse, die verschiedene Netzwerkpfade zu einem Ziel ermöglichen [Akella et al., 2003]. Multihoming wird in der Regel über unterschiedliche ISPs realisiert, um *Pfadunabhängigkeit (Pfad-Diversität, PD)* zu gewährleisten [Apostolopoulos & Trott, 2004]. Es bietet die Basis für transportorientierte *Mehrwegprotokolle*, die in der Lage sind, die verfügbaren Anschlüsse auszunutzen. Dabei werden Mehrwegprotokolle auf den netzwerkorientierten Schichten des ISO/OSI-Referenzmodells eingesetzt.

Die wichtigsten Ziele und Vorteile von Mehrwegtechniken sind [Tsai & Moors, 2006]:

- *Lastverteilung bzw. Lastenausgleich (load balancing)* durch Umverteilung von Daten bei Überlastung eines Pfades
- *Aggregation der Datenübertragungsrate* durch gemeinsames Ausnutzen der Datenübertragungsraten mehrerer Pfade
- *Reduzierte Verzögerung* durch Ausfallsicherung der Pfade
- *Robustheit,* da keine einzelnen Fehlermöglichkeiten (*Single-Point-of-Failure – SPOF*) vorhanden sind. Techniken der Ausfallsicherung (*Failover*) und Dienstübergabe (*Roaming*) können bei Ausfällen den Datenstrom umlegen
- *Fehlertoleranz* durch Redundanz mehrerer unterschiedlicher Pfade

Zumeist wurde Multihoming in Verbindung mit Funktechnologien untersucht [Frantti & Majanen, 2013], da viele Probleme in diesem Bereich durch die Natur eines geteilten Funkmediums hervorgerufen werden und entscheidende Einflüsse auf die Dienstgüte verzögerungsintoleranter Anwendungen haben. Hierzu gehören Paketverluste und schwankende Verbindungen, die einen unstetigen Datenfluss hervorrufen.

Um Schwankungen bei Verzögerungszeiten unterschiedlicher Pfade im Internet zu überwinden, wurde in [Kim et al., 2012] ein System entwickelt, durch das sich die Unabhängigkeit multipler Pfade zu einem gewissen Grad bestimmen lässt. Dabei entwickelten die Autoren ein adaptives Pfad-Management für die Unabhängigkeit der Pfade.

Der Algorithmus arbeitet mit Sondierungspaketen, um die Pfade zu vermessen und um die beste Auswahl an Pfaden für eine Übertragung zu nutzen. Diese Technik wurde für Verbesserungen in Networked Control Systems entwickelt, die in kleineren hochspezialisierten Netzwerken realisiert werden kann.

Mobile Endgeräte (z.B. Smartphones) profitieren von ihren Multihoming-Fähigkeiten durch eine mögliche Nutzung mehrerer Übertragungstechnologien. Beim Wechsel der Anbindung kann unter Einsatz von Multipathtechniken eine beschleunigte Verbindungsübergabe auf Ende-zu-ende-Ebene geschehen [Barré et al., 2011]. Hierbei werden mehrere Verbindungen offengehalten und bei Verlust der verwendeten Verbindung sofort ein direktes Failover ermöglicht.

Das Internet ist eine Zusammensetzung aus vielen heterogenen Netzwerken, die als autonome Systeme fungieren. Die traditionelle Nutzung der Ressourcen dieser Systeme vollzieht sich in der Weiterleitung von Datenpaketen und dem dahingehenden Aufbau effizienter Routingmechanismen. Darauf beruhen die bisher eingesetzten Protokolle und Techniken. Die vermaschte und redundante Struktur der Netzwerke bietet allerdings eine Basis für eine weitaus variablere Nutzung. Sie kann der logische Ausweg für die Lösung vieler Probleme sein, die durch die veränderte Verwendung des Internets im Vergleich zu seinen Anfängen entstanden sind. Durch sie werden das Internet und die Entwicklung neuer Protokolle jedoch vor neue Schwierigkeiten gestellt, die im Forschungsbereich der Mehrwegprotokolle thematisiert werden. Die zu lösenden Probleme lassen sich in folgende Kategorien einteilen (nach [Qadir et al., 2015] mit eigenen Erweiterungen):

1. *Verwendung von Netzressourcen:* Die parallele Übertragung von Datenströmen muss innerhalb des Netzwerks organisiert werden. Dieser Bereich beschäftigt sich mit der eigentlichen Nutzung von Datenübertragungen über mehrere Pfade sowie damit, wie diese praktisch innerhalb eines Netzwerks umgesetzt wird. Es geht darum, die Ressourcen, die ein Netzwerk bereitstellt, möglichst effektiv zu nutzen und gleichmäßig auszulasten (Ressource Pooling).

2. *Integrität von Datensendungen:* Eine gleichzeitige Nutzung mehrerer Pfade für einen Datenstrom offenbart Probleme bei der Integrität von Datensendungen. Datenströme müssen auf verschiedene Pfade verteilt sowie an bestimmten Entitäten im Netzwerk bzw. an ihrem Ziel wieder zusammengesetzt werden. Hier werden z.B. Laufzeitunterschiede wirksam, die bestimmte Mechanismen bei der Wahl des Pfades, am Ziel und innerhalb des Netzwerks voraussetzen.

3. *Skalierung innerhalb des Netzwerks:* Die Nutzung von Mehrwegprotokollen stellt für die Ressourcen des Netzwerks einen erhöhten Rechenaufwand dar. Sicherzustellen ist die Funktion des Netzwerks auch bei hohen Lastsituatio-

nen. Eingesetzte Technologien müssen den erhöhten Rechenaufwand leisten können und möglichst effektive Algorithmen einsetzen.

4. *Abrücken vom Paradigma des autonomen Systems:* Die Bereitstellung eines Netzwerks geschieht durch den NSP bzw. durch die Organisation und die Vermittlung, die enthaltene autonome Systeme durchführen. Eine Ende-zu-Ende-Verbindung geschieht unter Nutzung mehrerer Netzwerke, die alle verschiedenen Regeln unterliegen können. Einem Endbenutzer ist es daher so gut wie unmöglich, eine Verbindung über verschiedene Netzwerke zu planen und dabei eine Mehrwegnutzung mit bestimmten Metriken zu forcieren. Techniken müssen entworfen werden, die dem Endbenutzer Werkzeuge bieten, einen vorbestimmten Pfad nutzen zu können.
5. *Datensicherheit* bei der Übertragung mehrerer paralleler Datenströme.

Das in dieser Arbeit durchgeführte Projekt konzentriert sich vor allem auf den zweiten Themenschwerpunkt – Probleme der Datenintegrität, die durch eine Mehrwegübertragung, in diesem Falle auf Ende-zu-Ende-Ebene, entstehen können. Dies wird gestützt durch Techniken aus dem ersten Themenbereich, der Ausnutzung von Ressourcen innerhalb des Netzwerks. Zum Teil werden Bereiche des vierten Themenbereichs angesprochen, die eine Nutzung unabhängiger Pfade diskutieren.

5.1.1 Multipath Routing

Das *Multipath-Routing* erlaubt den Aufbau und die Nutzung multipler Pfade für das Routen auf der Vermittlungsebene in einem Netzwerk zwischen einer Quelle und einem Ziel [Tsai & Moors, 2006]. Multipath-Routing-Protokolle werden meist aufgrund eigener Ziele[32] durch den NSP installiert. In der Regel ist es einem Endbenutzer nicht möglich, auf diese Techniken zuzugreifen oder sie forciert zu nutzen. Die Aufgaben von Multipath-Routingprotokollen erstrecken sich über die drei folgenden Bereiche [Tsai & Moors, 2006]:

- Pfadentdeckung
- Lastverteilung
- Pfadaufrechterhaltung

Die *Pfadentdeckung* beschreibt den Prozess, bei dem aus den verfügbaren Pfaden zwischen einer Quelle und einer Senke nach unterschiedlichen Kriterien wie Dienstgüte oder NSP-Regelungen die Besten ausgewählt werden [Tsai & Moors, 2006]. Hierdurch wird ebenfalls der Grad der PD bestimmt, der die statistische Unabhängigkeit festlegt.

Bei Strategien der *Lastverteilung* wird festgelegt, inwiefern die unterschiedlichen Pfade zwischen Quelle und Ziel durch die Daten ausgenutzt werden [Tsai & Moors, 2006]. So ist es z.B. möglich, nur einen einzelnen Hauptpfad zu nutzen und weitere nur für den Störungsfall als Absicherung vorzuhalten (ebd.). Andere beinhalten eine Vertei-

[32] Vgl. Ausführungen oben

lungslogik oder erlauben die gleichzeitige Nutzung von Pfaden (ebd.). Es müssen Techniken genutzt werden, die den Datenstrom am Ziel wieder vereinen.

Warteschlangen helfen bei der Kompensation von Pfaden mit heterogenen Dienstgüteeigenschaften, bei denen Out-of-Order-Datenpakete, d.h. Datenpakete, die nicht in der richtigen Reihenfolge ankommen, anfallen können. Die Lastverteilung bestimmt über die oben genannten Vorteile, die je nach Strategie genutzt werden können. Die Qualität der genutzten Strategien hängt von den Dienstgütemetriken, der Granularität der Verteilung und der Anzahl der Pfade ab [Tsai & Moors, 2006].

Die *Pfadaufrechterhaltung* ist notwendig, um die gewählte Strategie der Lastverteilung durchgängig verfolgen zu können, da die Pfade über die Zeit variieren [Tsai & Moors, 2006]. Die Qualität der Pfade verändert sich im Laufe der Zeit durch etwaige Routenänderungen, Pfadausfälle oder Änderungen im Datenverkehr. Diese Veränderungen müssen durch Überwachungsstrategien rechtzeitig erkannt und mit ausgleichenden Mechanismen kompensiert werden [Tsai & Moors, 2006]. Die Mechanismen enthalten auch Methoden, um Datenpakete, die durch die Veränderungen verloren gegangen sind, wiederherzustellen.

Untersuchungen im Bereich von Multipath-Routing werden bereits seit einiger Zeit betrieben. Eine erste Erwähnung einer vereinfachten Multipath-Routing-Technik erfolgte 1975 durch [Maxemchuk, 1975], bei der Daten verteilt durch ein Netzwerk geleitet werden. Hierfür wurden effizienzbetreffende Vorteile in Hinblick auf die verbesserte Ressourcen-Nutzung und eine für eine Anwendung vorteilhaftere Weiterleitung ausgemacht.

Vorteile für die Qualität von Echtzeitsendungen im Bereich der Sprachübertragung (VoIP) über ein Netzwerk wurden in [Bettermann & Rong, 2011] untersucht. Es wurde ein spezieller Routing-Algorithmus genutzt, bei dem unabhängige Pfade innerhalb einer Simulationsumgebung zu einer Verbesserung der VoIP-Qualität führten.

Das Shim6-Protokoll [Nordmark & Bagnulo, 06/2009] wurde entworfen, um Multihoming zur Ausfallsicherung und Lastverteilung unter IPv6 zu ermöglichen. Es nutzt eine Zwischenschicht innerhalb der Vermittlungsschicht, die oberhalb der Routingfunktionen und unterhalb der IP-Endpunkt-Subschicht fungiert [Launois & Bagnulo, 2006]. Die Nutzung von Shim6 sieht vor, dass ein Host in einem Multihoming-Netzwerk, das Anschlüsse zu verschiedenen NSPs besitzt, den Anschluss des Netzwerks, über den der Datenstrom geht, frei wählen kann [Barré et al., 2011]. Hierbei besitzt ein Host mehrere IPv6-Adressen, die mittels ihres Präfixes je einem angeschlossenen NSP zugewiesen sind (ebd.). Ein Störungsdetektionsalgorithmus ist in der Lage, kombinierte Verbindungen zu erkennen und den Verkehr unter Nutzung des nächsten NSPs umzuleiten (ebd.). Dabei bleibt für die darüber gelegene Transportschicht weiterhin nur eine IP-Adresse für Quelle und Ziel sichtbar [Launois & Bagnulo, 2006].

In [Aubry et al., 2015] wird eine Architektur vorgestellt, die auf Router-Level und unter Bereitstellung von IPv6-Gateways innerhalb eines AS eine Duplizierung der Pakete und Routing über mehrere Pfade ermöglicht. Hierfür müssen unabhängige Pfade be-

rechnet und darüber eine Verteilung vorgenommen werden (ebd.). Zu diesem Zweck werden Netzwerksegmente berechnet und mithilfe eines veränderten IPv6-Headers abschnittsweise zugestellt (ebd.). Diese Technik ermöglicht es für NSPs, einen Duplizierungsservice anzubieten, der eine höhere Zuverlässigkeit mit geringerer Latenz verspricht. Hierüber lassen sich auf Vermittlungsebene mit vermehrten Datenübertragungskosten stabilere Verbindungen erstellen.

Das Equal Cost Multipath Protocol (ECMP) ist ein Routing-Protokoll, bei dem kompatible Router über mehrere Routingtabellen verfügen und somit an jedem Knoten mehrere „bestmögliche“ Pfade für ein Ziel zur Verfügung stehen [Hopps, 11/2000]. Mithilfe von ECMP wird entschieden, zu welchem nächsten Abschnitt ein Datenpaket weitergeleitet wird. Dies kann zu Zwecken der Lastverteilung angewandt werden und ermöglicht damit höhere Datendurchsatzraten.

ECMP kann in Kombination mit anderen Routingprotokollen innerhalb eines Netzwerks oder einer Netzwerkentität verwendet werden (ebd.). Probleme gibt es bei diesem Protokoll vor allem bei der Entscheidung, welchen Pfad ein Datenpaket verwenden soll. So kann es bei unterschiedlich gearteten Datenströmen zu rapiden Änderungen des Datenverkehrs innerhalb des Netzwerks und seinen Eigenschaften führen [Hopps, 11/2000]. ECMP wird daher meist nur beschränkt für bestimmte Multicast-Datensendungen in LAN oder Metropolitan Area Netzwerken (MAN) von Datenzentren eingesetzt [IEEE, 2016].

Eine Verbesserung der Eigenschaften von ECMP bieten [Xu & Li, 2014] mit einer Anwendung, die für bestimmte Datenflüsse eine neue Transportverbindung erstellt und die Datensegmente dupliziert übermittelt. Damit nutzt der duplizierte Datenstrom mit einer gewissen Wahrscheinlichkeit einen anderen statistisch unabhängigen Pfad im Falle von Überlastungen, die durch Massendatenströme hervorgerufen wurden (ebd.). Die Technik bietet eine Verbesserung vor allem für kurzzeitige Datenströme und wird innerhalb der Anwendungsschicht implementiert, sodass keine Änderungen auf Netzwerkebene durchgeführt werden müssen. In Verbindung mit ECMP kann diese eingesetzt werden, bietet aber im Internet kaum Vorteile, da hier kein flussbasiertes Routen stattfindet und sich die duplizierten TCP-Datenströme die Datenübertragungsrate eines Anschlusses teilen.

Weitere Forschungen beziehen sich auf die Verbesserung eines Überlastkontrollmechanismus auf Vermittlungsebene mithilfe des *Max Flow Multipath Protokolls* [Mahlous et al., 2009]. Sie besteht darin, die Weiterleitung in einem mehrwegfähigen Router soweit zu optimieren, dass Datenströme nur auf Pfade geleitet werden, die einen bestimmten Schwellwert der bereits ausgenutzten Datenübertragungsrate unterschreiten (ebd.). Diese Vorgehensweise bietet Vorteile im Vergleich zu ECMP (ebd.): Eine Schwellwertoptimierung der verwendeten Datenpfade ist sinnvoll, um Entscheidungen bei der Verteilung der Daten auf verschiedene Pfade zu vereinfachen. Dies ermöglicht eine gute Anpassung an die Situation im Netzwerk.

Die hier beschriebenen Protokolle stellen nur einen Ausschnitt der tatsächlich entwickelten Protokolle und Techniken dar. Eine ebenfalls aktuelle Übersicht bietet die oben erwähnte Quelle [Qadir et al., 2015]. Auf eine Beschreibung aller Besonderheiten wird hier verzichtet, da es in dieser Arbeit vor allem um Mehrwegtechniken auf Ende-zu-Ende-Ebene geht.

Eine Ende-zu-Ende-Verbindung bietet keine Kontrolle über die tatsächlich verwendeten Pfade. Diese kann nur über Vereinbarungen mit dem NSP, Multihoming mithilfe mehrerer NSPs sowie Steuerungsmöglichkeiten, die durch den NSP angeboten werden, erreicht werden. Auf der Strecke zwischen UDE und UKM wurden unabhängige Pfade durch die Nutzung verschiedener NSPs identifiziert.

5.1.2 Mehrweg-Protokolle auf Ende-zu-Ende-Ebene

Die Arbeit eines Mehrwegprotokolls auf Ende-zu-Ende-Ebene besteht im Allgemeinen darin, die Daten auf die verfügbaren Anschlüsse zu verteilen und am Ziel wieder zusammenzufügen. Das Stream Control Transmission Protokoll (SCTP) [Ong & Yoakum, 5/2002] ist ein verbindungsorientiertes Netzwerkprotokoll der Transportschicht, welches in der Lage ist, mehrere Datenverbindungen zu nutzen. In den letzten zehn Jahren sind für SCTP mehrere Modifikationen [Dreibholz et al., 2011] entwickelt worden, die speziell auf diese Möglichkeit eingehen:

- z.B. cmpSCTP [Liao et al., 2008], welches den Transport von Echtzeit-Kommunikationsdaten über kabellose Netzwerke verbessert;
- z.B. W-PR-SCTP [Fiore et al., 2007], welches die Funktion der partiellen Zuverlässigkeit von SCTP verwendet, um die Datendurchsatzrate für Multimedia-Datenströme zu verbessern.

SCTP wurde z.B. für Server-Umgebungen entwickelt mit dem Ziel, die Verbindungsleistung zwischen den Servern zu verbessern und multiple Routen zu ermöglichen. Die großen Unterschiede zu TCP und UDP erschweren es SCTP im Internet eingesetzt zu werden. Netzwerkentitäten sind in der Regel nur mit einer bestimmten Auswahl an Protokollen kompatibel. Diese beschränken sich meist auf die bekannten, herkömmlich genutzten Regelwerke [Honda et al., 2011].

Eine Entwicklung, die ebenfalls auf Multihoming setzt und sich speziell mit der Übertragung von verlustbehafteten Echtzeitdatenströmen auseinandersetzt, ist das Multipath Real-Time Protocol (MPRTP). Es setzt oberhalb der Transportschicht auf und kann mehrere Verbindungen per Multihoming nutzen [Singh et al., 2013]. Es nutzt eine minimale Erweiterung des anwendungsorientierten Real-Time Protocols (RTP) mit einem eigenen Datenverteilungsalgorithmus und einer für mehrere Pfade optimierten Empfangswarteschlange für das Dejittering. Bei diesem Protokoll werden die Datendurchsatzraten der verschiedenen Anschlüsse aggregiert und mithilfe eines Monitorings auf Paketverluste und Datenübertragungsänderungen überwacht.

Erste Ansätze, die das TCP-Transportprotokoll für eine Multihoming-Umgebung nutzen, gibt es mit der Erweiterung mTCP [Zhang et al., 2004]. Dies ist ein Protokoll,

das höhere Datenübertragungsraten durch Aggregation erlaubt und Monitoring-Techniken für die Überwachung der einzelnen Pfade nutzt. Damit kann bei einem Pfadausfall ein Failover durchgeführt werden, der in wenigen Sekunden vollzogen wird [Zhang et al., 2004]. Außerdem verwendet es ein Overlay-Netzwerk. Dabei handelt es sich um ein virtuelles Netzwerk in den anwendungsorientierten Schichten unter Nutzung der gegebenen Netzwerkinfrastruktur. Es erweitert dabei die Funktionen und Möglichkeiten der gegebenen Infrastruktur.

Allerdings sind viele der hier genannten Protokolle nur schwer innerhalb des Internets zu implementieren. Die im Internet vorhandene Infrastruktur müsste modifiziert oder ausgetauscht werden, da sich in der Praxis viele Systeme nicht zwingend an das Ende-zu-Ende-Prinzip halten: Datenpakete, die nicht mit einem bestimmten erwarteten Muster korrelieren, werden gelöscht. Das SCTP-Protokoll bietet viele Vorteile gegenüber UDP und TCP. Es ist jedoch zurzeit nur für bestimmte Netzwerke mit entsprechender Infrastruktur verwendbar. Für die Verbesserung von Dienstgüte auf einer Strecke zwischen Malaysia und Deutschland über das Internet ist das SCTP-Protokoll daher ungeeignet und folglich kein möglicher Kandidat.

Das Multipath Transport Control Protocol (MPTCP) baut auf TCP auf und ist damit weitgehend mit den im Internet verwendeten Systemen kompatibel. Das 2011 entwickelte Protokoll wurde mit der Zielsetzung angefertigt, im Internet zu funktionieren, dabei trotz der Nutzung mehrerer Datenpfade fair zu anderen Datenströmen zu sein und Netzwerkressourcen möglichst gleichmäßig auszunutzen [Barré et al., 2011]. Eine Standardisierung seit 2013 ermöglicht die Erweiterung des Protokolls in verschiedenen Bereichen [Ford et al., 2013]. Die weiteren Funktionen des Protokolls werden in Kapitel 5.1.3 weiter erläutert.

Eine Studie, die die Fähigkeiten von MPTCP in Zusammenhang mit dem Reduzieren von Latenz in Cloud-basierten mobilen Anwendungen untersucht hat, hat Vorteile hinsichtlich einer verkürzten Übertragungszeit identifiziert [Grinnemo & Brunstrom, 2015]. Darin wurden mehrere Cloud-basierte Anwendungen getestet, die über verschiedene Pfade an den Client Daten versenden. Grundsätzlich ist es von Vorteil, MPTCP in diesem Zusammenhang einzusetzen, jedoch wurden auch Nachteile in bestimmten Situationen nachgewiesen: Eine starke Heterogenität der Pfadeigenschaften hinsichtlich der Latenzzeit könnte den Leistungsgewinn schmälern und führt zu einer starken Erhöhung der Latenz sowie zu einem erhöhten Segmentverlust, welche um einiges größer ausfallen können, als es ohne Mehrwegnutzung der Fall wäre. Die Studie hat ergeben, dass bestimmte Modifikationen für MPTCP bei der Sendung von Datensegmenten auf verschiedenen Pfaden empfehlenswert wären. Die Empfehlungen beziehen sich vor allem auf eine veränderte Behandlung heterogener Pfadunterschiede. Der Standard-Scheduler von MPTCP sieht vor, dass der Pfad mit der kürzesten RTT zuerst verwendet wird, und zwar so lange, bis dieser voll ausgenutzt ist. Danach erst werden andere Pfade verwendet, um die Daten zu senden.

5.1.3 Grundlagen Multipath-TCP

MPTCP [Ford et al., 2013] wurde 2011 entwickelt, um für Geräte mit mehreren Netzwerkanschlüssen ein verwendbares zuverlässiges Transportprotokoll zur Verfügung zu stellen. Das Protokoll ist in der Lage, die Multihoming-Fähigkeiten eines Gerätes für Anwendungen nutzbar zu machen [Barré et al., 2011]. Vor allem wurden hierbei mobile Endgeräte betrachtet, die bereits von Anfang an über mehrere Netzanbindungstechnologien verfügen. Durch die Nutzung von MPTCP kann Anwendungen ein nahtloses Roaming zwischen verschiedenen Datennetzen ermöglicht werden, indem mehrere gleichzeitige Verbindungen über verschiedene Netzanbindungen aufrechterhalten werden. Durch Bündelung der Kapazitäten mehrerer Netzanbindungen lässt sich auch der Datendurchsatz erhöhen, indem mehrere Verbindungen gleichzeitig für den Datenverkehr benutzt werden [Barré et al., 2011].

MPTCP wurde mit der Zielsetzung entwickelt, im Internet eingesetzt zu werden. Ein Problem ist hier, dass verschiedene Middlebox-Geräte wie z.B. Network Address Translation (NAT) Gateways, Firewalls und Proxies nur Pakete bestimmter erwarteter Protokolle weiterleiten und diese sogar verändern oder zerteilen können [Honda et al., 2011]. MPTCP muss deshalb nicht nur eine Rückwärtskompatibilität mit TCP besitzen, es soll ebenfalls für die meisten Internetgeräte den Anschein einer regulären TCP-Übertragung erwecken. Diese und andere Aspekte wurden seit der ersten Standardisierung fortwährend verbessert [Paasch et al., 2013].

MPTCP wurde innerhalb der Transportschicht als eine übergeordnete Instanz von TCP und unterhalb der anwendungsorientierten Schichten implementiert. Damit nutzt es TCP und erweitert es um Funktionen, die für eine Mehrwegbenutzung nötig sind [Paasch, 2014]. Abbildung 5.1 zeigt die Gegenüberstellung von TCP und MPTCP innerhalb der ISO/OSI-Referenzmodellschichten drei bis fünf. Darin wird deutlich, dass MPTCP für einen sogenannten *Subflow* (Teildatenstrom) jeweils eine TCP-Verbindung nutzt und diese mithilfe einer semantischen Teilschicht bzw. der MPTCP-Instanz verwaltet [Ford et al., 03/2011]. Diese Teilschicht stellt ebenfalls die Schnittstelle für die Anwendung bereit.

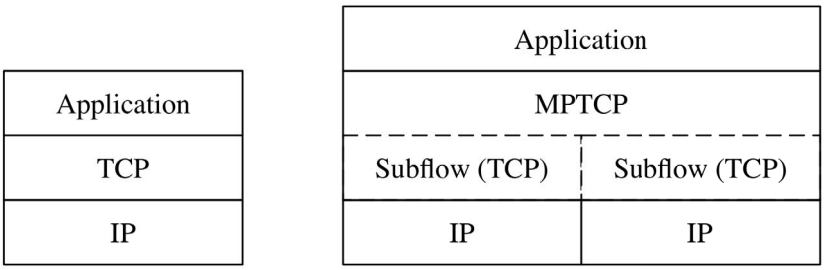

Abbildung 5.1: Vergleich des TCP und MPTCP Protokollstapels[Ford et al., 2013]

Die grundsätzliche Idee bei MPTCP ist es, mehrere einzelne TCP-Verbindungen zu verwenden, die jeweils einen Subflow bilden. Jeder dieser Subflows bildet eine eigene Ende-zu-Ende-Sitzung. Zu sendende Daten einer Applikation werden durch die MPTCP-Instanz auf die verschiedenen Subflows verteilt. Auf Empfängerseite werden

die empfangenen Daten aus den verschiedenen Subflows wieder zusammengeführt und als einheitlicher Datenfluss an die Applikation weitergegeben. [Ford et al., 03/2011]

Ein Endgerät kann so viele Subflows aufbauen, wie es IP-Adressen gibt. Ebenfalls möglich sind kreuzverbundene Subflows, sodass bei zwei Endgeräten mit z.B. jeweils zwei IP-Adressen eine Gesamtzahl von vier Subflows erstellt werden kann [Ford et al., 2013].

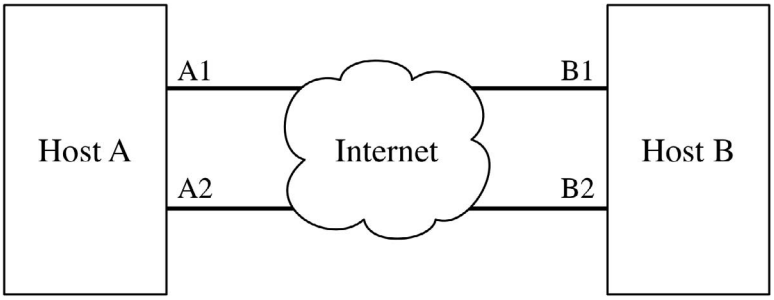

Abbildung 5.2: MPTCP Verbindung zwischen zwei Endgeräten mit mehreren Netzwerkschnittstellen[Ford et al., 2013]

MPTCP wird durch Funktionen innerhalb des TCP-Headers realisiert. Wie aus Kapitel 3.3.6.3 hervorgeht, gibt es im TCP-Header einen Freiraum für optionale Informationen. MPTCP nutzt diesen Freiraum für MPTCP-eigene Optionen, wie beispielsweise eine eigene *Datensequenznummer* und verschiedene Signalisierungen. Die meisten Middlebox-Geräte im Internet unterstützen die Verwendung zusätzlicher Optionen innerhalb dieses Header-Feldes [Raiciu et al., 2012].

Durch Nutzung des *Option*-Feldes im TCP-Header können MPTCP-Signalisierungen durchgeführt werden. Alle Signalisierungen werden innerhalb dieses Feldes abgebildet. Abbildung 5.3 zeigt den grundsätzlichen Aufbau des MPTCP-Headers innerhalb des Option-Feldes von TCP. Hierbei bilden die ersten 8 Bit den Typ der TCP-Option, die im Fall von MPTCP immer *30* ist. Danach folgt die Länge des MPTCP-Headers. Der *Subtype* gibt den Statusindikator für die MPTCP-Option an. Alle nachfolgenden Daten hängen von diesem Statusindikator ab.

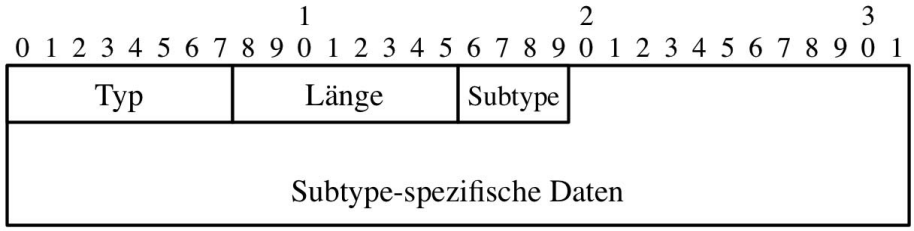

Abbildung 5.3: MPTCP Header als TCP Option[Ford et al., 2013]

Für die Übertragung von Daten auf mehreren Pfaden lassen sich zwei Hauptfunktionen definieren [Ford et al., 03/2011]:

- *Pfad-Management:* Hierdurch werden nutzbare Subflows erkannt und zwischen den Endgeräten aufgebaut. Das Pfad-Management ist dafür verantwortlich, die Subflows zu verwalten.
- *Daten-Scheduling:* Daten, die von einer Anwendung gesendet werden, werden durch den Scheduler auf die verfügbaren Subflows verteilt. Wie diese Verteilung durchgeführt wird, entscheidet der Scheduling-Algorithmus. Die Verteilung kann über ein Round-Robin-Verfahren stattfinden, es können aber auch andere Möglichkeiten wie das Priorisieren eines einzelnen Pfades genutzt werden.

Verschiedene Aspekte von Schedulern wurden in [Paasch et al., 2014] und [Adhari et al., 2011] näher erörtert, wobei das Sende- und Empfangsverhalten mehrerer Subflows unter bestimmen Umständen untersucht wurden. Probleme bilden hier Out-of-Order-Datensegmente [Yang et al., 2014], die durch die unterschiedliche Subflow-Nutzung sowie durch die heterogenen Eigenschaften von Pfaden entstehen können und so zu Verzögerungen führen. Bisher implementiert wurden Round-Robin und ein Pfad-Priorisierungs-Verfahren, bei dem der Subflow mit der kleinsten RTT verwendet wird, bis das Überlastungsfenster ausgefüllt ist, um dann mit dem nächsten Subflow fortzufahren. Das Pfad-Priorisierungsverfahren bietet die bessere Alternative um Out-of-Order-Segmenten vorzubeugen, da die Daten nicht forciert verteilt versendet werden.

Anwendungen unter MPTCP ist es erlaubt, bestimmte Parameter von MPTCP selbst zu steuern. Diese *Multipath-TCP-aware*-Applikationen bieten die Möglichkeit, MPTCP in Hinblick auf die eigenen Funktionen einzusetzen, indem die programmierbare Schnittstelle (*Application Programmable Interface, API*) von MPTCP genutzt wird [Ford et al., 03/2011]. Eine API für die vollständige Kontrolle des Verhaltens und des Pfad-Managements unter MPTCP ist in Arbeit und Konzepte befinden sich in der Entwicklung [Hesmans & Bonaventure, 2016] [Hesmans et al., 2015]. Diese können bereits zum Teil genutzt werden und sollen auch in dieser Arbeit eingesetzt werden, um eigene API-Implementationen nutzen zu können.

Multipath-TCP-unaware-Applikationen sind Anwendungen, die keine Handhabe bieten, die speziellen Funktionen von MPTCP zu nutzen. Die Benutzung von MPTCP ist dann für die Anwendung und den Benutzer transparent[33]. Die Nutzung mehrerer Subflows kann erfolgen, ohne dass die Anwendung Kenntnisse von der darunterliegenden MPTCP-Struktur besitzt, wenn die Gegenstelle ebenfalls MPTCP-kompatibel ist und über mehrere Endpunkte verfügt [Ford et al., 03/2011]. Hierbei werden alle potenziell möglichen Subflows erstellt, indem ein *Full-Mesh-Pfadmanager* eingesetzt wird.

Abbildung 5.4 zeigt eine Darstellung des MPTCP Protokoll-Stapels, in dem Applikationen über das Socket-Interface auf MPTCP zugreifen. Eine Anwendung verwendet hierbei dasselbe Socket-Interface, wie für eine reguläre TCP Anwendung.

[33] im Sinne von „nicht sichtbar verwendbar“

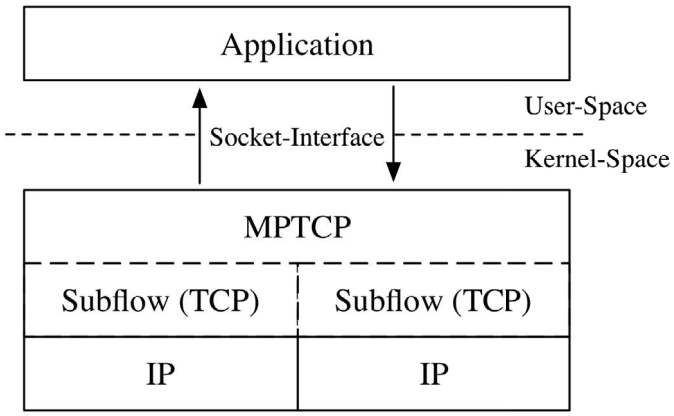

Abbildung 5.4: Socket API für Anwendungen und Datenverteilung auf MPTCP-Subflows[Scharf & Ford, 03/2013]

Bei der Sequenzierung der Segmente für eine zuverlässige Datenübertragung ist es notwendig, dass nicht nur TCP-Segmente eines einzelnen Subflows (*Subflow-Level*) nummeriert werden [Ford et al., 03/2011], sondern auch die Segmente der darüber liegenden MPTCP-Schicht (*Data-Level*) [Ford et al., 2013]. Auf diese Weise können die Segmente wieder zu einem vollständigen Datenstrom auf Empfängerseite zusammengebaut werden, obwohl dieser vorher auf mehrere Verbindungen verteilt gesendet wurde. Bei der Verwaltung der Subflows werden die TCP-Sequenznummern (*Subflow-Sequenznummer*) mit den MPTCP-Sequenznummern (*Datensequenznummer*) verbunden.

Ein Verbindungsaufbau unter MPTCP ist mit dem 3-Wege-Handshake unter TCP vergleichbar [Ford et al., 2013]. MPTCP sendet hierbei innerhalb eines SYN-Segments den zusätzlichen Statusindikator MP_CAPABLE. Ein MPTCP-kompatibler Server beantwortet die Anfrage mit einem SYN/ACK-Segment, welches ebenfalls den Statusindikator MP_CAPABLE gesetzt hat. Ein ACK-Segment vom Client vervollständigt den Handshake. Wird MPTCP vom antwortenden Gerät nicht unterstützt, so wird das Segment entweder verworfen oder ohne die MP_CAPABLE-Option gesendet. In diesem Fall wird für die nächsten Operationen reguläres TCP verwendet [Ford et al., 2013]. Das Beendigen eines Subflows erfolgt in ähnlicher Weise durch die entsprechenden MPTCP-Optionen.

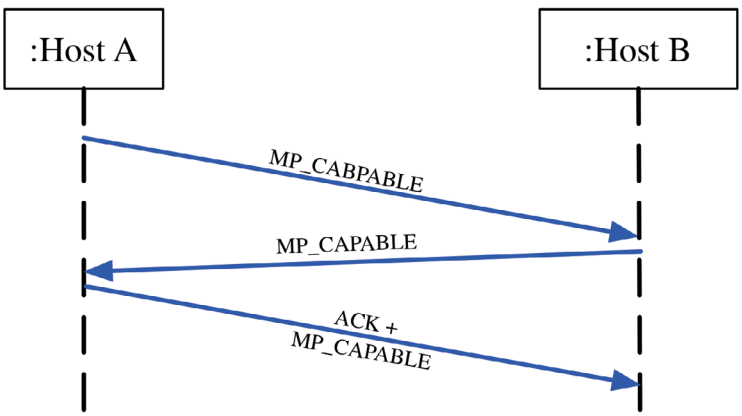

Abbildung 5.5: 3-Wege-Handshake Optionen für MPTCP

Um MPTCP vollständig nutzen zu können, werden zusätzliche Subflows benötigt. Das Pfad-Management identifiziert zusätzliche Pfade durch das Vorhandensein mehrerer IP-Adressen [Ford et al., 2013]. Jedes Endgerät ist in der Lage, neue Pfade zu initiieren (ebd.). Können weitere Subflows erstellt werden, werden diese mithilfe eines 4-Wege-Handshakes – ähnlich dem Initial-Handshake – aufgebaut. Zusätzliche Absicherungen sorgen dafür, dass die zusätzliche IP-Adresse des neuen Subflows vom selben Endgerät stammt [Ford et al., 2013]. Dies wird mithilfe eines *Hash Message Authentication Codes (HMAC)* sichergestellt.

Um Daten zu übermitteln, werden sie zunächst im MPTCP-Hauptpuffer gespeichert. Der Scheduler gibt die Datensegmente an den Subflow-Puffer weiter [Ford et al., 2013]. Der Subflow sendet das Segment mit der Datensequenznummer und der Subflow-Sequenznummer zur Gegenstelle (ebd.).

Nach Empfang übergibt der Subflow die Datensegmente in der Reihenfolge der Subflow-Sequenznummern an die MPTCP-Instanz. Diese überprüft, ob das Segment bereits auf einem anderen Subflow empfangen wurde (ebd.). Wurde das Datensegment vorher noch nicht empfangen, wird es in den MPTCP-Hauptpuffer eingespeist. Von dort aus werden die Daten mithilfe der Datensequenznummer in der richtigen Reihenfolge an die Applikation übergeben. Im Falle von verlorengegangenen Datensegmenten werden die Daten, wie bei TCP durch ein fehlendes ACK-Segment erkannt und daraufhin erneut gesendet. Mithilfe von MPTCP-ACK-Segmenten können Datensegmente auf anderen Subflows erneut gesendet werden. Für eine Applikation ist dieser Vorgang transparent. Kann keine Verbindung über MPTCP geschlossen werden, fällt die Verbindung auf herkömmliches TCP zurück. Nach der Herstellung einer MPTCP-Verbindung wird der *Meta-Socket* genutzt, der die Subflows verwaltet. Der Scheduler ist dafür verantwortlich, die Daten auf die Subflows zu verteilen. Umgekehrt senden die Subflows empfangene Daten an den Meta-Socket.

Eine Pufferung gesendeter Datensegmente auf Subflow-Ebene wird durch die in Kapitel 3.3.6.3 beschriebene Vorgehensweise durchgeführt. Eine weitere Pufferung geschieht auf Datenebene (Meta-Socket), um verlorene Daten mit den anderen Subflows abgleichen zu können. Dies ist wichtig, wenn ein Subflow ausfällt und die darauf transportierten Daten verloren sind. Der schlimmste Fall wäre hierbei der Verlust des Subflows mit der größten Verzögerungszeit, da hier die meisten Daten vorliegen [Ford et al., 03/2011]. Ein Empfänger muss die Daten sämtlicher Subflows solange speichern, bis die verlorenen Datensegmente empfangen wurden. Hieraus lässt sich die folgende Formel für das Empfangsfenster der gemeinsamen Flusskontrolle definieren (ebd.):

$$Max\ Receive\_window\ =\ 2\ \cdot\ sum(BW\_i)\ \cdot\ RTT\_max \qquad (5.1)$$

wobei BW_i die gesamte Datenübertragungsrate aller Subflows beschreibt und RTT_max die höchste RTT aller Subflows definiert. Diese Formel ist ebenfalls für den gemeinsamen Sendepuffer auf Datenebene gültig.

Der in Kapitel 3.3.6.3 erläuterte Überlastkontrollmechanismus von TCP wird unter MPTCP weiterhin durch die Subflows verwendet. In MPTCP gibt es Bestrebungen, eine

gekoppelte Überlastkontrolle zu entwickeln, mit der mehrere Subflows so zu benutzen sind, dass sie aufeinander abgestimmt ihren Datenverkehr verteilen können – ohne mit anderen Datenströmen gemeinsam genutzte Datenleitungen unfair auszunutzen [Raiciu et al., 2012]. Das Ziel ist es, die Aggressivität mehrerer genutzter Subflows auf dasselbe Niveau von TCP herunterzubringen. Ein Vorteil ist es, dass bei dieser Vorgehensweise mehr Daten über weniger überfüllte Pfade geleitet werden und sich somit der gesamte Netzwerkverkehr gleichmäßiger innerhalb des Netzwerks verteilt [Wischik et al., 2008]. Forschungen im Bereich der gekoppelten Überlastkontrolle richten sich sowohl auf die Effizienz bei der Erkennung als auch auf das Verhalten von überlasteten Subflows [Peng, et al., 2013] [Wischik et al., 2011] [Nguyen & Nguyen, 2011].

Bei der momentan implementierten gekoppelten Überlastkontrolle nach [Raiciu et al., 10/2011] wird der von TCP verwendete Mechanismus bis auf die Anstiegsphase des *Congestion Avoidance Algorithmus* übernommen. Der Algorithmus bremst für jeden Subflow die Anstiegsphase insofern, als dass der gesamte MPTCP-Subflow nicht aggressiver sein darf, als ein TCP-Datenfluss unter den besten Umständen (ebd.).

In [Baidya & Prakash, 2014] stellten die Autoren Schwächen des standardisierten Überlastprotokolls von MPTCP bei stark heterogenen Pfaden fest. Die gesamte Datenübertragungsrate wird wegen des gekoppelten Mechanismus durch Pfade abgebremst, die unter starken Einschränkungen der Datenübertragungsrate leiden, je nachdem welcher Scheduling-Mechanismus angewandt wird (ebd.). Eine Verbesserung bringt ein Mechanismus, der in der Lage ist, schlechte Pfade zu erkennen und diese nur rudimentär zu behandeln, bis sich die Situation ändert (*Slow Path Adaption*) (ebd.). Die Autoren stellten die folgenden Faktoren auf, die die Gesamtleistung von MPTCP bestimmen (ebd.):

1. *Netzwerkgeschwindigkeit* als der maximale Wert der möglichen Datenübertragungsrate, die eine Pfadgeschwindigkeit beeinflusst
2. *Anzahl der Subflows*, die theoretisch durch Aggregation die Datendurchsatzrate erhöhen
3. *Überlast im Netzwerk*, die die Leistung eines Subflows auf die Netzwerksituation beschränkt
4. *Externe Interferenzen*, die eine Heterogenität der genutzten Pfade hervorruft und das Übertragungsverhalten des Subflows beeinflusst
5. *Übertragungsmenge*, die zwischen lang anhaltendem Verkehr und kurzem unterschieden werden kann

Die vorgeschlagene Methodik zielt darauf ab, Pfade zu identifizieren, die stark heterogen und schlechter als andere sind [Baidya & Prakash, 2014]. Dies gelingt mit der Beobachtung der RTT sowie mit der bisher übertragenen Datenmenge eines Subflows (ebd.). Der Subflow wird nur mit geringem Datenverkehr belastet, um eine schlechte Leistung der anderen Subflows zu vermeiden (ebd.). Der schlechte Subflow wird weiterhin verwendet, um eine Besserung der Situation durch stetige Sondierung zu ermög-

lichen (ebd.). Diese Methoden eignen sich für eine bessere Koordination und Lastverteilung der Subflows.

Die Erkennung von guten und schlechten Pfaden kann ebenfalls dazu genutzt werden, Subflows zu erstellen oder zu schließen. [Li et al., 2015] haben diese Methode eingesetzt, um die Datenübertragungsrate von MPTCP für die Verwendung von Datenübertragungen in Datenzentren zu optimieren und so nur die Subflows zu involvieren, die über eine ausreichende Leistung verfügen. Dies optimiert den Datenfluss bei unterschiedlich gearteten Datensendungen und erhöht die Datenübertragungsrate.

Die aktuelle Implementierung von MPTCP ist nach wie vor experimentell, obwohl sie bereits in einigen Anwendungen zum Einsatz kommt [IETF, 2016]. Die zum Zeitpunkt dieser Arbeit existierende Version ist v0.89 und basiert auf dem LTS Linux Kernel 4.1.x. Diese Version dient als Basis für die weitere Verwendung in dieser Arbeit.

5.2 Dienstgüteverbesserung durch zusätzliches Senden von redundanten Datenpaketen

Redundanz in der Informationstheorie bezeichnet allgemein „die Weitschweifigkeit einer Nachricht“ [Brockhaus Enzyklopädie Online, 2017]. Zeichen oder Symbole, die keinen zusätzlichen Informationsgehalt bedeuten, sind redundant und können dazu dienen, verlorengegangene Informationen zu rekonstruieren (ebd.). Im technischen Sinne bezeichnet Redundanz einen Mehraufwand, der für die Funktion eines Systems nicht direkt nötig ist (ebd.). In vielen Bereichen ist das Hinzufügen von Redundanz eine verbreitete Technik, um Störungen zu begegnen. So können redundant verbaute Elemente dazu genutzt werden, die Zuverlässigkeit der Informationsübertragung zu verbessern.

Eine Anwendung von Redundanz ist der Einsatz im Rahmen von Rechenoperationen zur Verkürzung der Rechenzeit [Vulmiri et al., 2012]. Mithilfe zusätzlicher Redundanz können Verzögerungen verringert werden. Hierbei werden in einem belasteten System Rechenoperationen mehrmals unter Verwendung unterschiedlicher Ressourcen initiiert. Es wird dann das Ergebnis der zuerst beendeten Operation verwendet. [Vulmiri et al., 2012] argumentieren, dass die Kosten für benötigte zusätzliche Ressourcen in bestimmten Fällen die Kosten für verspätete Operationen übersteigen. Dies kann vor allem bei Netzwerkoperationen der Fall sein, da dort die Kosten für zusätzlichen Datendurchsatz vergleichsweise niedrig sind. Im Falle von Netzwerkoperationen können selbst kleine Ausfälle die Kosten von zusätzlicher Datenübertragungsrate überwiegen.

Bei der folgenden Formel werden die applikationsspezifischen Latenzkosten den Kosten der zusätzlich benötigten Datenübertragungsrate gegenübergestellt [Vulmiri et al., 2012]. Es wird eine Grenze ermittelt, ab der sich der Aufwand für eine Latenzreduktion mit zusätzlichem Bedarf an Datenübertragungsrate lohnt:

$$l \cdot v \geq b \tag{5.2}$$

wobei *l* den durchschnittlichen Latenzgewinn in Millisekunden für jedes zusätzliche Kilobyte darstellt, *v* die Kosten für eine Millisekunde Latenzreduktion und *b* die Kosten für ein Kilobyte bei einem ISP [Vulmiri et al., 2012]. Diese Rechnung bezieht sich:

- auf die geschätzten Kosteneinsparungen durch den zeitlichen Gewinn einer Übertragung, welche stark applikationsabhängig ist,
- auf die Latenzzeit einer Übertragung selbst.

Eine beispielhafte Gegenüberstellung der Kosten bei amerikanischen ISPs und dem zeitlichen Gewinn für einen durchschnittlichen Arbeitslohn zeigte eine Grenze bei *10 ms* zusätzlichem zeitlichen Gewinn beim teuersten Anbieter auf, ab der sich der Kauf von zusätzlicher Datenübertragungsrate lohnen würde [Vulmiri et al., 2012]. Eine Übertragung dieses Beispiels auf das in der Arbeit verwendete Einsatzszenario käme zu einem weitaus höheren zeitlichen Gewinn, da die Kosten einer Tele-Operation mit dem darin involvierten Risikofaktor bei weitem höher liegen.

Redundante Datenpakete in Form von FEC in multimedialer Echtzeitübertragung wurden zuerst für die Verbesserung der Audioqualität eingesetzt [Hardman et al., 1995]. Hierbei werden zusätzliche synthetische Sprachinformationen in später gesendeten Paketen mitgeführt, die dazu genutzt werden, auftretende Lücken durch verlorene Datenpakete innerhalb der „echten“ Sprachdaten aufzubessern (ebd.). Dazu werden Sprachkodierungsalgorithmen genutzt, aus denen synthetische Informationen herausgezogen werden. Dies ermöglicht eine FEC ohne umfangreiche zusätzliche Daten. Durch die Nutzung von adaptiven Mechanismen, die sich an die Situation im Netzwerk anpassen, wird die Audioübertragung weiter hinsichtlich ihrer Fehleranfälligkeit verbessert. [Bolot et al., 1999] entwickelten eine Technik, die eine FEC an die zu empfangene Qualität und die variierenden Datenpaketverluste im Netzwerk anpasst.

Redundante Techniken zur Verbesserung von Audioübertragung setzen zunächst auf das Präparieren von Nachrichten. Hierbei wird die Natur des menschlichen Gehörs berücksichtigt. Dazu gehören:

- die Möglichkeit eines fehlerbehafteten Übertragens von Daten, ohne dass die empfangene Nachricht dadurch unbrauchbar würde
- eine gewisse Toleranz bei der Verzögerung und Pufferung von Daten
- die Möglichkeit für das Hinzufügen von interpolierten Informationen

Die genannten Techniken werden aber weiterhin durch die in Kapitel 4.3 beschriebene Fehlerkorrelation bei hintereinander gesendeten Datenpaketen in ihrer Wirksamkeit geschmälert. Eine Chance, diese Probleme zumindest teilweise zu beheben, besteht darin, mehrere redundante Datenpakete einer Sprachübermittlung auf unterschiedlichen Netzwerkpfaden zu übertragen [Liang et al., 2001]. Damit wird unterbunden, dass Effekte, die einen bestimmten Netzwerkpfad betreffen, statistisch abhängige Auswirkungen auf die redundant gesendeten Daten der anderen Pfade haben. In Kombination mit FEC, deren Kodierungsstruktur auf die unterschiedlichen Pfade verteilt wird, und der Nutzung einer Kostenfunktion, die Verzögerungen gegenüber Datenpaketverlust und

Sprachqualität bemisst, kann eine höhere Sprachqualität erzielt werden [Liang et al., 2001]. Diese Technik zeigt eine bessere Leistung als eine FEC über einen einzelnen Pfad.

Eine FEC, die in Zusammenhang mit Mehrwegtechniken eingesetzt wurde, wird in [Ao et al., 2012] vorgestellt. Darin haben die Autoren ein auf Kodierung basierendes Schema für Wireless-Netzwerke entwickelt, um die Übertragungsverzögerung von Datenpaketen über verschiedene Pfade zu optimieren und die Zuverlässigkeit zu erhöhen. Das Schema erlaubt eine Codierung unter Einbeziehung verschiedener Prioritätsstufen und Pfadeigenschaften, um den Overhead einer FEC gegenüber der Zuverlässigkeit so zu justieren, dass die Daten auf verschiedene Pfade aufgeteilt werden können (ebd.). Dies setzt jedoch umfangreiche Berechnungen voraus, die sowohl am Sender als auch am Empfänger implementiert werden müssen. Das Aufteilen und Zusammenfügen von Daten an beiden Endpunkten führt zu neuen Schwierigkeiten, die für eine möglichst unverzögerte Zustellung problematisch sind.

Redundante Techniken werden auch für den industriellen Einsatz verwendet. Anwendungen der harten Echtzeit können dort von redundanten Netzwerktopologien in den unteren Netzwerkschichten profitieren. Das *Parallel Redundancy Protocol* beschreibt den Einsatz eines Ethernet-Systems in redundanter Nutzung [Kirrmann et al., 2007]. Hierbei besitzen Netzwerkgeräte zwei Ethernet-Anschlüsse, die an jeweils eigene Netzwerke mit gleicher Topologie angeschlossen sind. Die Daten werden auf beiden Netzwerken redundant gesendet. So können harte Echtzeitanwendungen über kleinere Netzwerkstrukturen gefahren werden. Ausfälle durch verzögerungsfreies Übernehmen (*Takeover*) lassen sich vermeiden sowie Verzögerungen berechenbar machen.

Mithilfe von Redundanz und gezielten Strategien in Hinblick auf die Kodierung der gesendeten Daten lassen sich signifikante Verbesserungen vor allem bei verlustbehafteten Datenübertragungen wie Video und Audio erzielen. Aber auch progressiv genutzte Redundanz kann sich positiv auf eine Datenübertragung im Allgemeinen – auch im verlustfreien Echtzeitbereich – auswirken. Zusätzliche Verbesserungen bieten Mehrwegtechniken, bei denen redundante Datenpakete über verschiedene Pfade gesendet werden.

5.3 Dienstgüteverbesserungen durch redundante Datensegmente in Verbindung mit Multipath-TCP

Die oben genannten Methoden beschreiben mögliche Techniken, eine verbesserte Dienstgüte zu erreichen. Reguläre Maßnahmen, die eine Bevorzugung innerhalb des Best-Effort-Ansatzes ermöglichen, sind auf Netzwerkebene zu implementieren. Ein Benutzer kann dies nur unter hohem Aufwand und mit zusätzlichen Kosten – durch Einkaufen von Priorisierungen auf allen Teilstrecken liegenden NSPs – erreichen. Bisherige Ende-zu-Ende-Maßnahmen führen nicht weit genug, um Störungsquellen zu minimieren. Mit zusätzlicher Redundanz lassen sich Datenpaketverluste und damit Verzögerungszeiten ausgleichen. Problematisch ist hierbei die statistische Abhängigkeit der Datenpaketverluste.

In Kapitel 4.4 wurde bereits eine mögliche Lösung durch die Verwendung mehrerer gleichzeitiger Verbindungen vorgestellt. Die unterschiedlichen Verbindungen können unter der Verwendung verschiedener ISPs weitestgehend heterogen und damit statistisch unabhängig voneinander geschlossen werden. Es kann die folgende Hypothese aufgestellt werden, die in dieser Arbeit die Problembearbeitung darstellt:

Unter gleichzeitiger Verwendung mehrerer (statistisch unabhängiger) Verbindungen ist es möglich, Varianzen in Latenzzeiten zu glätten, Paketverluste zu minimieren sowie Ausfälle der Verbindungen auszugleichen und damit die Zuverlässigkeit einer Verbindung zu erhöhen.

Mehrweg-Transportprotokolle bieten die Möglichkeit, eine Datenverbindung auf Ende-zu-Ende-Ebene aufzubauen und Daten auf mehreren Anschlüssen (Multihoming) zu senden. Bisherige Protokolle können nicht zufriedenstellend im Internet verwendet werden. Dies gilt im besonderem Maße auch für eine Strecke zwischen Malaysia und Deutschland, bei der zahlreiche Netzwerke und Knotenpunkte durchlaufen werden.

Der Multipath-TCP (MPTCP) Standard [Ford et al., 2013] bietet die Möglichkeit, eine internetkompatible Erweiterung für ein redundantes Senden von Datensegmenten über mehrere Datenverbindungen in der Transportschicht zu entwickeln. Dementsprechend lässt sich eine zweite Hypothese aufstellen:

Anwendungen mit speziellen Dienstgüteanforderungen können eine redundante MPTCP-Erweiterung nutzen und hierdurch Vorteile hinsichtlich einer zuverlässigeren Verbindung und einer stabilisierten Verbindungslatenz im Internet erzielen.

Die Transportschicht bietet folgende Vorteile:

1. Ende-zu-Ende-Funktionen, die im Gegensatz zu Hop-zu-Hop-Funktionen zwischen den Endpunkten eingesetzt werden können. Die Aushandlung sowie Abwicklung des Sendens und Empfangens bleibt hierbei den Endpunkten überlassen und gilt über die gesamte Verbindung. Dies gilt ebenfalls für die Anpassung auf die Situation im Netzwerk.
2. Die Behandlung von zusätzlich gesendeten Daten auf Anwendungsebene würde ein neues Protokoll erfordern, auf welches eine Anwendung modifiziert werden muss. Die Kontrolle unterschiedlicher Pfade und deren Koordination ist schwierig.

Eine Möglichkeit ist es, ein Redundanz-Schema auszuarbeiten, welches es bei zusätzlichen Kosten hinsichtlich der Datenübertragungsrate ermöglicht, dieselben Daten auf verschiedenen Verbindungen redundant zu senden. Der zu entwickelnde Paket-Scheduler repliziert alle von der Applikation hereinkommenden Daten und verteilt diese auf verschiedene Datenverbindungen. Das Ziel ist es, Latenzspitzen zu glätten und eine Ausfallsicherung für die Daten von möglichst geringer Zeit zu bieten. Eine erste Erwähnung einer solchen Erweiterung wurde in [Scharf & Ford, 03/2013] geboten. Darin heißt es unter anderem, dass eine Erweiterung für latenzkritische Anwendungen sinnvoll erscheint.

Die grundlegende Funktionsweise besteht darin, Daten, die auf einer Verbindung gesendet werden sollen, zu replizieren und diese zugleich über andere mögliche Verbindungen zu senden. Datensegmente, die auf einer Verbindung verlorengehen, werden durch Datensegmente der anderen Verbindungen ausgeglichen – so muss nicht auf eine wiederholte Sendung gewartet werden. Auf Empfängerseite wird stets dasjenige Datensegment angenommen, welches zuerst ankommt. Weitere eintreffende Duplikate werden verworfen. Hierbei ist es wichtig zu verstehen, welche Abhängigkeiten bei diesem Schema bestehen und welche Probleme seine Nutzung mit sich bringen kann. Diese Aspekte sollen in dieser Arbeit untersucht werden.

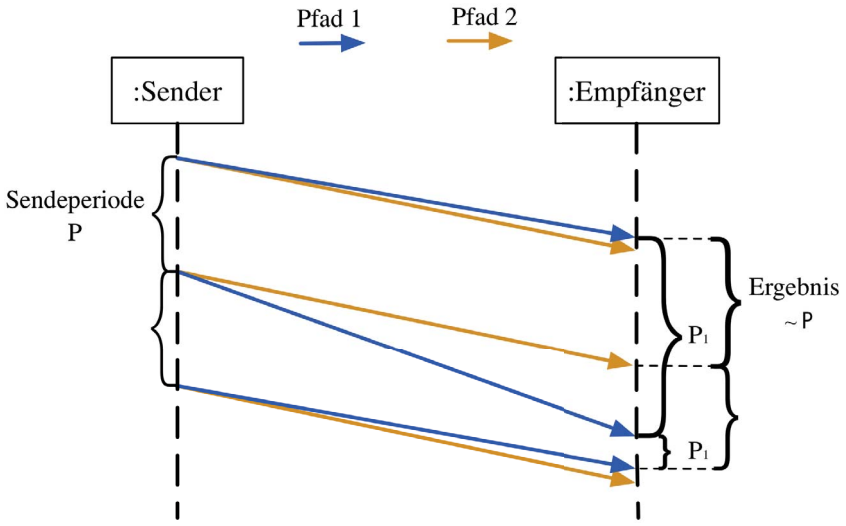

Abbildung 5.6: Redundantes MPTCP-Schema bei Latenzvariationen

Abbildung 5.6 zeigt ein vereinfachtes Schema einer redundanten Verbindung mit zwei genutzten Pfaden. Datensegmente werden auf allen verfügbaren Pfaden verdoppelt gesendet. Das zweite Datensegment auf Pfad 1 unterliegt einer Störung, die z.B. durch einen Segmentverlust hervorgerufen wird. Der Empfänger nimmt stattdessen das auf Pfad 2 ankommende Datensegment an. Die Verwendung des einzelnen Pfades 1 würde hingegen einen Latenzunterschied in der Applikation auslösen. Eine drastischere Störung, die ein weitaus späteres Eintreffen zur Folge hätte, würde die Annahme der weiteren Segmente sogar verhindern. Das Annehmen des Datensegments auf dem anderen Pfad verhindert einen größeren Einbruch der Segmentperiode mit einer geringen Erhöhung von Jitter durch den Laufzeitunterschied von Pfad 1 und Pfad 2.

In Abbildung 5.7 ist ein fiktives Beispiel eines Sendeverlaufs in Form eines Liniendiagramms dargestellt. Die Vertikalachse stellt hierbei die resultierende Verzögerung des Netzwerks dar. Die zwei Pfade A und B besitzen unterschiedliche Verzögerungen. Grundsätzlich besitzt Pfad A eine kürzere Verzögerung, unterliegt aber an einigen Stellen einer Verschlechterung der Qualität. Diese macht sich in zwei Spitzen bemerkbar, die die Verzögerung von Pfad B überragen. Das Ergebnis auf Empfangsseite wird durch die rote Linie dargestellt, aus der ein geglättetes Ergebnis resultiert.

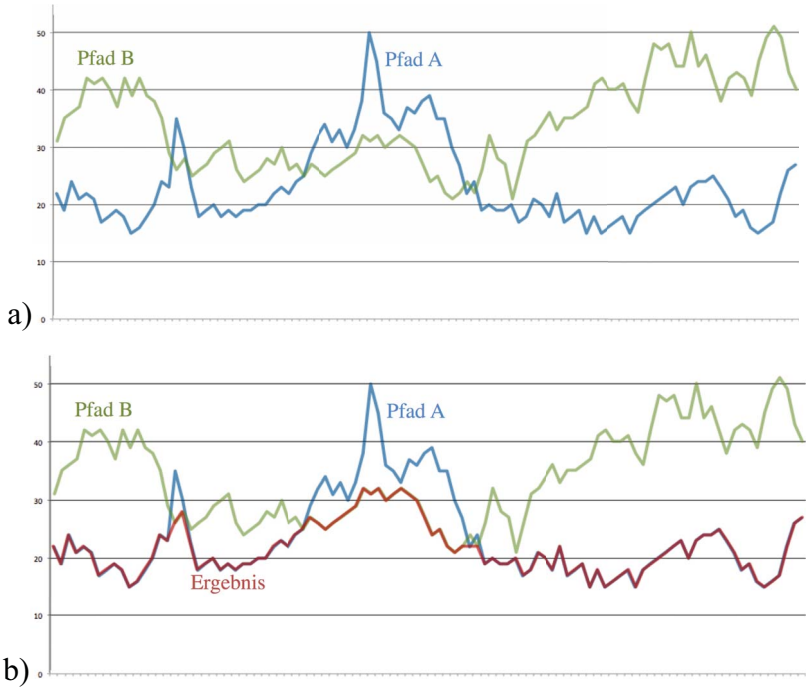

Abbildung 5.7: **Beispiel der Verzögerungszeiten unter Verwendung zweier redundanter Pfade, a) Übertragungsmuster zweier ankommender Pfade, b) resultierendes Empfangsmuster (rote Kurve)**

Für die Absicherung von Datensegmenten werden bei diesem Schema zusätzliche Redundanzen gesendet. Je nachdem, wie stark eine Absicherung geschehen soll, entscheidet die Höhe der Redundanz. Im oben genannten Beispiel werden lediglich zwei Pfade genutzt, was mit einer Verdopplung der Datensegmente verbunden ist. Zur Absicherung der eigentlichen Segmente kann die Verwendung von weiteren Pfaden oder die Erhöhung der Redundanz beitragen, auch wenn die Redundanz die Anzahl der genutzten Pfade übersteigt. In diesem Fall werden zusätzliche Datensegmente auf den bereits genutzten Pfaden gesendet. Eine ungebundene Justierung der Redundanzquote soll die Absicherung der Datensegmente variabel machen.

Dies kann in Abhängigkeit von der verfügbaren Datenübertragungsrate und den vorhandenen Pfaden geschehen. Ein adaptiver Algorithmus wird eingesetzt, um eine automatische Anpassung an die vorliegende Situation vorzunehmen. Hierfür ist es notwendig, die Beziehung zwischen den folgenden Aspekten zu eruieren:

- die Ausnutzung der Datenübertragungsrate durch die Applikation
- die gewünschte Redundanz
- die vorliegenden Pfade
- die gegebene Netzwerksituation

Der Algorithmus soll in der Lage sein, die Last optimal auf unterschiedliche Pfade zu verteilen.

Durch die Verteilung der Last auf unterschiedliche Pfade können sich abwechselnde Sendemuster zwischen den Pfaden entstehen, was die Gefahr von Out-of-Order-

Datensegmenten erhöht. Dies soll durch eine Priorisierung der für eine Applikation sinnvollsten Pfade weitgehend kompensiert und die Last entsprechend verteilt werden. Es erlaubt der Applikation, die besten Pfade automatisch auszuwählen.

Die Beobachtung der Pfadqualität ist von großer Bedeutung für ein einwandfreies Funktionieren der adaptiven Algorithmen. Techniken der Qualitätsüberprüfung müssen entwickelt werden. Veränderungen der Pfadqualität müssen ebenso erkannt werden und eine Regulierung zur Folge haben wie harte Störungen, die im Aussetzen eines Pfades resultieren. Ein Monitoring der verfügbaren Pfade soll ein Eingreifen des Benutzers in der letzten Folge erlauben. Ebenso müssen verschiedene Konfigurationen des Protokolls und das gewünschte Verhalten durch eine Anwendung einstellbar sein.

Der Ausfall eines einzelnen Pfades wirkt sich auf die Gesamtleistung der Verbindung aus: Redundanzen fallen weg oder die Datenübertragungsrate verringert sich. Durch die Nutzung adaptiver Mechanismen kann beim Ausfall eines Pfades gegengesteuert werden, indem die Daten neu verteilt werden und so ein stetiger Datenfluss der zu übertragenden Nutzdaten sichergestellt wird. Zugleich kann ein Monitoring die Pfade stetig überprüfen und im Falle eines Ausfalls neue Pfade anlegen. Durch adaptive Techniken soll die Pfadauswahl so bestimmt werden, dass eine Verwendung des überlasteten Pfades weitgehend vermieden werden kann.

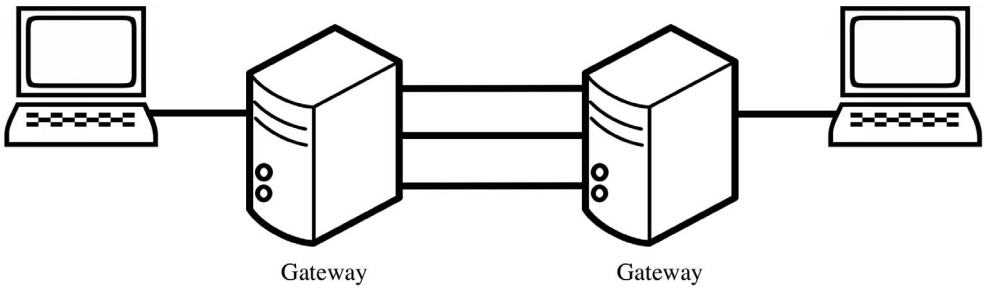

Abbildung 5.8: Gateway-Konfiguration

Damit Anwendungen ohne ausreichende Fähigkeiten, die für eine Nutzung des Protokolls nötig sind, trotzdem von der vorgeschlagenen Technologie profitieren können, ist die Entwicklung einer Netzwerkentität von Vorteil, die jeweils an den die Netzwerkverbindung einschließenden Endpunkten platziert werden kann und das Anschließen weiterer Endpunkte erlaubt. Dies kann der Überbrückung von kritischen Netzwerkdistanzen dienen. Abbildung 5.8 zeigt eine schematische Darstellung der Gateway-Konfiguration.

5.4 Parallelentwicklung ReMP TCP

Eine zeitgleiche Entwicklung eines ähnlichen redundanten Verfahrens, das an der TU Darmstadt entwickelt wurde, trägt den Namen ReMP TCP [Frömmgen et al., 2016]. Eine Implementierung findet sich unter [Frömmgen & Erbshäußer, 2015]. ReMP TCP ist ein Scheduler für MPTCP, welcher redundante Datensegmente auf unterschiedlichen Subflows sendet. Damit können die durchschnittliche Latenz und der Jitter verringert werden, wie es auch bei dem in dieser Arbeit beschriebenen Verfahren der Fall ist.

ReMP TCP sendet redundante Datensegmente auf den langsameren Subflows, solange dies durch das Überlastprotokoll von MPTCP zugelassen wird. Die Veröffentlichung beschreibt eine Feld-Evaluation des Protokolls und einen Vergleich der Leistung mit anderen Protokollen.

ReMP TCP erlaubt eine dynamische Aktivierung während des Betriebs unter MPTCP. Dies ermöglicht eine Aushandlung zwischen den Endpunkten, ob eine Redundanz gewünscht ist. Der Scheduler funktioniert auch, wenn er nur auf einer Seite aktiviert wurde.

Der Scheduler besitzt zwei Modi hinsichtlich des Verhaltens bei wiederholtem Senden von verlorenen Datensegmenten:

- der sogenannte *conservative Mode* verhält sich wie das TCP Protokoll und führt eine erneute Übertragung durch;
- der *agressive Mode* sendet das Datensegment nicht erneut, wenn es auf einem anderen Subflow eine Empfangsbestätigung gegeben hat.

Um Inkonsistenzen innerhalb der TCP-Sequenz zu vermeiden, wird die TCP-Sequenz des verlorengegangen Datensegments weitergeführt.

Das Protokoll besitzt eine ähnliche Funktionalität wie das in dieser Arbeit vorgeschlagene Verfahren – es bestehen jedoch mehrere Unterschiede. ReMP TCP sendet auf einem primären Subflow alle Daten. Alle anderen Subflows werden dann für Redundanzen verwendet, wenn sie zum Senden bereit sind. Eine dynamische Koordination der gesendeten Daten auf den verschiedenen Subflows findet nicht statt. Die Anpassung der Subflow-Nutzung folgt dem gegebenen Überlastkontrollprotokoll von MPTCP. Es kann keine Redundanz zugesichert werden und es kann nicht auf Veränderungen der Netzwerksituation reagiert werden, indem Daten umverteilt werden. Eine unabhängige Festlegung der gewünschten Absicherung durch den Benutzer oder die Applikation ist nicht möglich. Ebenso wird kein Monitoring der Netzwerksituation zur Verfügung gestellt.

Abbildung 5.9 zeigt ein Redundanz-Datensegmentdiagramm[34] von ReMP TCP unter Last. Wie daraus hervorgeht, werden in zwei Abschnitten keine Redundanzen gesendet. Dies resultiert daraus, dass die Datenübertragungsrate des primären Subflows nicht durch die der anderen beiden Subflows gedeckt wird. Vorteilhaft ist hier, dass eine zugesicherte Datenübertragungsrate durch den primären Subflow zur Verfügung gestellt wird. Damit ist aber eine Abhängigkeit vom primären Subflow gegeben. Abbrüche oder Verschlechterungen des primären Subflows werden nicht dynamisch kompensiert. Das Verfahren beschreibt eine „Maybe"-Strategie, die dann Redundanzen versendet, wenn es möglich ist.

[34] Vgl. Ausführungen in Kapitel 6.1.1

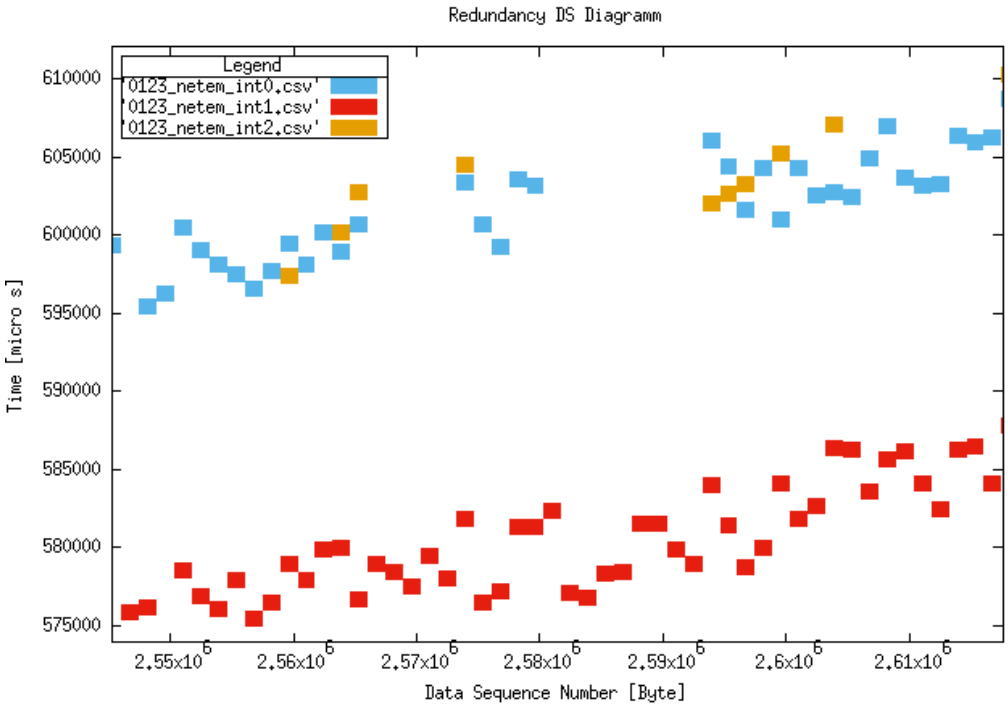

Abbildung 5.9: Redundanz-Datensequenzdiagramm von Re MP TCP unter Last und ausgenutzten Subflows [Hunger et al., 2016]

Eine geordnete Verteilung der Redundanzen auf mehrere Subflows existiert nicht. ReMP TCP wurde nur in Hinblick auf die Nutzung von zwei Subflows entwickelt. Dies ist im mobilen Bereich durchaus sinnvoll, da für mobile Endgeräte normalerweise selten mehr als zwei Kommunikationsverbindungen benutzt werden. Redundanzen werden immer dann hinzugefügt, wenn es möglich ist – egal wo oder wann es geht. Dies führt ebenfalls zu den in Abbildung 5.9 beobachteten Löchern.

Der Ausfall des primären Subflows führt zu einem Ausfall der gesamten Verbindung. Die anderen Subflows sind nicht in der Lage den Verlust zu kompensieren. Gesendete Redundanzen können nicht die primären Datensegmente ersetzen.

Eine zugesicherte Redundanz für die stationäre Verbindung im Bereich der Telemedizin ist von Vorteil. Eine intelligente Verteilung der Redundanzen auf die einzelnen Ressourcen ist hierfür die bessere Alternative. Von Vorteil ist bei ReMP TCP die automatische Vermeidung von Out-of-Order-Segmenten, da stets alle primären Datensegmente auf einem Subflow gesendet werden. Ein dynamisches Erzeugen von verschiedenen Subflows und eine praktikable Neuerstellung bei einem Subflow-Abbruch bietet ReMP TCP nicht an. Es ist sinnvoll, dass bei Subflow-Ausfall zum einen die Lastverteilung neu geregelt wird und zum anderen neue Subflows erstellt werden, die möglichst unabhängig zur identifizierten Störung verlaufen. Die Adaption eines solchen Störungsfalls steht im Fokus in dieser Arbeit.

6 Entwicklung eines redundanten MPTCP-Schedulers (rMPTCP)

In diesem Kapitel geht es um die Entwicklung eines redundanten Mehrwegprotokolls, das den in Kapitel 5.3 beschriebenen Lösungsansatz umsetzt. Hierzu wird als Basis das vorhandene Protokoll Multipath-TCP (MPTCP) verwendet und eine Modifikation entwickelt, die mithilfe von adaptiv eingesetzter Redundanz Latenzzeiten glätten und minimieren sowie die Zeit der Ausfallsicherung vermindern soll.

Eingangs werden die Grundkonzepte und mathematischen Grundlagen des redundanten Multipath Transport Protokolls (rMPTCP) beschrieben. Es folgen:

- der Aufbau und die Funktion des Daten-Schedulers
- eine Diskussion über die Optimierung des Sende- und Empfangsverhaltens
- eine Modellierung des Pfad-Managements

Für die Erkennung von Störungen werden Funktionalitäten entwickelt, die eine adaptive Anpassung an die Situation erlauben.

Es werden zum Teil Inhalte aus den wissenschaftlichen Veröffentlichungen verwendet, die der Autor dieser Arbeit bereits zum Thema getätigt hat [Hunger & Klein, 2016] [Hunger et al., 2016]. Bei der Entwicklung des rMPTCP-Schedulers haben zudem mehrere universitäre Abschlussarbeiten unter der Betreuung des Autors unterstützend mitgewirkt [Zhang, 2016], [Verbunt, 2017].

6.1 Aufbau und Grundlagen von rMPTCP

In diesem Kapitel werden die Grundkonzepte von rMPTCP beschrieben. Hierzu gehören die mathematische Basis sowie das grundlegende Funktionsprinzip. Es wird ein Überblick über den Ablauf des Protokolls und seiner Implementation gegeben.

6.1.1 Mathematische und methodische Grundlagen von rMPTCP

Datensegmentverluste und eine darauffolgende Neuübertragung lösen in der Regel eine Erhöhung der Latenz und damit der Ansprechzeit der Applikation aus. Unter der Annahme, dass mehrere parallele Pfade und ihre Knotenpunkte statistisch unabhängig sind sowie ihre Dienstgüteeigenschaften homogen sind, wird die Fehlerwahrscheinlichkeit durch die folgende Formel dargestellt [Hunger & Klein, 2016]:

$$P_{MP} = \prod_{i=1}^{n} P_{Pfad,i} \tag{6.1}$$

wobei *n* die Anzahl der parallelen Pfade ist und $P_{Pfad,i}$ die Wahrscheinlichkeit, dass ein Fehler auf Pfad *i* eintrifft. Je mehr Redundanz durch zusätzliche parallele Pfade gegeben ist, desto geringer ist die Fehlerwahrscheinlichkeit.

Bei einer Fehlerwahrscheinlichkeit eines Knotenpunkts von p_{node} gilt für die Pfadzuverlässigkeit *R* eine Wahrscheinlichkeit von:

$$R_{intakt} = (1 - p_{node})^k \tag{6.2}$$

wobei *k* die Anzahl der Knotenpunkte darstellt und alle dieselbe Fehlerwahrscheinlichkeit besitzen. Die sich hieraus ergebende Wahrscheinlichkeit eines Fehlers auf dem gesamten Pfad beträgt:

$$P_{Err,Pfad} = 1 - (1 - p_{node})^k \tag{6.3}$$

Aus oben gegebener Formel 6.1 lässt sich die Fehlerwahrscheinlichkeit der gesamten Verbindung P_{MP} für mehrere unabhängige Pfade errechnen:

$$P_{MP} = [1 - (1 - p_{node})^k][1 - (1 - p_{node})^l] \dots [1 - (1 - p_{node})^m] \tag{6.4}$$

wobei *k*, *l* und *m* die Anzahl der Knotenpunkte auf den einzelnen Pfaden sind.

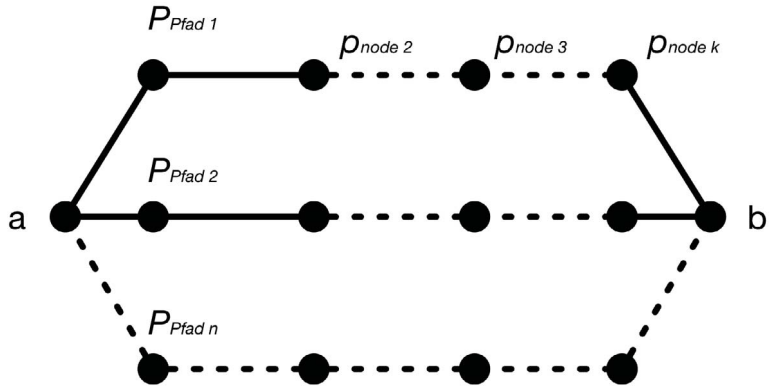

Abbildung 6.1: Gesamtwahrscheinlichkeit des Paketverlusts statistisch unabhängiger Pfade

Abbildung 6.1 zeigt ein Modell der Verbindungen zwischen zwei Entitäten *a* und *b*. Die Fehlerwahrscheinlichkeit wird für ein Paket, das redundant über alle drei Pfade versendet wird, mit der Formel 6.4 angegeben.

Im oben gezeigten Fall sind alle Pfade statistisch unabhängig. Gemeinsame Knotenpunkte führen zu einer Erhöhung der Fehlerwahrscheinlichkeit. Sie stellen einen einzelnen Fehlerpunkt dar, der sich unter Überlast auf beide Verbindungen auswirkt. Auf der Verbindungsstrecke zwischen UKM und UDE existieren gemeinsame Knotenpunkte der verschiedenen Routen – vor allem auf den ersten und auf den letzten Meilen. Die statistische Abhängigkeit lässt sich mit dem Netzwerkverfahren im Teilgebiet der Zuverlässigkeit nach [Kochs, 1984] modellieren. Die Zuverlässigkeit eines einzelnen Pfades mit hintereinandergeschalteten Knoten lässt sich wie folgt darstellen:

$$R_{pfad} = R_1 \cdot R_2 \cdot \dots \cdot R_k \tag{6.5}$$

wobei R_k die Zuverlässigkeit eines einzelnen Knotens beschreibt. Demgegenüber stehen parallel genutzte Pfade mit einer Zuverlässigkeit von:

$$R_{||} = R_{pfad,1}||R_{pfad,2}|| \dots ||R_{pfad,n} \tag{6.6}$$

wobei $R_{pfad,n}$ die Zuverlässigkeit eines einzelnen parallel verlaufenen Pfades darstellt. Die Zuverlässigkeit aller parallel verlaufenden Pfade n kann hiermit durch die folgende rekursive Funktion berechnet werden [Kochs, 1984]:

$$R_{||} = f_{||}(n) = f_{||}(n-1) + R_{pfad,n} - f_{||}(n-1) \cdot R_{pfad,n}, \quad \text{mit} \quad f_{||}(0) = 0 \tag{6.7}$$

Diese lässt sich in die folgende Form bringen (Eine Herleitung kann im Anhang eingesehen werden):

$$R_{||} = f_{||}(n) = f_{||}(1) + \sum_{i=2}^{n} R_{Pfad,i} \cdot \left(1 - f_{||}(i-1)\right) \tag{6.8}$$

$$mit\ f_{||}(1) = R_{Pfad,1} \quad und\ n > 1,\ n \in \mathbb{Z}.$$

Die Zuverlässigkeit eines Netzwerks, bestehend aus unabhängigen parallelen ($R_{||}$) sowie gemeinsam genutzten (R_{pfad}) Pfaden kann mithilfe der folgenden Verknüpfung beider Funktionen 6.5 und 6.6 berechnet werden:

$$R_{gesamt} = R_{||} \cdot R_{pfad} \tag{6.9}$$

Datenpaketverluste, die auf einer Datenverbindung stattfinden, sind meist korreliert. Durch die Nutzung mehrerer Datenverbindungen können Paketverluste vermieden werden. Dies ist abhängig vom Grad der statistischen Abhängigkeit. Die *Pfad-Diversität* (PD) ist ein Maß dafür, in welcher Weise sich die Pfade voneinander unterscheiden bzw. wie hoch die statistische Unabhängigkeit ist. Diese ist durch die Anzahl der gemeinsam genutzten Knoten vorgegeben. Die PD wird im Rahmen dieser Arbeit mit der folgenden Formel definiert:

$$D = \left(1 - \frac{k}{n}\right) \tag{6.10}$$

wobei k die Anzahl der gemeinsamen Knoten ist und n die Anzahl der gesamten Knoten eines einzelnen Pfades. Besitzen alle Pfade dieselbe Anzahl an Knoten, so ist die PD beider Pfade identisch. Bei einer vollständigen Unabhängigkeit und gleicher Anzahl an Knoten ist *D=1*. Sie sind damit *divers*. Besitzen alle Pfade nur gemeinsame Knoten, so ist *D=0*.

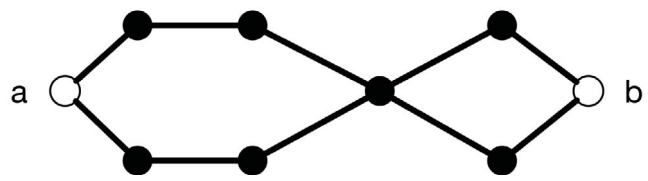

Abbildung 6.2: Pfad-Diversität mit *D = 0,75*

Abbildung 6.2 stellt ein Beispiel eines Mehrwegnetzwerks dar, bei dem ein einzelner Knoten von allen genutzten Pfaden verwendet wird. Die PD beträgt hierbei *D=0,75*.

Unter der Annahme, dass alle Knoten dieselbe Fehlerwahrscheinlichkeit besitzen, lässt sich Funktion 6.9 zusammen mit Funktion 6.10 auf die folgende Zuverlässigkeit für verschiedene Fälle berechnen:

a) Einzelwegverbindung:

$$R_{gesamt} = R^k \tag{6.11}$$

b) 2-Mehrwegverbindung:

$$R_{gesamt} = (R^{D_1 \cdot n_1} + R^{D_2 \cdot n_2} - R^{D_1 \cdot n_1} \cdot R^{D_2 \cdot n_2}) \cdot R^k \tag{6.12}$$

c) 3-Mehrwegverbindung:

$$R_{gesamt} = (R^{D_1 \cdot n_1} + R^{D_2 \cdot n_2} + R^{D_3 \cdot n_3} - R^{D_1 \cdot n_1} R^{D_2 \cdot n_2} - R^{D_1 \cdot n_1} R^{D_3 \cdot n_3} - R^{D_2 \cdot n_2} R^{D_3 \cdot n_3} + R^{D_1 \cdot n_1} R^{D_2 \cdot n_2} R^{D_3 \cdot n_3}) \cdot R^k \tag{6.13}$$

wobei *k* die Anzahl der gemeinsamen Knoten ist und *n* die Anzahl der gesamten Knoten eines einzelnen Pfades. Die Fehlerwahrscheinlichkeit der gesamten Verbindung beträgt:

$$P_{Err} = 1 - R_{gesamt} \tag{6.14}$$

Abbildung 6.3 zeigt eine Darstellung der Fehlerwahrscheinlichkeit eines Netzwerks mit *n=10* Knoten und einer Knotenzuverlässigkeit von *R=0,9* bei unterschiedlicher PD. Wie aus der Grafik hervorgeht, sinkt die Fehlerwahrscheinlichkeit mit vermehrter Nutzung mehrerer unabhängiger Pfade.

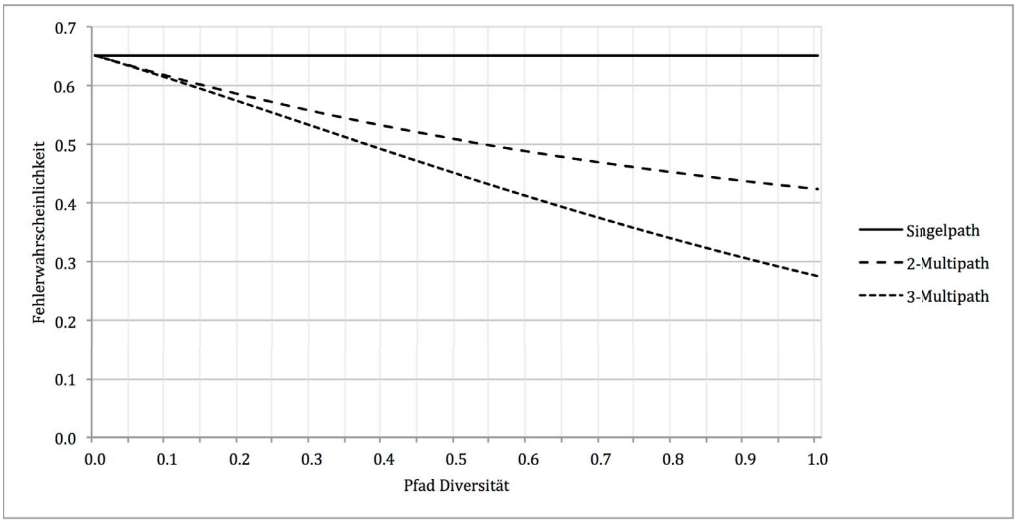

Abbildung 6.3: Fehlerwahrscheinlichkeit unter verschiedener Pfad-Diversität (*n=10, R=0,9*)

Bei der im Rahmen dieser Arbeit entwickelten Methodik werden auf mehreren Pfaden redundante Segmente versendet. Dabei wird mit einer frei definierbaren Anzahl an gleichzeitig versendeten Segmenten gearbeitet. Eine Redundanz im Sinne der technischen Zuverlässigkeit ist durch die Nutzung mehrerer Verbindungen gegeben. Diese ist abzugrenzen von einer Redundanz hinsichtlich der Leistungsfähigkeit einer Verbindung bezogen auf die Anzahl der redundant versendeten Segmente. Die Anzahl der Segmente kann von der Anzahl der Verbindungen abweichen. Die *Redundanz* R_L gibt die Höhe der

zusätzlich versendeten Segmente an. Um eine logische Unterscheidung zwischen den Segmenten zu ermöglichen, wird hierbei zwischen primären Segmenten und redundanten Segmenten unterschieden. Im technischen Sinne sind diese Segmente identisch.

Ist eine leistungsbezogene Redundanz von $R_L=1$ gegeben, so wird jedes Datensegment genau einmal verdoppelt. Die zur Verfügung stehenden Pfade werden genutzt, um jeweils eins der Datensegmente zu übertragen. Eine leistungsbezogene Redundanz von $R_L=0$ besagt, dass nur ein primärer Datenstrom existiert. Dieser kann ebenfalls unter Verwendung verschiedener (redundanter) Pfade übertragen werden, um z.B. eine erhöhte Datenübertragungsrate zu erreichen.

Der in dieser Arbeit vorgestellte Scheduler bedient sich einer weiteren Größe, der sogenannten *Redundanzquote Q*. Die Redundanzquote gibt an, wie oft ein Segment gleichzeitig übertragen wird. Dieser Wert wird verwendet, um eine logischere Berechnung nach Segmentanzahl zu ermöglichen. Sie ist durch die folgenden Funktionen gegeben:

$$Q = R_L + 1 \tag{6.15}$$

$$Q_{Prozent} = (R_L + 1) \cdot 100 \tag{6.16}$$

Zwei weitere einzuführende Begriffe sind die Pfad-Homogenität bzw. -Heterogenität. Diese Größen beschreiben die Gleichheit der Eigenschaften der verschiedenen Pfade. Diese Eigenschaften beziehen sich im Rahmen dieser Arbeit auf die vier Dienstgüte-Parameter.[35] Sind mehrere Pfade homogen, dann besitzen sie dieselben Eigenschaften und können genau gleich behandelt werden. Dies hat vor allem Auswirkungen auf die Verteilung der Datensegmente auf die verschiedenen Pfade.

Die Datenübertragungsrate entscheidet darüber, wie viele Datensegmente gesendet werden können. Sind die Pfade heterogen, so können die gleichzeitig zu sendenden Segmente nicht gleich verteilt werden. Die Verzögerungszeit eines Pfades wirkt sich auf den Zeitpunkt aus, an dem ein Segment beim Empfänger eintrifft. Das Produkt dieser beiden Größen (BDP) bestimmt, wie viele Segmente gesendet werden, ohne dass ein Bestätigungssegment vom Empfänger eintreffen muss. Heterogenität verringert den Vorteil einer Mehrwegkommunikation.

Auf Ende-zu-Ende-Ebene ist keine wirkliche Kontrolle über den Pfad möglich, den die Segmente nehmen, da dies Aufgabe der Vermittlungsschicht ist. Ein Pfad bildet die physikalische Verbindung zwischen zwei Endpunkten. Unter MPTCP werden die zur Verfügung stehenden Pfade genutzt, um mehrere logische TCP-Verbindungen zu schließen. Diese sogenannten *Subflows* unterliegen hierbei den Eigenschaften des genutzten Pfades. Subflow-Diversität ist nur dann möglich, wenn die darunterliegenden Pfade ebenfalls disjunkt sind. Mehrere Subflows können sich gemeinsam einen Pfad teilen. Der Vorteil besteht darin, dass die Eigenschaften des Pfades für alle Subflows homogen sind. Der Nachteil ist die Abhängigkeit auftretender Fehler und Überlastungen.

[35] Vgl. hierzu Kapitel 2.4

Im Zuge der Entwicklung des redundanten Mehrwegprotokolls wurde nach einer besseren Form der Darstellung für übertragene Segmente über eine Mehrwegverbindung gesucht. Übertragene Segmente werden hierbei über unterschiedliche Subflows übertragen, welche unterschiedliche Eigenschaften hervorbringen. Es ist von Vorteil, die verwendeten Subflows der Übertragung mit in einem Diagramm darzustellen. Unter rMPTCP können mehrere Segmente dieselbe Datensequenznummer besitzen, wenn sie redundant übertragen wurden.

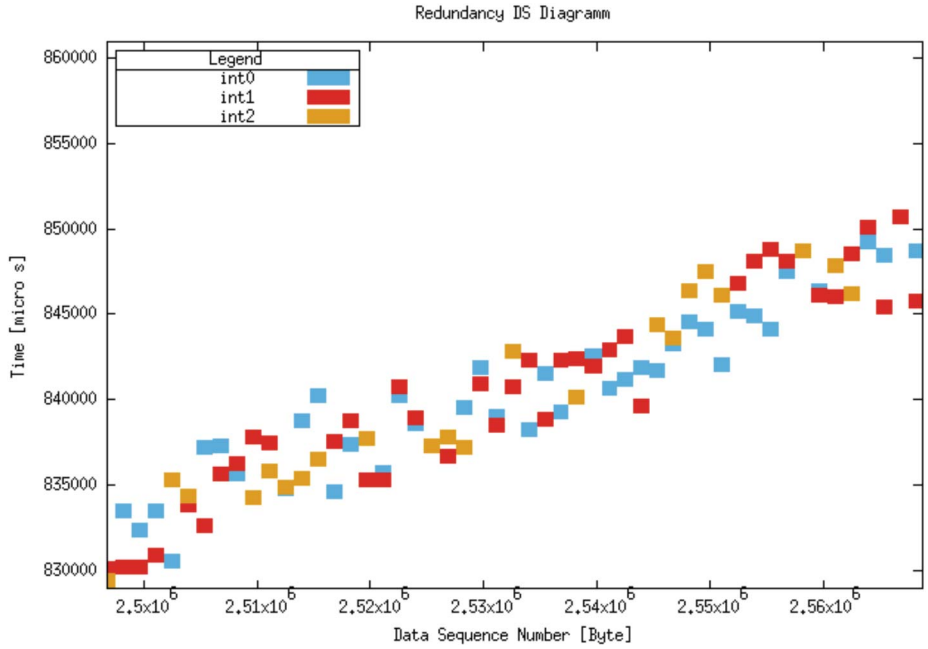

Abbildung 6.4: Redundanz-Datensequenzdiagramm unter rMPTCP [Hunger et al., 2016]

Das Redundanz-Datensequenzdiagramm (RDS-Diagramm) ist eine Darstellung der zeitlichen und logischen Abfolge der Datensegmente, die in Form eines Rechtecks eingetragen sind. Übertragene Segmente besitzen die folgenden drei Dimensionen:

1. Sende- bzw. Empfangszeit, die durch die vertikale Achse repräsentiert wird,
2. Datensequenznummer, die auf der horizontalen Achse aufgetragen wird,
3. Übertragungsweg bzw. Subflow, der durch die Farbe des Rechtecks dargestellt wird.

Abbildung 6.4 zeigt ein Beispiel eines RDS-Diagramms. Bei diesem Beispiel existiert eine Übertragung über drei Subflows (hier: blau, rot, gelb). Abstände zwischen den Datensegmenten auf der vertikalen Achse markieren zeitliche Unterschiede. Im genannten Beispiel zeigen sich kleinere zeitliche Unterschiede zwischen den Subflows.

In der in dieser Arbeit beschriebenen Technologie werden häufig Mittelwertberechnungen benötigt, die mithilfe einer einfachen Glättungsfunktion implementiert werden. Diese eignet sich speziell für den Einsatz im Netzwerkbereich, um intensive Rechenoperationen zu vermeiden. Die hierfür genutzte Formel nach [Jacobson, 1988] ist:

$$S_{n+1} = S_n + \alpha(C_n - S_n) \tag{6.17}$$

wobei S der geglättete Mittelwert ist, C der aktuelle Messwert und α der Glättungs-Koeffizient für die Geschwindigkeit der Anpassung des Mittelwerts. α wird in der Regel auf einen festen Wert gesetzt, der experimentell für die genutzte Anwendung hergeleitet wird. Die Formel glättet S und verhindert größere Sprünge des Wertes.

6.1.2 Funktionsbeschreibung von rMPTCP

Das redundante Multipath Transport Control Protocol (rMPTCP) ist eine Modifizierung von MPTCP und fungiert als redundantes Übertragungsschema. Hierbei werden Daten auf mehreren verfügbaren Subflows redundant zur selben Zeit gesendet. Im Allgemeinen ist die effektive Datenübertragungsrate mit der von Standard-TCP zu vergleichen – sie ist dabei aber weniger abhängig von Störungen.

rMPTCP verfolgt die folgenden Ziele [Hunger & Klein, 2016]:

1. *Zeit eines Failovers*: Im Falle des Zusammenbruchs eines Subflows soll ein Failover im besten Fall eine Schaltzeit von Null haben.
2. *Latenzglättung und Optimierung*: Die allgemeine Latenzzeit einer Datenübertragung soll auf ein Minimum reduziert werden. Es sollen Latenzspitzen und Latenzvarianzen (Jitter) minimiert und geglättet werden.
3. *Datenintegrität und Zuverlässigkeit*: Die Integrität und Richtigkeit der Daten muss unter allen Umständen gewährleistet sein, auch wenn die Vermeidung von Datenverlusten die Latenzzeit erhöhen sollte.
4. *Einsatzfähigkeit*: Das Protokoll muss für den Einsatz innerhalb eines heterogenen Netzwerks wie dem Internet geeignet sein.

Unter der Verwendung von mehreren Subflows lassen sich Datensegmentduplikate gleichzeitig versenden. Eine Empfangsstation muss lediglich das erste erhaltene Segment in Empfang nehmen und alle anderen Segmente verwerfen, die dieselbe Datensequenznummer besitzen. Falls ein Subflow von Überlastungen auf seinem Datenpfad betroffen ist, können Datensegmente auf anderen Subflows weiterhin das Ziel erreichen. Auf diese Weise können Latenzvariationen mithilfe der verschiedenen Subflows ausgeglichen werden.

Eine geringe Zeit für ein Failover wird dadurch erreicht, dass mindestens zwei Subflows dieselben Datensegmente senden. Wenn ein Subflow zusammenbrechen sollte, wird keine technische Umschaltung benötigt. Das bedeutet, dass die Datensegmente, die auf dem zusammengebrochenen Subflow verblieben sind, nicht wiederhergestellt und erneut übertragen werden müssen, da die Datensegmente bereits auf anderen Subflows gesendet wurden. Eine Umschaltung zu einem anderen Subflow, die mit einem Verbindungsaufbau und -abbau einherginge, ist somit nicht notwendig. Im besten Fall kann hierdurch die Zeit eines Failovers auf Null gesenkt werden. Voraussetzung dafür ist, dass beide Subflows dieselbe Datenübertragungsrate unterstützen.

Ein möglicher zeitlicher Unterschied ergibt sich, wenn bereits von Anfang an ein Zeitunterschied zwischen den Subflows existiert. Dies ist der Fall, wenn die Dienstgü-

teeigenschaften, wie z.B. das OWD der durch die Subflows verwendeten Pfade, unterschiedlich sind. Um diese Heterogenitätsprobleme zu minimieren, ist ein Pfad-Management einzusetzen, welches Subflows auswählt, die bestimmten Pfadeigenschaften hinsichtlich der Datenübertragungsrate und der Umlaufzeit folgen.

Durch die Nutzung von MPTCP als Modifikationsbasis und das darunterliegende TCP werden die Anforderungen drei und vier hinreichend erfüllt[36]. Ein verlorenes Datensegment auf einem Subflow wird mithilfe des ARQ-Protokolls von TCP wiederhergestellt, sollte kein entsprechendes redundant gesendetes Segment auf einem anderen Subflow rechtzeitig empfangen werden. Die Datenintegrität wird mithilfe der zur Verfügung stehenden Techniken unter TCP sichergestellt. MPTCP wird im Internet weitgehend unterstützt, wodurch rMPTCP ebenfalls internetfähig ist.

Eine rMPTCP-Verbindung besitzt mindestens einen primären und einen sekundären Subflow. Der primäre Subflow wird benötigt, um die Hauptverbindung aufrechtzuerhalten und andere Subflows zu erstellen. Diese Signalisierungsdaten werden lediglich auf dem primären Subflow ohne leistungsbezogene Redundanz ausgetauscht.

Jeder Subflow besitzt einen eigenen Sequenznummernbereich, der durch die darunterliegende TCP-Verbindung bestimmt wird. Diese Sequenznummer wird auf Daten-Ebene mit der MPTCP-Datensequenznummer gekoppelt. Sie ist in jedem replizierten Subflow gleich. Auf der Empfangsseite muss eine Station nur das erste empfangene Datensegment annehmen und alle anderen Datensegmente mit derselben Datensequenznummer löschen.

Abbildung 6.5 zeigt eine schematische Darstellung einer rMPTCP-Verbindung mit zwei Subflows. Die Rechtecke stellen die Datensegmente dar, von denen jedes zwei Segmentnummern besitzt. Die obere Sequenznummer bezieht sich auf die Datenebene der MPTCP-Sequenzierung und ist auf beiden Subflows gleich. Die untere Nummerierung stellt die TCP-Sequenzierung auf der Subflow-Ebene dar, deren Nummernraum sich bei den verschiedenen Subflows unterscheidet.

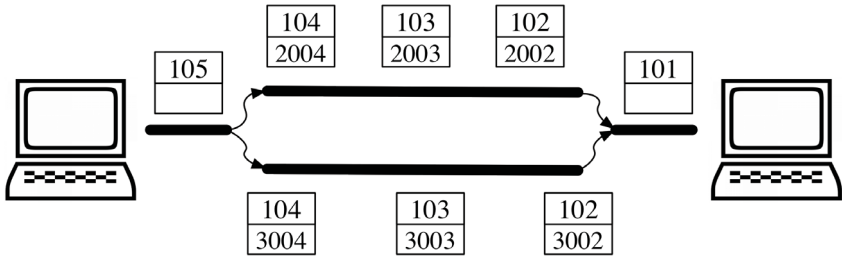

Abbildung 6.5: rMPTCP Sende-Schema

Die Entwicklung von rMPTCP erstreckt sich auf folgende Bereiche:

1. *Paket-Scheduler:* die Verteilung der Daten auf die verfügbaren Subflows und das Setzen von Redundanzen. Anpassen des Sendeverhaltens in Hinblick auf die Situation innerhalb des Netzwerks.

[36] Vgl. die aufgelisteten Ziele für rMPTCP in Kapitel 6.1.2

2. *Senden und Empfangen:* Besonderheiten im Umgang mit den Datensegmenten beim Senden und Empfangen sowie die Verwendung benötigter Warteschlangen und Pufferspeicher.
3. *Pfad-Management:* Auswirkungen unterschiedlicher Pfadeigenschaften auf die verwendete Datenverbindung und Anpassung der Pfade in Hinblick auf die Situation innerhalb des Netzwerks.
4. *Ausfallerkennung:* Erkennen von Ausfällen und entsprechende Anpassung hinsichtlich der genutzten Pfade.
5. *Application Programmable Interface (API):* Nutzen von Protokolleinstellungen und Besonderheiten unter rMPTCP durch rMPTCP-aware Applikationen.

6.1.3 Ablauf von rMPTCP

In den oben angesprochenen fünf Bereichen besitzt der Paket-Scheduler eine Schlüsselrolle. Er muss in der Lage sein:

1. die verfügbare Datenübertragungsrate so zu nutzen, dass einer Applikation möglichst viel hiervon zur Verfügung steht;
2. redundante Daten so zu verteilen, dass die primär gesendeten Daten der Applikation möglichst nicht beeinträchtigt werden, aber möglichst effektiv abgesichert sind;
3. alle Datensegmente, ob primäre oder redundante, auf die Situation im Netzwerk angepasst zu senden und im Störungsfall die Datensegmente umzuverteilen;
4. einer Applikation oder einem Benutzer die Option zu geben, bestimmte Anforderungen der Anwendung entgegenzunehmen und anzuwenden.

Das Pfad-Management dient dazu, aus einem Pool an verfügbaren Pfaden die bestgeeignetsten herauszufinden. Es soll in der Lage sein:

1. bestmögliche Pfadkombinationen herauszufiltern, die möglichst unabhängige Pfade besitzen und sich nicht die Datenübertragungsrate eines Netzwerkinterfaces teilen;
2. die Pfade so zu priorisieren, dass primäre Datensegmente möglichst über einen einheitlichen Subflow gesendet werden, damit keine zusätzlichen Latenzbelastungen durch Out-of-Order Segmente entstehen;
3. bei veränderten Situationen im Netzwerk und bei Ausfall einer Verbindung kurzzeitig Ersatz-Subflows zur Verfügung zu stellen, die einen Abfall der Datenübertragungsrate verhindern und einen sauberen Übergang der Daten von einem Pfad zum anderen ermöglichen.

Das Aktivitätsdiagramm in Abbildung 6.6 zeigt eine schematische Darstellung des rMPTCP-Ablaufs: Der Prozess beginnt damit, die Benutzereingaben der API entgegenzunehmen und als Parameter innerhalb der Implementierung zu verwenden. Es folgt die Bestimmung der bestmöglichsten Pfade und der besten Kombination hieraus. Zu diesem Zweck wird eine Messung der Pfadeigenschaften vorgenommen und eine Priorisierung anhand der Messung durchgeführt.

Wird ein Störungszustand eines Subflows erkannt, können zwei adaptive Algorithmen genutzt werden, um auf die Situation einzugehen. Ein Algorithmus gleicht die Störung mithilfe von neuen Subflows aus, der andere passt die Nutzung der neuen bzw. der verbleibenden Subflows an, um möglichst die Anforderungen der Anwendung halten zu können.

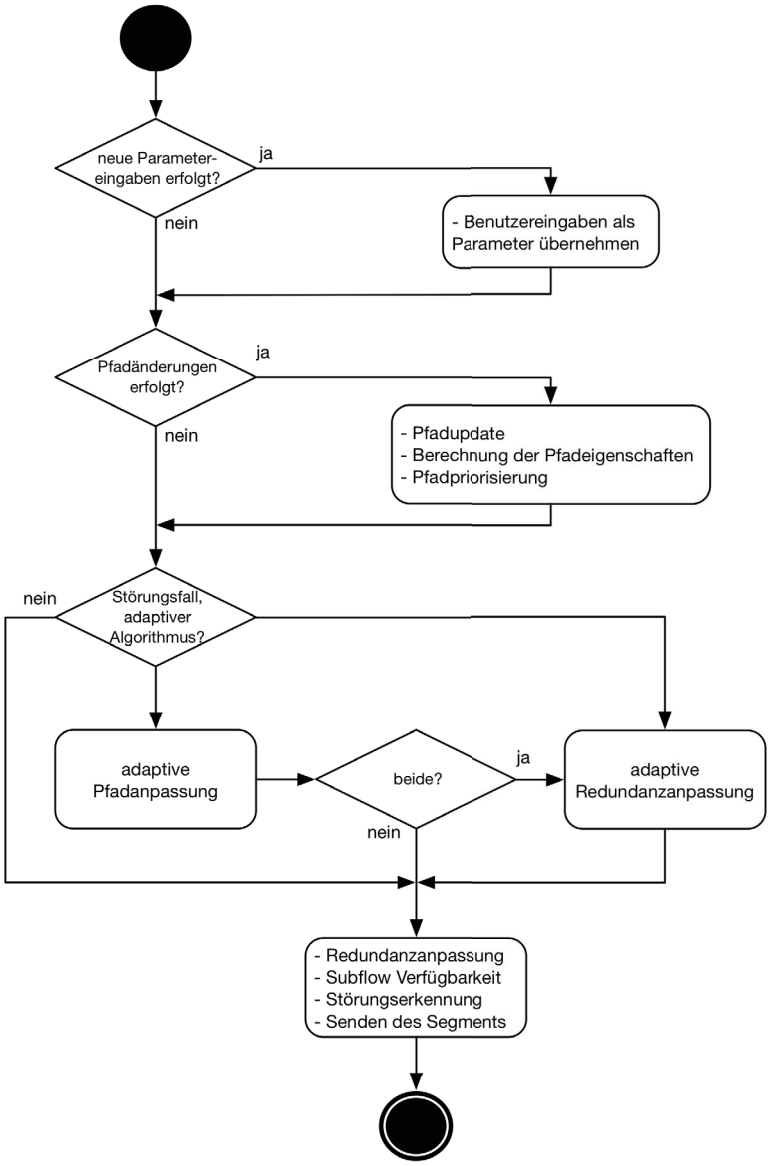

Abbildung 6.6: Aktivitätsdiagramm für das Senden eines Segments

Einem zu sendenden Datensegment werden in dem Maße zusätzliche redundante Segmente hinzugefügt, wie es die Datenübertragungsrate der Subflows und die Anforderungen des Benutzers erfordern. Danach wird die Verfügbarkeit der zu nutzenden

Subflows geprüft und gewartet, bis die erforderlichen Subflows bereit sind, um die Segmente zu senden. Eine weitere Störungsüberprüfung wird aktiviert, um den Subflow beim Senden zu überwachen. Diese Überwachung bewirkt eine Schaltung verschiedener Störungszustände.

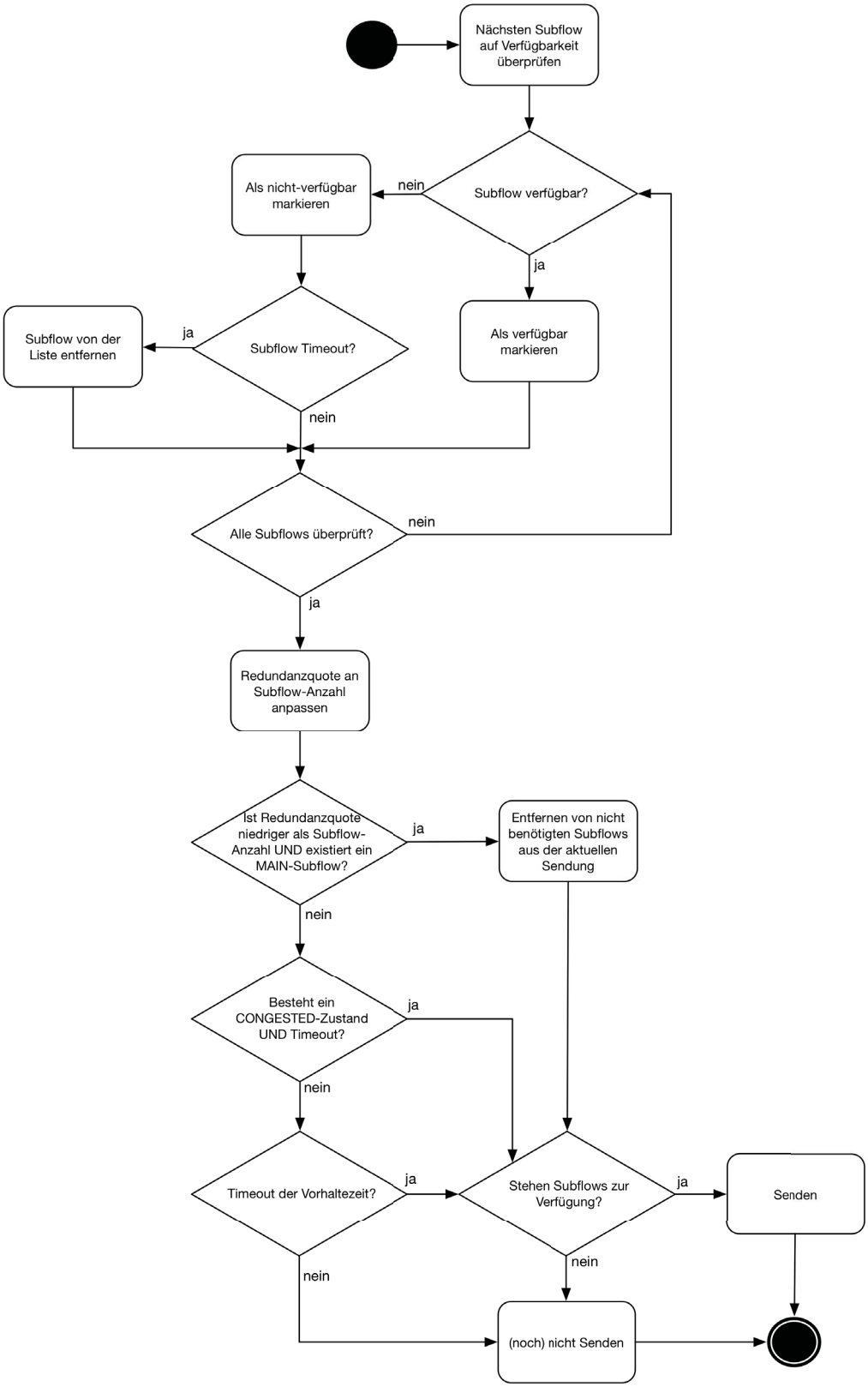

Abbildung 6.7: Aktivitätsdiagramm des rMPTCP-Schedulers für die Subflow-Verfügbarkeit

Wesentlich für rMPTCP ist es herauszufinden, ob genügend Subflows zur Verfügung stehen, damit die gewünschte Anzahl an Segmenten versendet werden kann. Das oben dargestellte Aktivitätsdiagramm in Abbildung 6.7 zeigt den Funktionsablauf für die Überprüfung der Subflows auf Sendebereitschaft.

In diesem Ablauf werden zunächst alle vorhandenen Subflows überprüft und, falls ein Subflow bereits längere Zeit nicht mehr zum Senden verwendet werden konnte, aus der aktuellen Sendeliste entfernt. Ist ein Subflow verfügbar, so wird er markiert, um für die Sendung eines Datensegments verwendet zu werden. Die Anzahl der Subflows ist grundlegend für die Nutzung der in dieser Arbeit entwickelten Algorithmen. Steht nur ein Subflow zur Verfügung, so müssen die primären Daten ohne zusätzliche Redundanz gesendet werden. Stehen mehrere Subflows bereit, die keinem Überlastzustand unterliegen, so können diese für eine festgelegte Redundanzquote genutzt werden. Sind mehr Subflows verfügbar, als es die voreingestellte Redundanzquote definiert, können nicht benötigte Subflows entfernt werden. Wird ein Überlastzustand erkannt, muss so schnell wie möglich mit der minimalen Anzahl an Subflows gesendet werden.

Die in den beiden Abbildungen dargestellten Aktivitätsdiagramme nutzen Algorithmen, die im Folgenden erläutert werden. Das Ziel ist es, ein Datensegment so abgesichert wie möglich zu senden. Die genutzten Algorithmen sind in der Lage, die vorhandenen Ressourcen bestmöglich und nach den Erfordernissen der Applikation entsprechend zu nutzen.

Der in dieser Arbeit vorgestellte Scheduler unterscheidet zwischen verschiedenen Zuständen der Subflows. Diese sind wichtig, um die Nutzbarkeit oder Bereitschaft eines Subflows festzustellen. Die folgenden Subflow-Zustände werden unterschieden:

- *Aktiv* (active): Ein Subflow wird aktiv in einem Sendeschema verwendet. Der Scheduler wählt diesen Subflow bei Bedarf aus, um Datensegmente zu versenden;
- *Nutzbar* (enabled): Ein Subflow besteht, wird aber nicht aktiv zum Senden verwendet. Er wird aufrechterhalten, um einen schnellen Failover zu ermöglichen;
- *Verfügbar* (available): Ein Subflow ist zum Senden verfügbar und kann für das nächste anstehende Segment genutzt werden;
- *Main*: Ein Subflow wird als hauptsächlicher Träger der primären Datensegmente ausgewählt;
- *Support:* Ein Subflow wird lediglich als Träger von redundant gesendeten Segmenten verwendet;
- *Ersatz* (Substitute): Ein Subflow wird als suboptimaler Ersatz im Falle eines Failovers aktiv eingesetzt;
- *Überlastet* (Congested): Ein Subflow besitzt Anzeichen von Überlastungen und wird nach einer festgelegten Zeit in diesem Zustand durch Ersatz-Subflows ersetzt, bis sich die Situation wieder verbessert hat;
- *Backup*: Ein Subflow dient als Notfall-Backup.

6.2 Daten-Scheduler unter rMPTCP

Aufgabe des Daten-Schedulers ist es, die Datensegmente auf den einzelnen Subflows zu verteilen. Unter rMPTCP bieten sich hierfür mehrere Strategien an, die für unterschiedliche Applikationen eine Rolle spielen können. Diese Strategien werden durch die Verteilung der redundanten Segmente auf die einzelnen Subflows definiert:

rMPTCP-Scheduler Strategie 1: Eine rMPTCP-Verbindung sendet auf allen Subflows die gleiche Anzahl an Datensegmenten und bietet so eine gesicherte leistungsbezogene Redundanz. Es werden alle Segmente auf allen Subflows stets redundant gesendet. Die effektive Datenübertragungsrate definiert sich durch den Subflow mit der kleinsten Datenübertragungsrate bzw. der kleinsten Menge an Daten, die ein Subflow im Stande ist ins Netz zu senden. Müssen mehr Datensegmente gesendet werden, als einer der Subflows übermitteln kann, kommt es zu einer Verlangsamung der Verbindung. Der Vorteil ist aber, dass jedes gesendete Datensegment über eine zugesicherte leistungsbezogene Redundanz verfügt. Ein Nachteil ist die geringere Datenübertragungsrate bei eigentlich vorhandenen Ressourcen für mehr Datendurchsatz. Diese Strategie eignet sich besonders für echtzeitlastige Anwendungen, die eine kleinere Datenübertragungsrate benötigen, als der langsamste Subflow besitzt.

rMPTCP-Scheduler Strategie 2: Hierbei wird der Hauptdatenstrom auf der schnellsten Verbindung gesendet, jedoch auf den weiteren Subflows nur dann, wenn dort genügend Datenübertragungsrate bereitgestellt werden kann. Es erfolgt eine Verringerung der leistungsbezogenen Redundanz, wenn die nötige Datenübertragungsrate nicht gegeben ist. Im schlechtesten Fall würde dies mit dem Wegfall der leistungsbezogenen Redundanz auf den anderen Subflows einhergehen, wenn bei diesen die Ressourcen bereits ausgenutzt sind. Diese Strategie eignet sich für datenübertragungsintensive Anwendungen bzw. Massendaten, denen ein bestimmter Prozentsatz an leistungsbezogener Redundanz genügt. Eine zugesicherte Latenzglättung ist hier nicht gegeben.

Bei Subflows mit heterogenen Eigenschaften, die eine von der Applikation gegebene Datenübertragungsrate nicht überall erfüllen können, ist das Ergebnis also entweder eine unvollständige Datenreplikation oder eine Verlangsamung der primären Verbindung, die eine erhöhte Latenz produziert. In Hinblick auf Applikationen wie der latenzminimierten Steuerung eines ARTP ist die rMPTCP Scheduler Strategie 1 vorzuziehen: Zum einen ist der Bedarf der Datenübertragungsrate relativ niedrig und zum anderen soll stets eine durch leistungsbezogene Redundanz abgesicherte Verbindung verfügbar sein. Hierbei ist es notwendig, dass alle Subflows die benötigte Datenübertragungsrate der Applikation bereitstellen.

Hybride Vorgehensweisen bieten sich bei der Nutzung von mehr als zwei Subflows an. Hier können redundant gesendete Segmente so verteilt werden, dass eine Zusicherung von mindestens einer festen partiellen, einer einfachen oder auch einer mehrfachen leistungsbezogenen Redundanz gegeben ist. Diese müssen in Hinblick auf die verfügbaren Subflows und speziell bei Änderungen der Subflows (z.B. Ausfall) sowie ihrer Eigenschaften berücksichtigt werden. Es ist das Ziel dieser Arbeit, eine Vorgehensweise

zu entwickeln, die eine ausgewogene Bereitstellung des leistungsbezogenen Redundanzanteils ermöglicht.

Wird der rMPTCP-Scheduler-Strategie 1 gefolgt, kann die Lastverteilung der primären sowie der redundant gesendeten Datensegmente auch auf mehrere Subflows beliebig aufgeteilt werden. Das heißt, immer wenn genügend Subflows zur Verfügung stehen, um eine gewünschte leistungsbezogene Redundanz zu erfüllen, werden diese genutzt. Dies bietet den Vorteil, dass der nächste zum Senden verfügbare Subflow Datensegmente schneller verschicken kann, ohne zunächst darauf warten zu müssen, bis alle anderen Subflows ebenfalls sendebereit sind. Dies verringert die Wartezeit eines Segments, das gesendet werden muss.

Ein Nachteil bei dieser Sendemethode ist allerdings, dass die ungleichmäßige Verteilung der primären und redundanten Datensegmente zu Out-of-Order Segmenten führen kann. Dies bedeutet, dass bei heterogenen Pfadeigenschaften Datensegmente nicht in der gesendeten Reihenfolge beim Empfänger ankommen können. Dies führt wiederum zu erhöhten Latenzzeiten bzw. Jitter. Eine Priorisierung eines einzelnen Subflows, auf dem die Datensegmente mit erhöhter Priorisierung versendet werden, ist hier notwendig.

Eine weitere Möglichkeit der Verbesserung ist der Einsatz zusätzlicher leistungsbezogener Redundanz auf mehreren Subflows, um die Segmentverlustwahrscheinlichkeit weiter zu verringern. Das Ziel ist es hier, dies in Abhängigkeit von der eigentlichen Datenauslastung durch den primären Datenstrom und der Datenübertragungsbeschränkung durch die einzelnen Subflows sowie der Situation im Netzwerk durchzuführen. Bei der Entwicklung des Daten-Schedulers hat unterstützend die Abschlussarbeit [Verbunt, 2017] mitgewirkt.

6.2.1 Redundanzquotierung

Eine Steuerung der leistungsbezogenen Redundanz kann auch manuell erfolgen. Einer Applikation bzw. dem Benutzer einer Applikation ist es damit möglich, selbst zu entscheiden, wie viel leistungsbezogene Redundanz benötigt wird. Die *Redundanzquote Q* spezifiziert die Anzahl der Subflows, die für das Senden eines Datensegments gleichzeitig verwendet werden. Die nutzbare Datenübertragungsrate ist von der Höhe der Redundanzquote abhängig. Wird keine leistungsbezogene Redundanz eingesetzt, können alle Netzwerkschnittstellen für die Sendung von primären Datensegmenten verwendet werden. Unter Hinzunahme von zusätzlichen Redundanzsegmenten muss immer auf alle Netzwerkschnittstellen gewartet werden, die gleichzeitig senden sollen. Hierdurch wird die Datenübertragungsrate beeinflusst.

Das redundante Senden von Segmenten wird zunächst durch die (ganzzahlige) Anzahl der Subflows definiert. In nächster Näherung wird die *maximale effektive Datenübertragungsrate* $E_{eff,max}$ *(effective Utilization)*, die bei einer ganzzahligen *Redundanzquote Q* aus der insgesamt verfügbaren *Datenübertragungsrate* B_{tot} hervorgeht, nach folgender Formel berechnet [Hunger et al., 2016]:

$$E_{eff,max} \approx B_{tot} \cdot \left(\frac{1}{2}\right)^{Q-1} \tag{6.18}$$

Eine Redundanzquote von *Q=2* bedeutet, dass jedes zu sendende Datensegment unter Nutzung von zwei Subflows redundant gesendet wird. Bei einer Redundanzquote von *Q=2* und drei verfügbaren Subflows wird diese Redundanzquote im Rahmen dieser Arbeit als *2-out-of-3-Redundanz (2oo3)* bezeichnet.

Ein Subflow-Ausfall mit einer bestimmten zugesicherten Redundanzquote kann jedoch zu einem Abfall der Datenübertragungsrate führen. Bei einer durchgängigen Redundanzquote, die nicht durch ganzzahlige Werte bestimmt wird, sondern durch Dezimal- oder Prozentangaben, werden Datensegmente nicht stetig auf derselben Anzahl der Subflows übertragen. Dies ermöglicht den Subflows eine variablere Ausnutzung der Datenübertragungsrate. Im Falle eines Subflow-Ausfalls lässt sich damit die leistungsbezogene Redundanz dynamisch auf das Maximalmögliche reduzieren und auf die noch vorhandenen Subflows verteilen, ohne dabei die von der Applikation benötigte Datenübertragungsrate zu beeinträchtigen oder die leistungsbezogene Redundanz drastisch (d.h. ganzzahlig) zu verringern.

Die effektive Datenübertragungsrate E_{eff} ist hier ein Maß für die tatsächliche Datenmenge, die eine Applikation ohne redundant gesendete Daten auf den möglichen Subflows emittieren kann. Die maximale effektive Datenübertragungsrate $E_{eff,max}$ ist abhängig von der gewählten leistungsbezogenen *Redundanz* R_L und von der möglichen *Datenübertragungsrate* B_i der einzelnen Subflows. Je höher R_L gewählt wird, desto geringer ist die maximale effektive Datenübertragungsrate, da weniger Datenübertragungsrate für die tatsächlichen Daten zur Verfügung steht. Werden z.B. die primären Daten auf einem einzelnen Subflow gesendet, so müssen die anderen Subflows die vorgelegte Datenübertragungsrate in Hinblick auf die verlangte leistungsbezogenen Redundanz unterstützen. Die folgende Formel 6.19 gibt Aufschluss über die maximale effektive Datenübertragungsrate bei der Verwendung von zwei Subflows S_0, S_1, wobei $B_0 \geq B_1$:

$$E_{eff,max} = \begin{Bmatrix} \frac{B_{tot}}{R_L+1}, & wenn\ \frac{B_1}{R_L} \geq B_0 \\ \frac{B_1}{R_L}, & wenn\ \frac{B_1}{R_L} < B_0 \end{Bmatrix} \quad , R_L \in \mathbb{R} \mid 0 \leq R_L \leq 1 \tag{6.19}$$

Es sind zwei Fälle zu unterscheiden, die durch die Heterogenität der beiden Subflows bestimmt werden.

- Beim ersten Fall ist eine Ausnutzung der Datenübertragungsrate B_{tot} aller Subflows möglich. Redundant gesendete Daten werden in Abhängigkeit von R_L entweder über einen oder über beide Subflows gesendet. Der langsamere Subflow S_1 kann solange eine bestimmte Datenübertragungsrate bedienen, bis diese die Kapazität von S_1 überschreitet.
- Ab diesem Punkt tritt der zweite Fall ein, der mit einer Absenkung der effektiven Datenübertragungsrate einhergehen muss, um eine weitere Erhöhung der leistungsbezogenen Redundanz R_L zu ermöglichen. Dies ist dann der

Fall, wenn R_L gleich dem Quotienten der Datenübertragungsraten B_0 und B_1 der beiden Subflows ist (zu erkennen als Knick im Graphen).

Die folgende Abbildung 6.8 zeigt mehrere Graphen, die die effektive Datenübertragungsrate von mehreren Fällen mit verschiedener Heterogenität für zwei Subflows darstellen.

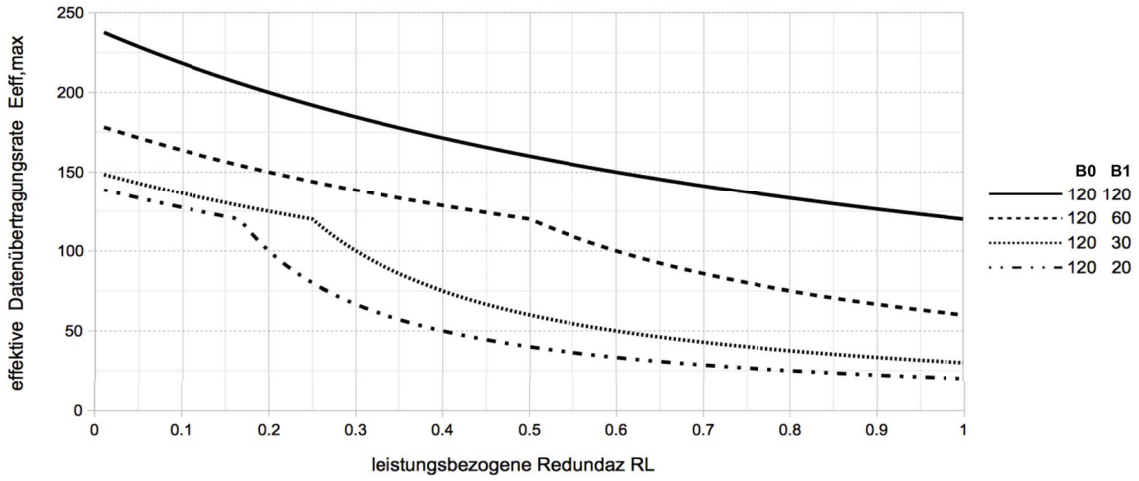

Abbildung 6.8: Redundanz-Datenübertragungsrate bei zwei Subflows und durchgehender Redundanz

Eine Erweiterung der Gleichung 6.19 auf drei Subflows S_0, S_1, S_2 bietet die folgende Formel 6.20, wobei $B_0 \geq B_1 \geq B_2$ ist:

$$E_{eff,max} = \left\{ \begin{array}{ll} \frac{B_{tot}}{R_L+1}, & wenn\ \frac{B_1+B_2}{R_L} \geq B_0 \\ \frac{B_2}{R_L-1}, & wenn\ \frac{B_2}{R_L-1} < B_1\ und\ R_L > 1 \\ \frac{B1+B2}{R_L}, & sonst \end{array} \right\}, R_L \in \mathbb{R} \mid 0 \leq R_L \leq 2 \qquad (6.20)$$

Bei dieser Variante lassen sich drei Fälle unterscheiden:

- Zu Beginn werden alle Subflows für die anfallenden Daten der Applikation genutzt. In Abhängigkeit von R_L werden alle Datensegmente, sowohl primäre als auch redundante, auf drei Subflows verteilt. Solange diese die Datenübertragungsrate aller Daten bedienen können, fällt die Kurve streng monoton.
- Wenn die Datenübertragungsrate der beiden schwächeren Subflows nicht mehr ausreicht, um genügend Datensegmente zu senden, muss die maximale effektive Datenübertragungsrate auf dem schnellsten Subflow abgesenkt werden. Die maximale effektive Datenübertragungsrate wird dann durch den zweitschnellsten Subflow bestimmt.
- Wird die leistungsbezogene Redundanz weiter erhöht, hängt die maximale effektive Datenübertragungsrate vom langsamsten Subflow ab, was durch den dritten Fall beschrieben wird.

Die Übergänge innerhalb der Fallunterscheidung werden ebenfalls als charakteristischer „Knick“ sichtbar. Die folgende Abbildung 6.9 zeigt mehrere Fälle mit verschiedener Heterogenität für drei Subflows S_0, S_1, S_2.

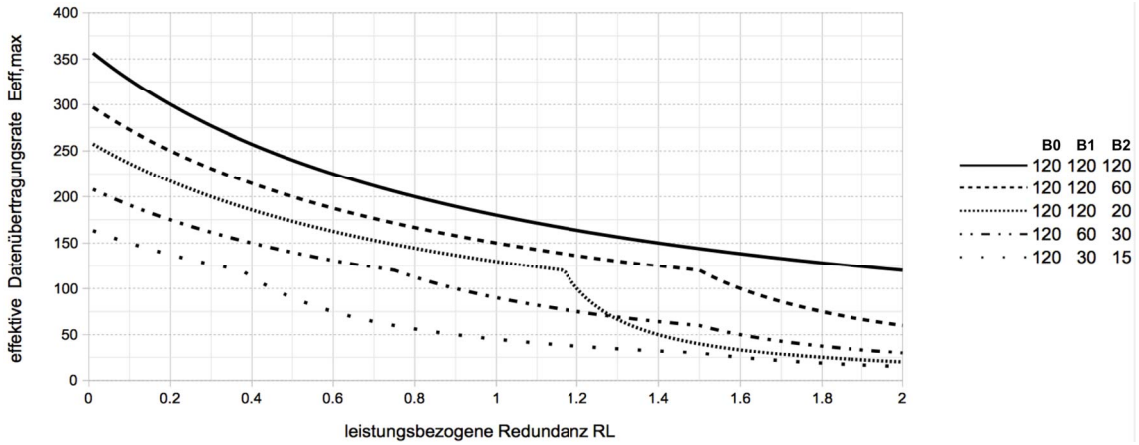

Abbildung 6.9: Redundanz-Datenübertragungsrate bei drei Subflows und durchgehender Redundanz

Eine technische Umsetzung, die eine stufenlose leistungsbezogene Redundanz implementiert, muss in erster Linie auf die zur Verfügung stehende Datenübertragungsrate der einzelnen Subflows eingehen. Die Kontrolle sowie die Überwachung dieser Parameter kann durch den Benutzer oder die Applikation hinsichtlich ihrer Dienstgüteanforderungen übernommen werden. Dies wiederum bietet Möglichkeiten einer Überwachung der Dienstgüte für kritische Applikationen. Bestimmte Grenzwerte sind einstellbar, die z.B. bei Über- bzw. Unterschreitung ein Alarmsignal abgeben können. Eine weitere Möglichkeit ist die automatische Anpassung der Parameter bei Problemen.

Ein Redundanz-Scheduler verteilt Datensegmente und zusätzliche Kopien hiervon in Abhängigkeit von der gewählten rMPTCP-Strategie[37] und der Anzahl der redundant gesendeten Segmente auf die Subflows. Um statistische Abhängigkeiten von Fehlern zu vermeiden, die auf demselben Subflow zu erwarten sind, werden zusätzliche Datensegmente stets auf einem anderen Subflow gesendet, auf dem nicht das primäre Gegenstück gesendet wird.

Eine zu berücksichtigende Dezimalzahl mit Nachkommastelle als leistungsbezogenem Redundanz- bzw. Quotenwert ermöglicht eine flexiblere Anpassung an:

- die Anzahl der verfügbaren Subflows
- die Eigenschaften der Subflows
- die Anforderungen der Applikation

Ein Scheduler, der die Datensegmente auf die verfügbaren Subflows verteilt, kann nur mit ganzzahligen Werten arbeiten, da die zu sendenden Datensegmente nur auf einer ganzzahligen Anzahl von Subflows mehrfach gesendet werden können. Um eine Dezimalzahl mit Nachkommastelle zu erreichen, müssen Datensegmente daher mit unterschiedlichen ganzzahligen Vielfachen versendet werden. Ein gewünschter Zielwert der

[37] Vgl. Kapitel 6.2

leistungsbezogenen Redundanz in Form einer Dezimalzahl R_{target} wird durch den Benutzer oder die Applikation definiert. Hieraus sind ganzzahlige Quotenwerte abzuleiten, die angeben, auf wie vielen Subflows ein aktuelles Datensegment zeitgleich versendet werden soll.

Mithilfe des Zielwerts R_{target} wird ein ganzzahliger Mindestwert durch Abrundung der Dezimalzahl sowie ein ganzzahliger Höchstwert durch Aufrundung berechnet. In Abhängigkeit von R_{actual}, dem momentanen Redundanzwert, gibt dieser Wert den Kontrollwert $R_{control}$ an, der die Anzahl der Subflows angibt, auf denen das nächste Datensegment zeitgleich versendet wird.

$$R_{control} = \begin{cases} \lfloor R_{target} \rfloor, & wenn\ R_{actual} \geq R_{target} \\ \lceil R_{target} \rceil, & wenn\ R_{actual} < R_{target} \end{cases} \tag{6.21}$$

Wenn der momentane Redundanzwert größer oder gleich des erforderlichen Zielwerts ist, wird der Redundanzwert des nächsten Datensegments ganzzahlig verringert. Ist der momentane Redundanzwert kleiner als der erforderliche Zielwert, so wird der Redundanzwert des nächsten Datensegments ganzzahlig erhöht. Auf diese Weise entsteht der in Abbildung 6.10 dargestellte Regelkreis, der die Anzahl der Subflows für das Senden berechnet, um die verlangte Redundanzquote zu erfüllen.

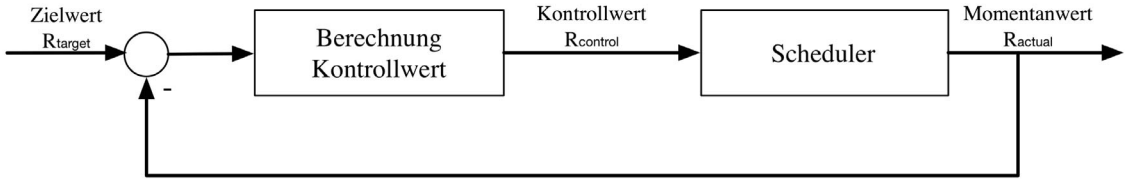

Abbildung 6.10: Berechnung der flexiblen Redundanz

Mit dieser Methode lassen sich feinere Redundanzquotenanpassungen vornehmen und auf Änderungen im Netzwerk eingehen. Der Ausfall eines Subflows führt zu einer veränderten Redundanzquote, wenn die Datenübertragungsrate des primären Datenstroms gehalten werden soll. Hierfür ist der Zielwert R_{target} anzupassen.

Zu beachten ist hier, dass zu sendende Datensegmente in Abhängigkeit von R_L bzw. Q über beliebige, d.h. verfügbare Subflows versendet werden. Bei einer Redundanzquote Q kleiner als die Anzahl der Subflows kann es daher vorkommen, dass Datensegmente ohne leistungsbezogene Redundanz hintereinander auf unterschiedlichen Subflows versendet werden. Eine erhöhte Sendegeschwindigkeit wäre der Vorteil. Nachteilig sind bei heterogenen Subflows Laufzeitunterschiede, die dadurch zu Out-of-Order-Segmenten führen können. Eine Kompensation dieses Problems wird in Kapitel 6.3.2 diskutiert.

6.2.2 Angepasste Redundanz bei Subflow-Ausfall

Wenn sich die Situation im Netzwerk ändert und Subflows von Senkungen der Datenübertragungsrate betroffen sind, müssen die oben beschriebenen Parameter angeglichen werden. Um die Datenübertragungsrate der Anwendung weiterhin zu unterstützen,

kann die Redundanzquote angepasst werden. Inwiefern dies durchgeführt wird, hängt von der Anzahl der verwendeten Subflows ab.

Bei Ausfall eines Subflows müssen die verbleibenden Subflows die Datenlast übernehmen. Sind anfangs nur zwei Subflows gegeben, muss die Redundanz auf $R_L=0$ heruntergefahren werden, da der einzelne verbleibende Subflow die primären Daten übertragen muss. Bei einer größeren Anzahl von Subflows hängt die resultierende Redundanz ab:

- von der Heterogenität der Subflows
- von dem ausfallenden Subflow
- von der benötigten Datenübertragungsrate der Anwendung

Bei drei gegebenen Subflows muss ein Ausfall durch die anderen beiden kompensiert werden. Die Redundanz muss soweit verringert werden, dass die beiden verbleibenden Subflows die von der Applikation geforderte Datenübertragungsrate erfüllen können. Abbildung 6.11 zeigt ein Beispiel mit drei Subflows S_0, S_1, S_2 mit einer Datenübertragungsrate von $B_0=120\ Mbit$, $B_1=60\ Mbit$ und $B_2=20\ Mbit$. Die anfängliche stufenlose leistungsbezogene Redundanz liegt bei $R_L= 1,4$, woraus eine maximal effektive Datenübertragungsrate von $E_{eff,max}= 50\ Mbit$ resultiert. Bei Ausfall eines Subflows sind mehrere Szenarien denkbar. Um die effektive Datenübertragungsrate zu halten, wird R_L, abhängig vom ausgefallenen Subflow, auf *1* bzw. *0,4* verringert.

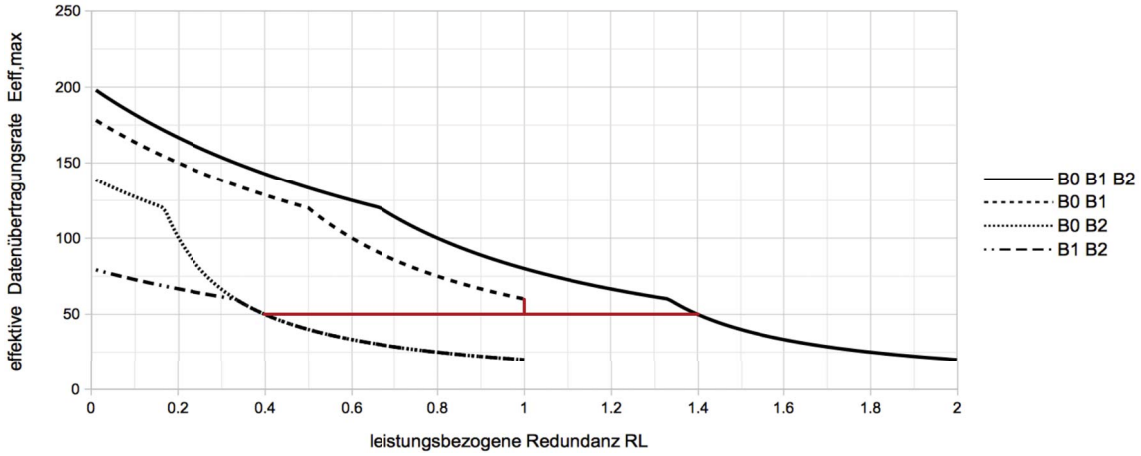

Abbildung 6.11: Redundanz-Datenübertragungsrate bei drei Subflows mit Zurückschaltung auf zwei Subflows

Wie aus der Abbildung hervorgeht, entsteht hierbei kein Unterschied beim Ausfall des sehr viel schnelleren Subflows S_0 im Vergleich mit einem Ausfall von S_1. Ein Ausfall von S_2 ermöglicht bei einer Senkung der Redundanz auf $R=1$ eine größere effektive Datenübertragungsrate $E_{eff,max}$.

Die Datenübertragungsrate der primären Daten ist von der Applikation abhängig. Die Messung hiervon ist essentiell, um eine Umschaltung des Schedulers auf die verbleibenden Subflows zu ermöglichen. Eine Messung der vollständigen Datenübertragungsrate der einzelnen Subflows ist während der normalen Nutzung schwierig, da eine

erschöpfende Messung nötig wäre. Während des Betriebs muss daher die *Auslastung* U_i der einzelnen Subflows erfasst werden.

Die Auslastung gibt die genutzte Datenübertragungsrate eines einzelnen Subflows an. Bei Ausfall eines Subflows muss die vorher genutzte Auslastung durch die anderen Subflows kompensiert werden. Die resultierende leistungsbezogene Redundanz R_{res} kann unter Verwendung des Durchschnittwerts der Gesamtauslastung $U_{tot,mean}$, dem Durchschnittswert der Auslastung des ausgefallenen Subflows $U_{fail,mean}$ sowie der durchschnittlichen effektiven Datenübertragungsrate $E_{eff,mean}$ berechnet werden. Dies geschieht mit der folgenden Formel 6.22:

$$R_{res} = Q_{res} - 1 = \frac{U_{tot,mean} - U_{fail,mean}}{E_{eff,mean}} - 1 \tag{6.22}$$

$$\text{und} \qquad E_{eff} = \frac{U_{tot}}{Q} \tag{6.23}$$

Die Berechnung gibt den minimalen Wert für eine resultierende Redundanz R_{res} aus. Die mögliche Datenübertragungsrate B_i eines Subflows nach einem Ausfall und einer Umverteilung der Datenübertragung auf die verbleibenden Subflows kann jedoch höher sein als die vorherige Auslastung U_i. Die Redundanz kann daher unter Beobachtung der effektiven Datenübertragungsrate E_{eff} und der benötigten Datenübertragungsrate der Anwendung langsam gesteigert werden.

Wird der Ausfall eines Subflows erkannt[38], müssen die Daten neu auf die verbleibenden Subflows verteilt werden. Eine Neuverteilung der Datensendung auf die Subflows kann mithilfe der Datenübertragungskennwerte der vorherigen fehlerfreien Übertragung erfolgen. Hierfür sind während des Betriebs passive Messungen durchzuführen, um die Kennwerte zu ermitteln. Messungen dieser Art sind bereits über den Protokollkern verfügbar. Abbildung 6.12 zeigt den Ablauf einer Redundanzquotenanpassung bei Subflow-Ausfall.

Zunächst muss der Subflow identifiziert werden, der einen Störungszustand hervorgerufen hat. Für die Berechnung der neuen Redundanzquote R_{res} wird Formel 6.22 hinzugezogen, indem die durchschnittliche Auslastung des fehlerhaften Subflows $U_{fail,mean}$ von der Gesamtauslastung $U_{tot,mean}$ subtrahiert und durch die durchschnittliche effektive Datenübertragungsrate $E_{eff,mean}$ geteilt wird.

Es erfolgt eine sich wiederholende Überprüfung der effektiven Datenübertragungsrate E_{eff} und der momentanen leistungsbezogenen Redundanz R_{actual}. Hierfür kommen alle *verfügbaren* und *nutzbaren* Subflows zum Einsatz, um die mögliche Datenübertragungsrate beurteilen zu können. Diese geht nicht zwingend aus den Messungen des vorherigen fehlerfreien Zustands hervor, da möglicherweise keine vollständige Auslastung gegeben war. Durch das Senden von Datensegmenten auf allen Subflows verändert sich die momentane leistungsbezogene Redundanz R_{actual}.

[38] Die Erkennung eines solchen Störungsfalls wird in Kapitel 6.5 entwickelt.

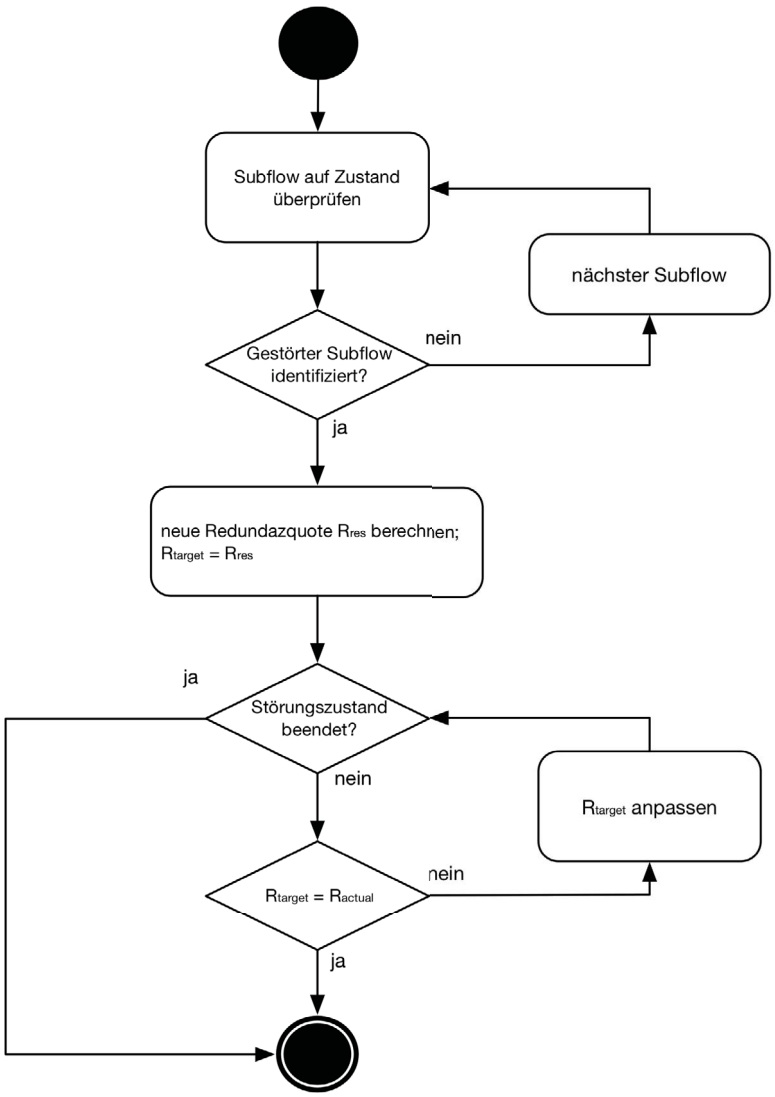

Abbildung 6.12: Aktivitätsdiagramm der adaptiven Redundanz im Störfall

In Abhängigkeit von R_{actual} und der momentanen effektiven Datenübertragungsrate E_{eff} muss R_{target} angepasst werden, um schließlich den neuen möglichen Redundanzwert zu erreichen. Wird die Störungsphase beendet, wird R_{target} auf den alten vom Benutzer bzw. von der Anwendung gewählten Wert zurückgesetzt. Der folgende Pseudocode implementiert den Vergleich und die Anpassung von R_{target}.

```
IF Eeff,mean > 1,05 * Emean,old AND Eeff > 1,05 * Emean,old {
      IF Ractual > Rtarget {
            Rtarget = Rtarget + 0,01;
            Rtarget = Rtarget + (Ractual – Rtarget)/16;
            }
      ELSE
            Rtarget = Rtarget + 0,01;
      }
ELSE
      Rtarget = Rtarget – 0,01;
```

Pseudocode 6.1: Pseudocode der adaptiven Redundanzregulierung

Ist die effektive Datenübertragungsrate größer als *105 %* der effektiven Datenübertragungsrate vor dem Störungsfall $E_{eff,old}$, dann kann R_{target} vergrößert werden.[39] Falls nicht, so wird R_{target} verringert. Für den Fall, dass R_{actual} größer ist als R_{target}, kann R_{target} auch schneller erhöht werden, um die verfügbare Datenübertragungsrate schneller auszunutzen.

6.3 Senden und Empfangen in rMPTCP

rMPTCP ist eine Protokollmodifikation mit verändertem Sende- und Empfangsmuster. Durch die Verwendung mehrerer Subflows müssen zusätzliche Aspekte hinsichtlich der Pufferung auf Sender- und Empfängerseite und des Umgangs mit auf mehreren Subflows empfangenen Datensegmenten beachtet werden. Im Folgenden werden der benötigte Speicherbedarf sowie eine Verbesserung für einen geordneten Empfang von Datensegmenten auf verschiedenen Subflows diskutiert. Bei der Entwicklung der Out-of-Order-Vermeidung hat unterstützend die Abschlussarbeit [Verbunt, 2017] mitgewirkt.

6.3.1 Sende- und Empfangsspeicher in rMPTCP

Sende- und Empfangsspeicher sind für die Absicherung bei Verlust von Datensegmenten zuständig: Ein Sender muss Daten bei Verlust erneut senden, weshalb sie gespeichert werden müssen. Beim Empfänger erzeugt ein verlorengegangenes Datensegment ein Loch im empfangenen Datenstrom. Darauf folgende Daten können solange nicht zur Applikation weitergeleitet werden, bis die fehlenden Daten vorhanden sind. Für diesen Fall benötigt der Empfänger ebenfalls einen Speicher, in dem die Datensegmente solange gesichert werden, bis das verlorengegangene Datensegment eingetroffen ist.

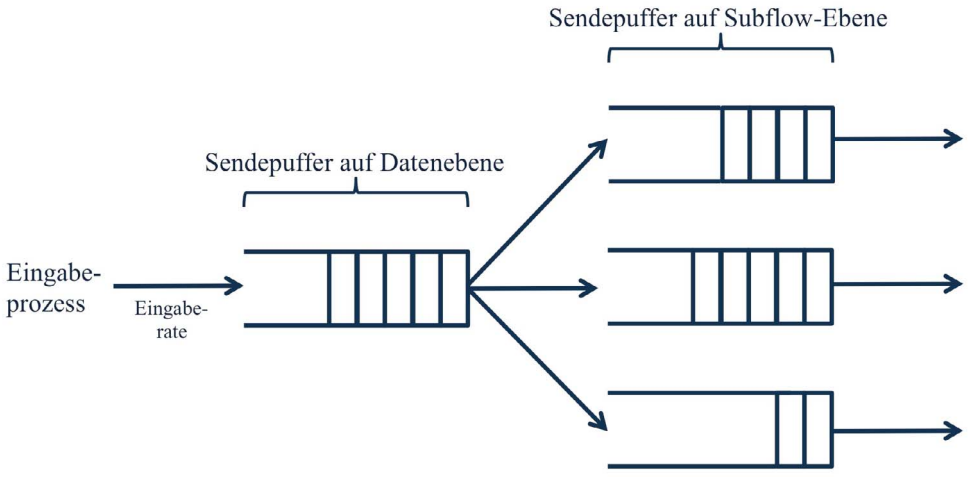

Abbildung 6.13: Sendepuffer in rMPTCP

Ein Sendespeicher speichert zu sendende Daten, bevor sie abgesendet werden und behält sie, bis sie vom Empfänger durch ein ACK-Segment bestätigt werden. Die Größe des Sendepuffers wird durch die Flusskontrolle und das Überlastkontrollprotokoll von TCP definiert. Auf Datenebene existiert unter MPTCP ebenfalls ein Speicher, der die

[39] Die Verwendung von 105% ist als Sicherheitsfaktor zu verstehen, um bei Grenzwerten nicht zu reagieren.

gesendeten Daten vorhält, bis sie den Empfänger erreicht haben. Dieser wird benötigt, wenn ein Subflow ausfällt und die Daten erneut auf einem anderen Subflow gesendet werden müssen oder wenn Datensegmente mit unterschiedlichen Datensequenznummern auf verschiedene Subflows verteilt werden müssen [Barré et al., 2011]. Unter rMPTCP existieren diese Probleme nicht, da die Subflows keine unterschiedlichen Daten versenden müssen. Abbildung 6.13 zeigt ein Schema der vorhandenen Sendespeicher.

Um die notwendigen Funktionen in rMPTCP auszuführen, müssen die Daten des Subflows mit dem größten Datenübertragungs-Verzögerungs-Produkt (Bandwidth-Delay-Product, BDP) gepuffert werden, sodass im Falle eines Ausfalls die Daten auf anderen Subflows gesendet werden können. Dies ist der Fall, wenn der Scheduler nicht mit voller leistungsbezogener Redundanz arbeitet. Der Sendepuffer auf Daten-Ebene wird durch die Standard-Puffer-Empfehlung von TCP abgedeckt [Hunger & Klein, 2016]:

$$SendBuff = 2 \cdot BDP_{Pfad\_Max\_BDP} \tag{6.24}$$

Ein Empfangspuffer muss sicherstellen, dass Datensegmentverluste auf Subflow-Ebene nicht die allgemeine Datenübertragungsrate stören. Vorgegeben durch das Transportschema von TCP wird ein verlorenes Datensegment auf einem Subflow in jedem Fall erneut gesendet – auch wenn es beim Empfänger auf Datenebene nicht mehr benötigt wird. Dies kann im schlimmsten Fall zu einer verzögerten Weiterleitung vom Subflow zur Datenebene führen, da vorher eingetroffene Datensegmente nicht weitergeleitet werden, bis das fehlende Segment eingetroffen ist. Abbildung 6.14 zeigt ein Schema der zur Verfügung stehenden Empfangspuffer.

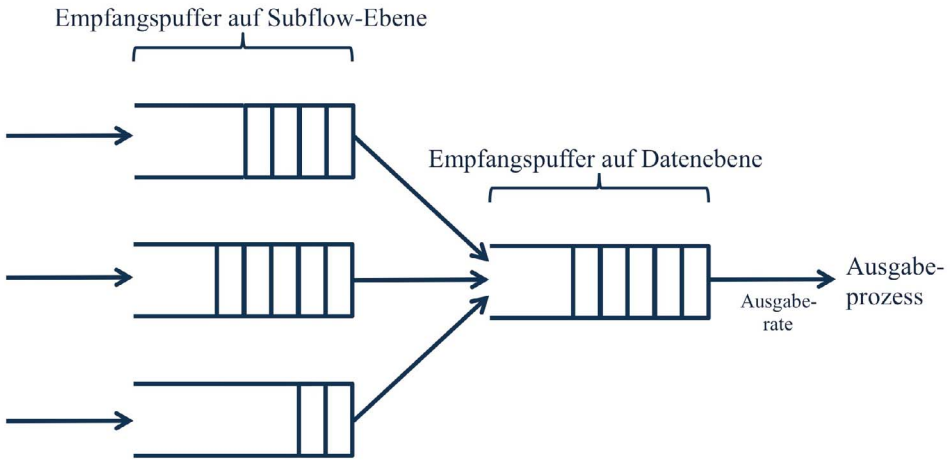

Abbildung 6.14: Empfangspuffer in rMPTCP

Bis zu diesem Zeitpunkt müssen alle vorher eingetroffenen Segmente zwischengespeichert werden. Unter rMPTCP benötigt der Empfangspuffer auf Datenebene eine Größe, die ausreichend ist, um den Pfad mit der kleinsten RTT zu versorgen. Im schlimmsten Fall entsteht ein Segmentverlust mit derselben Datensequenznummer auf allen verfügbaren Subflows. Dann muss zumindest gewartet werden, bis das verlorene

Segment auf dem schnellsten Pfad erneut empfangen wird. Das heißt, dass der Empfangspuffer auf Daten-Ebene ebenfalls die von TCP empfohlene Größe des Empfangsfensters für den schnellsten Pfad besitzen muss [Hunger & Klein, 2016]:

$$RcvBuff = 2 \cdot BDP_{Pfad\_Min\_RTT} \tag{6.25}$$

Die Speichernutzung kann für ein Rechnersystem von Wichtigkeit sein, da hierfür die entsprechenden Ressourcen bereitgestellt werden müssen. Der hier diskutierte Zusammenhang zeigt auf, dass die Speichernutzung unter rMPTCP geringer ist, als es bei MPTCP der Fall ist. Ein MPTCP-kompatibles System besitzt damit unter rMPTCP genügend Ressourcen für die Sende- und Empfangspufferung. Die Ressourcenerfordernisse für rMPTCP werden in dieser Arbeit ansonsten nicht weiter diskutiert.

6.3.2 Vermeidung von Out-of-Order-Datensegmenten

Mehrere Subflows mit heterogenen Pfadeigenschaften können dazu führen, dass Datensegmente, die auf unterschiedlichen Wegen gesendet werden, zu verschiedenen Zeiten ankommen. Hierdurch können Lücken beim Empfang der Datensendung entstehen, obwohl es keinen Datensegmentverlust gegeben hat. Diese Out-of-Order-Segmente führen zu erhöhter Latenz und Jitter, da bereits empfangene Segmente nicht an die Applikation weitergeleitet werden, solange die fehlenden Segmente nicht empfangen wurden.

Wenn Datensegmente auf allen zur Verfügung stehenden Subflows gleichberechtigt übertragen werden, entsteht eine erhöhte Gefahr von Out-of-Order-Segmenten. Die folgende Abbildung 6.15 zeigt das RDS-Diagramm einer beispielhaften Simulation der Empfangsseite mit drei zeitlich heterogenen Subflows:

- Subflow *int1* (rot) wird für das Senden aller Datensegmente eingesetzt;
- die Subflows *int0* (blau) und *int2* (gelb) senden abwechselnd jeweils die halbe Menge der Datensegmente. Damit wird eine leistungsbezogene Redundanz von $R_L=1$ erreicht.

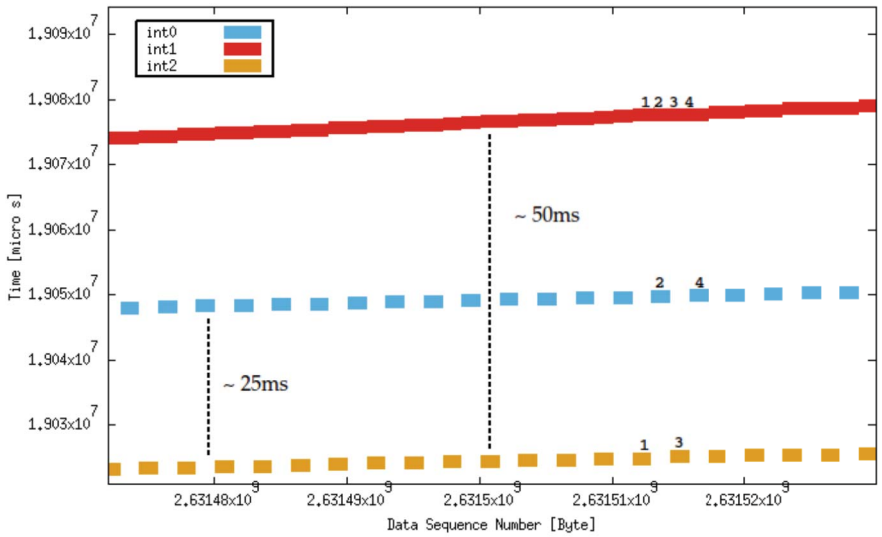

Abbildung 6.15: Erzeugung von Out-of-Order Datensegmenten, [Verbunt, 2017]

Eine mögliche Methode ist die Priorisierung eines oder mehrerer bestimmter Subflows, um möglichst alle primären Datensegmente auf Subflows mit ähnlicher RTT zu senden. Um die Latenz zu optimieren, werden die Subflows mit der geringsten RTT durch eine Kennzeichnung priorisiert. Weitere Subflows werden für den Transport der redundanten Datensegmente verwendet. Ein priorisierter Subflow wird im Rahmen dieser Arbeit als *Main*-Subflow bezeichnet.

Unterstützende Subflows, die zusätzliche redundante Datensegmente übertragen, werden *Support*-Subflows genannt. Datenverluste auf den Main-Subflows werden so durch Segmente auf den etwas langsameren Support-Subflows ausgeglichen. Der zeitliche Abstand zwischen den Subflows sollte so gering sein, dass sie die Latenz einer wiederholten Übertragung unterschreiten.

Wenn Support-Subflows keine primären Datensegmente mehr übertragen, beschränkt sich die Datenübertragungsrate vor allem bei geringer leistungsbezogener Redundanz auf die Main-Subflows *M.* In Hinblick auf die effektive Datenübertragungsrate ändern sich die Formeln 6.19 und 6.20 entsprechend. Bei größer werdender leistungsbezogener Redundanz ist die effektive Datenübertragungsrate vor allem von den schwächeren Subflows abhängig, die mehr und mehr die Last der zugesicherten redundant gesendeten Datensegmente tragen. Im Falle von homogenen Subflows mit gleicher RTT gelten weiterhin die Formeln 6.19 und 6.20.

$$E_{eff,max} = \begin{cases} \min(\sum_{i\epsilon M} B_i \,,\, \frac{B_{tot}}{R_L+1}), & wenn\ \frac{B_1}{R_L} \geq B_0 \\ \frac{B_1}{R_L}, & wenn\ \frac{B_1}{R_L} < B_0 \end{cases} \tag{6.26}$$

$$E_{eff,max} = \begin{cases} \min(\sum_{i\epsilon M} B_i \,,\, \frac{B_{tot}}{R_L+1}), & wenn\ \frac{B_1+B_2}{R_L} \geq B_0 \\ \frac{B_2}{R_L-1}, & wenn\ \frac{B_2}{R_L-1} < B_1\ und\ R_L > 1 \\ \frac{B1+B2}{R_L}, & sonst \end{cases} \tag{6.27}$$

Unter der Annahme, dass die Subflows mit der größten Datenübertragungsrate ebenfalls die Subflows mit der kleinsten RTT sind, gelten die Formeln 6.26 und 6.27.

Abbildung 6.16 zeigt eine Darstellung der effektiven Datenübertragungsrate bei mehreren Fällen mit verschiedener Heterogenität für eine größer werdende leistungsbezogene Redundanz unter der Verwendung von Out-of-Order-Vermeidung. Hierbei werden zwei aus insgesamt drei Subflows als Main definiert. Zu sehen ist die unterdrückte Datenübertragungsrate bei einem geringen Redundanzwert. Bei größer werdender leistungsbezogener Redundanz von R_L*=0,5* entsteht kein Unterschied im Vergleich zu Abbildung 6.9.

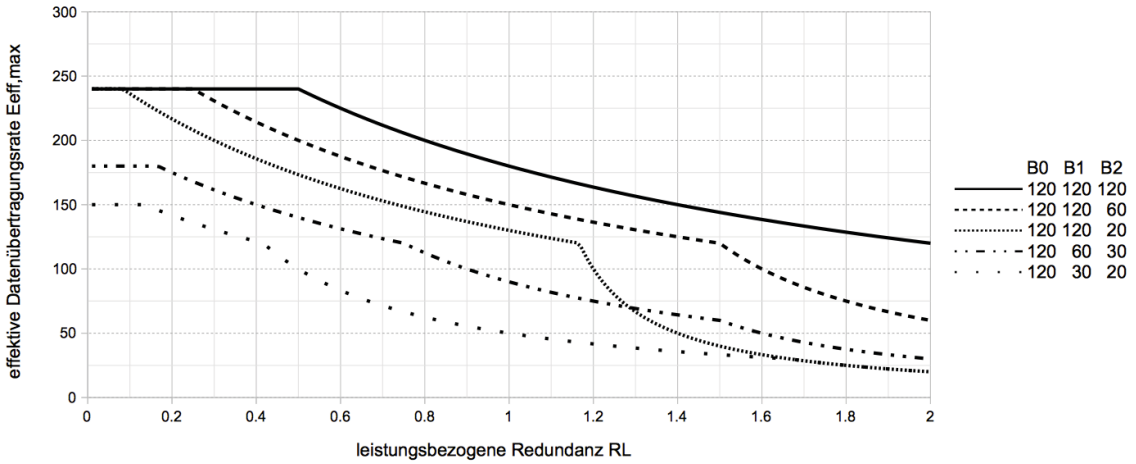

Abbildung 6.16: Redundanz-Datenübertragungsrate bei drei Subflows mit Out-of-Order Vermeidung

Um Out-of-Order Datensegmente zu vermeiden, müssen möglichst alle primären Datensegmente über Subflows mit zumindest ähnlichen Eigenschaften gesendet werden. Die hierfür besten Subflows werden als Main-Subflow markiert. Aus dem Pool der verfügbaren Subflows müssen die Main-Subflows nach bestimmten Kriterien ausgesucht werden:

1. RTT für eine möglichst niedrige Latenzzeit
2. Datenübertragungsrate, die möglichst hoch sein sollte, um den primären Datenstrom selbst und die anderen Support-Subflows nicht unnötig einzuschränken
3. der Subflow darf keiner Störung unterliegen

Um die Kriterien zu erfüllen, werden in allen nicht überlasteten *verfügbaren* Subflows diejenigen ausgewählt, die die niedrigste RTT besitzen und die höchste Datenübertragungsrate B_i haben. Alle Subflows mit diesen Eigenschaften innerhalb eines festgelegten Entscheidungsfensters werden zu Main-Subflows. Die Kriterien, die eine Auswahl ermöglichen, sind durch die Formeln 6.28 und 6.29 gegeben:

$$RTT_i < \delta \cdot RTT_{lowest} \tag{6.28}$$

$$B_i > \frac{B_{highest}}{\delta} \tag{6.29}$$

Das Entscheidungsfenster der zu identifizierenden Main-Subflows lässt sich mithilfe des Koeffizienten δ setzen. Als Standardwert für diesen Schwellenbereich wird $\delta=2$ gewählt: Ein Main-Subflow, der das Zweifache der kleinsten RTT besitzt, muss mehr als die Hälfte der schnellsten Datenübertragungsrate besitzen. Dies schränkt die Auswahl des Main-Subflows nicht zu sehr ein. Wird kein potenzieller Main-Subflow gefunden, werden alle Subflows als Main-Subflow gesetzt. Der Koeffizient δ kann je nach Anforderung durch den Benutzer oder die Anwendung justiert werden.

Ein Subflow wechselt nur dann zu einem Support-Subflow zurück, wenn die Gleichungen 6.28 und 6.29 mit einem um *1* erhöhten Koeffizienten erfüllt werden, was ein hin- und herspringen zwischen Main- und Support-Subflows verhindert:

$$RTT_i < (\delta + 1) \cdot RTT_{lowest} \tag{6.30}$$

$$B_i > \frac{B_{highest}}{\delta + 1} \tag{6.31}$$

Wenn mehr Main-Subflows zur Verfügung stehen, als für eine bestimmte effektive Datenübertragungsrate und die festgelegte leistungsbezogene Redundanz benötigt werden, können diese für eine erhöhte Redundanzquote eingesetzt werden. Sie werden daher als Support-Subflow gesetzt. Überschüssige Support-Subflows werden nicht ausgeschlossen, um eine konstante RTT und höhere Stabilität zu gewährleisten.

Der folgende Pseudocode beschreibt die Umsetzung der Out-of-Order-Priorisierung. Hierbei werden zunächst die kleinste RTT und die schnellste Datenübertragungsrate identifiziert. Danach werden die Main-Markierungen gesetzt bzw. entfernt, wenn die Kriterien zutreffen. Wurde kein Main-Subflow gefunden, der den Kriterien entspricht, werden alle Subflows zu Main-Subflows erklärt.

```
// Subflows mit der kleinsten RTT und der größten
// Datenübertragungsrate (B) identifizieren
FOR (i = 0; i < Anzahl_Subflows; i++) {
      IF (Subflow(i) != Congested) AND (Subflow(i) = enabled) {
            IF (RTT(Subflow(i)) < lowest_RTT)
                  lowest_RTT = RTT(Subflow(i));
            IF (B(Subflow(i)) > highest_B)
                  highest_B = B(Subflow(i));
      }
}
// Setzen der Main-Subflows
FOR (i = 0; i < Anzahl_Subflows; i++) {
      IF (Subflow(i) = enabled) {
            // Main-Markierung wegnehmen
            IF ( (RTT(Subflow(i)) > ((delta+1) * lowest_RTT) ) OR
            ( B(Subflow(i)) < (highest_B/(delta + 1)) ) OR
            ( Subflow(i) = Congested ) {
                  State(Subflow(i)) = Support;
            }

            // Main-Markierung setzen
            ELSE IF ( RTT(Subflow(i)) < (delta * lowest_RTT) )
            AND ( B(Subflow(i)) > (highest_B/delta) ) {
                  State(Subflow(i)) = Main;
                  Anzahl_Main = Anzahl_Main + 1;
            }
            // keine Markierung setzen, trotzdem zählen
            ELSE IF ( State(Subflow(i)) = Main)
                  Anzahl_Main = Anzahl_Main + 1;
      }
}
// Wenn kein Main erstellt werden konnten, alle als Main
IF (Anzahl_Main = 0)
      FOR (i = 0; i < Anzahl_Subflows; i++)
            State(Subflow(i)) = Main;
```

Pseudocode 6.2: Out-of-Order-Priorisierung der schnellsten Subflows

6.4 Pfad-Management in rMPTCP

Die Zusammenstellung der Subflows ist bei der gleichzeitigen Nutzung mehrerer Pfade von hoher Wichtigkeit. Es müssen Techniken entwickelt werden, die eine Auswahl der zu nutzenden Pfade nach bestimmten Kriterien treffen. Die Aspekte des Pfad-Managements betreffen bei der Entwicklung des redundanten Mehrwegprotokolls die folgenden Felder

- Pfadnutzen sowie Homogenität der Pfade
- Gemeinsame Nutzung von Ressourcen bzw. Pfad-Diversität
- Ausfallsicherung von Subflows

6.4.1 Pfadnutzen und Auflagen der Pfadauswahl

Wenn unter rMPTCP ein Segmentverlust auftritt oder ein Segment unerwartet mehr Zeit benötigt, können die benötigten Datensegmente von einem anderen Subflow entnommen werden. Ein deutlicher zeitlicher Unterschied zwischen den Pfaden kann jedoch dazu führen, dass kein Nutzen aus der Pfad-Diversität gezogen werden kann.

Verschiedene Pfade besitzen stets unterschiedliche OWDs. Abbildung 6.17 zeigt ein Sequenzdiagramm mit extremem Laufzeitunterschied. Die von zwei Subflows genutzten Pfade liegen zeitlich so weit auseinander, dass eine erneute Sendung im Falle einer Störung von Subflow 1 weniger Zeit benötigt als die reguläre Sendung auf Subflow 2. Der zeitliche Unterschied zwischen den Pfaden muss daher kleiner sein, als der Zeitaufwand für eine erneute Datensendung im Falle eines Timeouts benötigt. Das heißt, wenn eine erneute Datensendung schneller durchführbar ist, als der Vorgang für ein auf einem anderen Subflow repliziertes Datensegment benötigt, kann die Latenzzeit nicht geglättet werden.

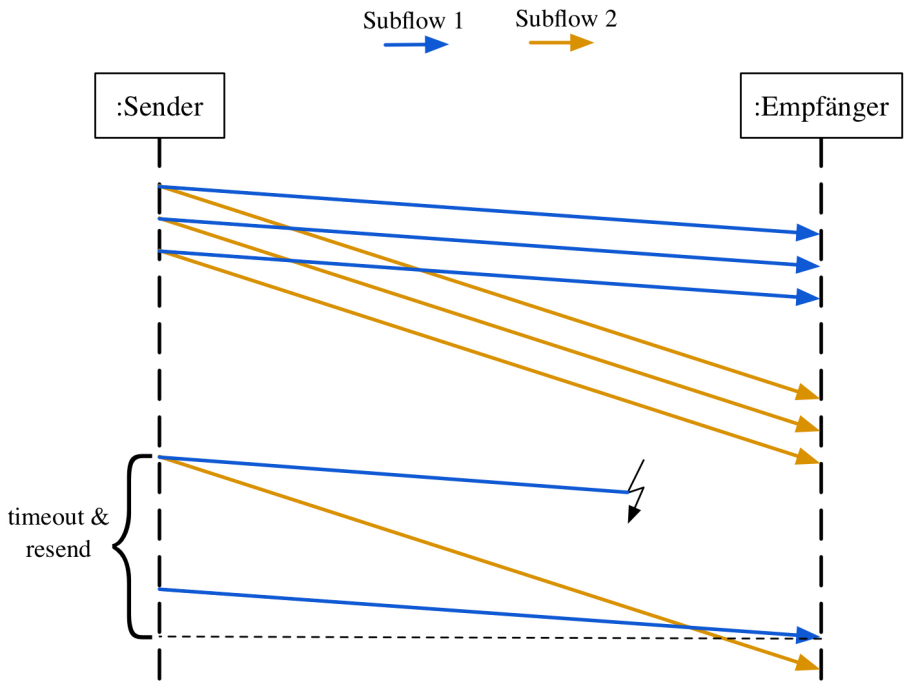

Abbildung 6.17: Sequenzdiagramm einer Kommunikation bei zu starker Heterogenität

Die Verwendung von mehreren Subflows, die Datensegmente redundant senden, kann bei zu hoher Pfad-Heterogenität nicht mehr zur Angleichung der Latenzzeit benutzt werden, sondern nur noch im Falle eines Subflow-Ausfalls für einen Notfall-Failover. Hieraus lässt sich die *Latenzangleichungs-Grenze* mit den folgenden Eigenschaften definieren [Hunger & Klein, 2016]:

$$OWD_{Subflow2} < Timeout_{Subflow1} + OWD_{Subflow1} \tag{6.32}$$

Falls ein Subflow diese Grenze unterschreitet, kann er noch als Backup-Subflow verwendet werden, der die Übertragung im Falle eines Komplettausfalls übernimmt. Ein Backup-Subflow muss jedoch weiterhin die Datenübertragungsrate der Applikation unterstützen können.

Bei Versagen eines Subflows wird ein noch verbleibender Subflow automatisch als primäre Datenverbindung genutzt. Die zeitliche Dauer für diesen Failover, die beim Empfänger als Verzögerung wahrgenommen wird, bestimmt sich durch den zeitlichen Unterschied zwischen dem versagenden Subflow und dem Verbleibenden mit der kleinsten OWD. Ein Zeitunterschied, der größer ist als die Zeit für den Aufbau einer neuen Verbindung zwischen den beiden Endpunkten zuzüglich der Zeit, um die verlorengegangenen Datensegmente erneut zu senden, ist für den Empfänger untauglich. In diesem Fall hat rMPTCP keinen Nutzen mehr. Die folgende Gleichung muss daher zutreffen [Hunger & Klein, 2016]:

$$OWD_{Subflow2} - OWD_{Subflow1} < Handshake_{TCP} + Resend_{lost\ Segments} \tag{6.33}$$

Die Zeit für einen Failover ist dann Null, wenn die Pfadeigenschaften zweier Subflows homogen sind.

6.4.2 Pfadauswahl unter rMPTCP

Das Einrichten von Subflows wird zwischen den verschiedenen Netzwerkschnittstellen (Network Interface Card, NIC) der Endgeräte durchgeführt. Unter MPTCP werden standardmäßig alle verfügbaren Subflows erschlossen [Paasch, 2014]. Dies bedeutet, dass alle NICs mit allen anderen NICs der Gegenstelle potenzielle Subflows bilden. Abbildung 6.18 zeigt eine Darstellung des sogenannten Fullmesh-Managements mit drei NICs und neun daraus resultierenden Subflows.

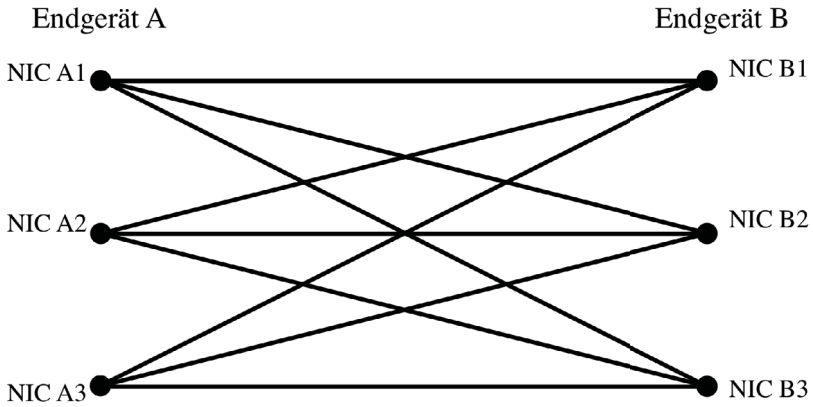

Abbildung 6.18: MPTCP Fullmesh-Pfad-Management

Bei einer Nutzung aller verfügbaren Subflows überschneiden sich die Subflows in Hinblick auf den genutzten Pfad. Sie bieten so nur partielle Vorteile, da die Datenübertragungsrate des einen Subflows die Datenübertragungsrate des anderen durch die gemeinsame Ressourcennutzung vermindert. Ein erhöhter Overhead auf den genutzten Pfaden senkt den Datendurchsatz. Statistische Abhängigkeiten erhöhen die Wahrscheinlichkeit einer korrelierten Störung und sind daher für rMPTCP ungeeignet.

Eine optimale Nutzung für rMPTCP ist die in Abbildung 6.19 dargestellte Eins-zu-Eins-Verbindung. Darin werden die nutzbaren NICs stets einmalig genutzt. Ein automatisches Pfad-Management sucht die am besten passenden Subflows unter der Vermeidung doppelter NIC-Nutzung heraus. Hierfür müssen alle Pfadkombinationen ohne gemeinsame Netzwerkschnittstellen für eine parallele Nutzung gefunden werden. Weitere Subflows können als Backup-Subflow offen gehalten werden und dienen als Ersatz im Falle einer Störung.

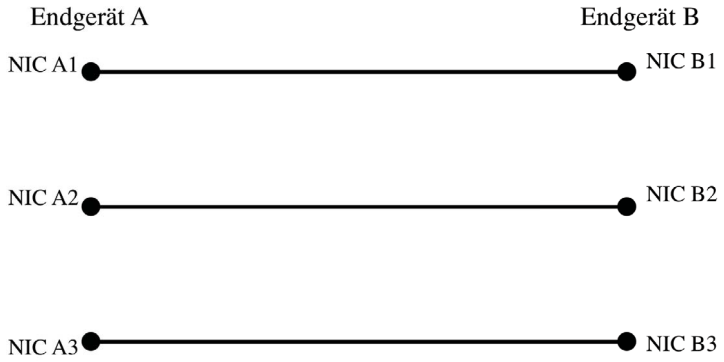

Abbildung 6.19: rMPTCP Eins-zu-Eins Pfad-Management

Für eine Verbindung unter rMPTCP ist eine Auswahl aus den zur Verfügung stehenden Pfaden zu treffen. Für die im Kontext dieser Arbeit beschriebenen Anwendungen sind die folgenden Kriterien ausschlaggebend:

- Datenübertragungsrate
- RTT

6.4.3 Adaptiver Pfadausgleich

Ein adaptives Pfad-Management kann dazu beitragen, die Redundanz und Datenübertragungsrate nach einem Subflow-Ausfall möglichst konstant zu halten. Die Verfügbarkeit aller nutzbaren Subflows eines Fullmesh-Managements kann in Überlastsituationen dafür verwendet werden, den Ausfall eines Subflows zu kompensieren. Bei der Entwicklung des adaptiven Pfadausgleichs hat unterstützend die Abschlussarbeit [Verbunt, 2017] mitgewirkt.

Abbildung 6.20 zeigt eine Eins-zu-Eins-Verbindung, bei der der Pfad zwischen den Netzwerkanbindungen NIC-A2 und NIC-B2 ausfällt (rot). Alternative Subflows können keine Eins-zu-Eins-Verbindung mehr bieten, da die anderen NICs bereits unter Verwendung stehen und selbst Eins-zu-Eins-Verbindungen unterhalten. Neue Subflows können also nicht mehr voneinander statistisch unabhängig sein.

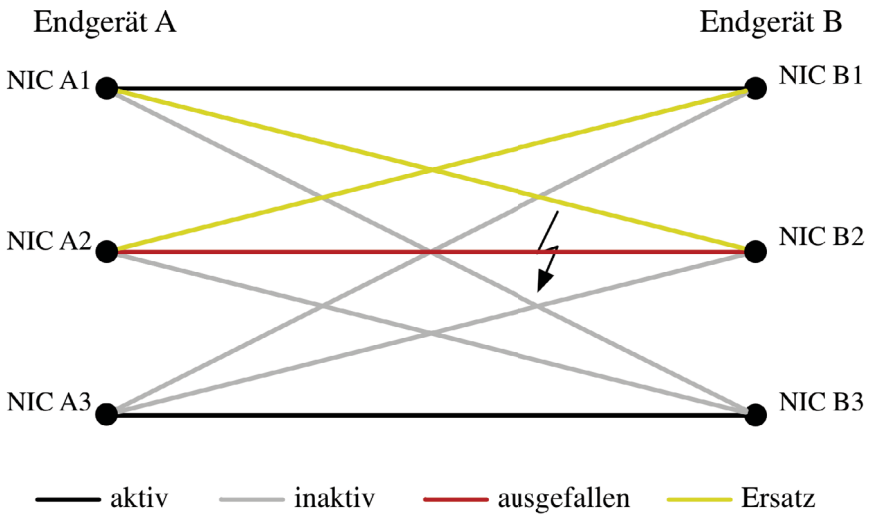

Abbildung 6.20: Subflow-Ausgleich bei Subflow-Ausfall

Die Quelle eines Fehlers, der zum Ausfall eines Subflows geführt hat, ist nicht ohne weiteres lokalisierbar und liegt an unbekannter Stelle auf der Netzwerkstrecke des ausgefallenen Subflows. Eine Möglichkeit ist in einem solchen Fall die Verbindung der beiden NICs zu nutzen, deren Subflow ausgefallen ist, und zwar mit jeweils einem anderen NIC des jeweils anderen Endgeräts (gelb). Dies bietet zwar keine vollständige PD, ein Fehler im Netzwerk wird jedoch mit hoher Wahrscheinlichkeit mit zumindest einem der neuen Subflows umgangen.

Bereits existierende Subflows können durch die neuen Subflows beeinträchtigt werden, da sie einen Teil des Pfades miteinander teilen. Der Grad der Beeinträchtigung ist abhängig von der ausgenutzten Datenübertragungsrate des versagenden und der noch funktionierenden Subflows: Fällt ein Subflow mit niedriger Datenübertragungsrate aus, so ist der Einfluss auf die anderen Subflows geringer, wenn ein Ersatz-Subflow hinzugefügt wird. Um möglichst an dieselbe Datenübertragungsrate des früheren Subflows anzuknüpfen, müssen Ersatz-Subflows vorzugsweise mit NICs verbunden werden, die die höchste Datenübertragungsrate bieten.

Ein Daten-Scheduler mit bereits diskutierter Konfiguration[40] versucht die maximal mögliche Datenübertragungsrate zu ermitteln und auszunutzen. Die Datenübertragungsraten eines neu erstellten Subflows B_{neu} sowie des bereits aktiven Subflows B_{alt} lassen sich ermitteln, indem die Datenübertragungsrate des noch funktionierenden Subflows B_{alt} halbiert wird unter der Voraussetzung, dass B_{neu} von dem Netzwerkinterface unterstützt wird.

$$B_{neu} = \frac{B_{alt}}{2} \tag{6.34}$$

In Abbildung 6.21 ist der Ausfall eines von drei Subflows S_0, S_1, S_2 mit Datenübertragungsraten von B_0=*120 Mbit,* B_1=*60 Mbit* und B_2=*20 Mbit* dargestellt. Der ausgefallene Subflow wird durch einen neuen nach der beschriebenen Methode ersetzt. Hierbei teilt sich der neue Subflow den Pfad mit dem Subflow mit der größten Datenübertra-

[40] siehe Kapitel 6.2

gungsrate. Die durchgehende Linie stellt einen voll funktionstüchtigen Subflows dar. Der Verlust eines Subflows führt im Falle einer leistungsbezogenen Redundanz von $R_L < 1{,}7$ zu einem Verlust an effektiver Datenübertragungsrate, die mit Herabsenkung der Redundanz ausgeglichen werden kann. War die leistungsbezogene Redundanz $R_L > 1{,}7$, so kann sie in allen Fällen bei gleichbleibender effektiver Datenübertragungsrate gehalten werden. Dagegen müsste bei der Benutzung einer reinen Redundanzanpassung ohne Pfadausgleich auf eine leistungsbezogene Redundanz von $R \leq 1$ heruntergeschaltet werden.

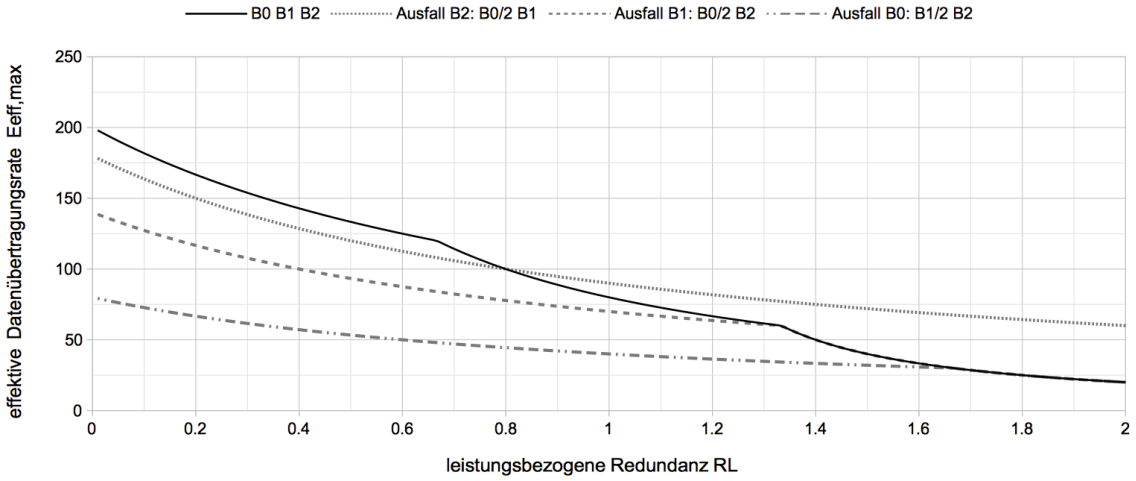

Abbildung 6.21: Redundanz-Datenübertragungsrate-Diagramm: Pfad-Ausgleich mit einem von drei ausgefallenen Subflows

Die Nutzung des adaptiven Pfadausgleichs in Kombination mit einem automatischen Abgleich der leistungsbezogenen Redundanz ermöglicht in den meisten Fällen einen kombinierten Ausgleich für eine gesicherte Datenübertragungsrate. Allerdings ist die maximale effektive Datenübertragungsrate nur kleiner oder gleich aller disjunkten Subflows, die verfügbar sind. Unter bestimmten Umständen muss die Datenübertragungsrate ebenfalls gedrosselt werden. Dies ist dann der Fall, wenn die leistungsbezogene Redundanz vor dem Ausfall bereits so gering ist, dass eine Herunterschaltung von R nicht ausreicht, um nach einem Ausfall und dem erfolgten Ausgleich dieselbe effektive Datenübertragungsrate zuzusichern.

Ersatz-Subflows kommen als normale *verfügbare* Subflows zum Einsatz. Wenn der Störungszustand des ersetzten Subflows in den Normalzustand (*Recovery-Modus*[41]) zurückkehrt, werden die *Ersatz-Subflows* in den *nutzbar*-Modus geschaltet. Abbildung 6.22 zeigt ein Aktivitätsdiagramm für das adaptive Pfadmanagement.

Wie bereits erwähnt, ist es wahrscheinlich, dass sich einer der neuen Ersatz-Subflows ebenfalls im Störungszustand befindet, da er Teile desselben Pfads des ausgefallen Subflows nutzt. Dieser Subflow würde mit seinen wiederholt gesendeten Segmenten die Verbindung der anderen Subflows beinträchtigen. Aus diesem Grund darf ein Ersatz-Subflow im Störungszustand nur ein einziges unbestätigtes Datensegment auf dem Netzwerkpfad besitzen. Dies bietet den Vorteil, dass ein Ersatz-Subflow nur

[41] siehe Kapitel 6.5.1

minimal genutzt wird, wenn er unter Datensegmentverlust leidet, aber weiterhin kontinuierlich überwacht und bei Änderung der Situation für weitere Operationen genutzt werden kann.

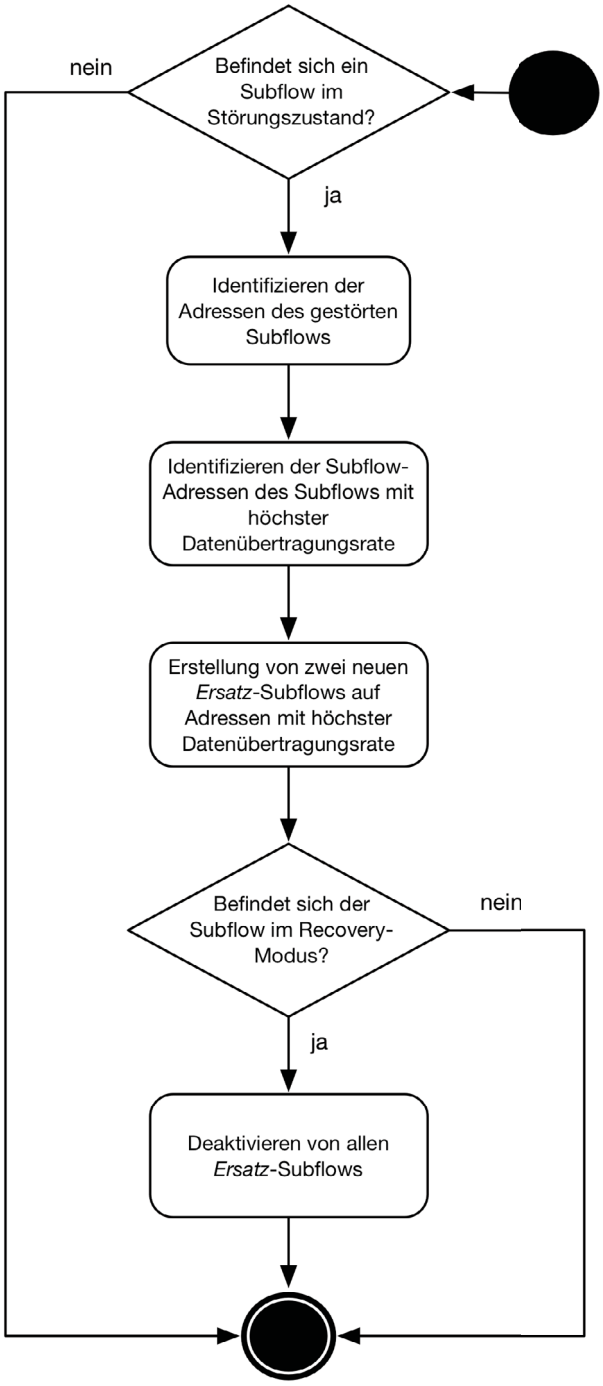

Abbildung 6.22: Aktivitätsdiagramm des adaptiven Pfad-Managements

Die Datenübertragungsrate der neu hinzugekommen Subflows ist anfangs unbekannt. Bei der Benutzung des adaptiven Pfad-Managements in Kombination mit der adaptiven Redundanzkontrolle wird die Redundanz daher zunächst heruntergeregelt. Durch das Einspringen der Ersatz-Subflows und durch eine selbstgeregelte Anpassung an die neue Situation wird die Datenübertragungsrate neu geregelt und eine Redundanzanpassung kann erneut stattfinden.

6.5 Störungserkennung

Eine Schlüsselfunktion der Algorithmen zur effektiven Anpassbarkeit ist die Erkennung von Störungszuständen. Eine Störung ist dann gegeben, wenn die Datenübertragungsrate nicht mehr länger durch die geschalteten Subflows und die verwendete leistungsbezogene Redundanz bedient werden kann. Dies ist dann der Fall, wenn ein Subflow ausfällt oder der genutzte Pfad aufgrund einer Verschlechterung der Situation im Netzwerk (z.B. Datenverluste) nicht mehr die vorgesehene Datenübertragungsrate unterstützt.

Methoden zur Messung von Störungen und ein darauf angepasstes Sendeverhalten sind seit der Entwicklung der ersten Internetprotokolle bis heute ein intensiv erforschtes Feld.[42] Eine Erkennung der Verbindungsqualität wird vor allem bei Überlastkontrollmechanismen eingesetzt. Diese erlauben jedoch nur ein relativ langsames Erkennen und keine rasche Anpassung an den vorliegenden Pfad. Die Umschaltung der Redundanzquote unter rMPTCP muss jedoch innerhalb kürzester Zeit erfolgen.

Die Erkennung einer veränderten Situation im Netzwerk unter rMPTCP kann unter verschiedenen Gesichtspunkten mithilfe unterschiedlicher Methoden durchgeführt werden. Laborversuche müssen zeigen, wie sich die Methoden im Einzelfall verhalten. Im Folgenden werden drei Methoden vorgestellt, die für rMPTCP erprobt werden können und im Zuge dieser Arbeit für die Algorithmen genutzt werden.

1. Die in Kapitel 3.3.6.3 vorgestellten Verfahren unter TCP zielen darauf ab, das Sendeverhalten auf veränderte Situationen im Netzwerk anzupassen. Eine Möglichkeit Störungen zu erkennen ist es, die Überlastkontrolle von TCP zu beobachten und sich daran anhand der Zustände zu orientieren. Ist eine Überlastsituation gegeben, kann rMPTCP diese auslesen und darauf reagieren. Allerdings ist eine Überlastsituation auf einem Subflow kein hinreichendes Indiz für die Unterschreitung der geforderten Datenübertragungsrate oder der leistungsbezogenen Redundanz, da ein Subflow dergestalt betroffen sein kann, dass trotzdem alle Bedingungen erfüllt werden können.

2. Eine weitere Methode ist die stetige Überwachung der momentanen effektiven Datenübertragungsrate E_{eff}. Sobald diese einen bestimmten Schwellwert unterschreitet, kann der Angleichungsmechanismus gestartet werden. Dieser Schwellwert kann mithilfe der durchschnittlichen effektiven Datenübertragungsrate $E_{eff,mean}$ berechnet werden. Ein Vergleich der beiden Werte erlaubt es, eine Beeinträchtigung zu erkennen.

3. Eine dritte Methode besteht darin, die Sendefrequenz bzw. die Sendeperiode der abgesendeten Datensegmente auf den verschiedenen Subflows zu überwachen. Ändert sich die Sendefrequenz der Segmente über einen bestimmten Schwellwert hinaus, dann kann dies auf eine Verschlechterung der Situation im Netzwerk bzw. auf einen Subflow-Zusammenbruch hinweisen. Die verbleibenden Subflows benötigen eine größere Zeit, um ein weiteres Datensegment inklusive redundanter Datensegmente zu senden. Die Sendefrequenz muss hierbei am Ausgangspunkt des Subflows überprüft werden. Wei-

[42] Vgl. Kapitel 3.3.6.3

chen diese Werte von der Dauer des Sendens eines Datensegments ab, so wird ein Timeout ausgelöst.

Ein Mechanismus, der Störungen auf einem der Subflows frühzeitig erkennt, darf keinen Fehlalarm auslösen und so die Verbindung zu einem plötzlichen Wechsel zwingen, obwohl die Situation noch in Ordnung ist. Eine Kombination mehrerer Methoden erscheint daher sinnvoll. Das Kontextdiagramm in Abbildung 6.23 zeigt ein die verschiedenen Methoden zur Störungserkennung. Der Überlastzustand wird zu gegebenen Zeiten abgefragt, wohingegen die anderen Methoden kontinuierliche Überprüfungen erlauben.

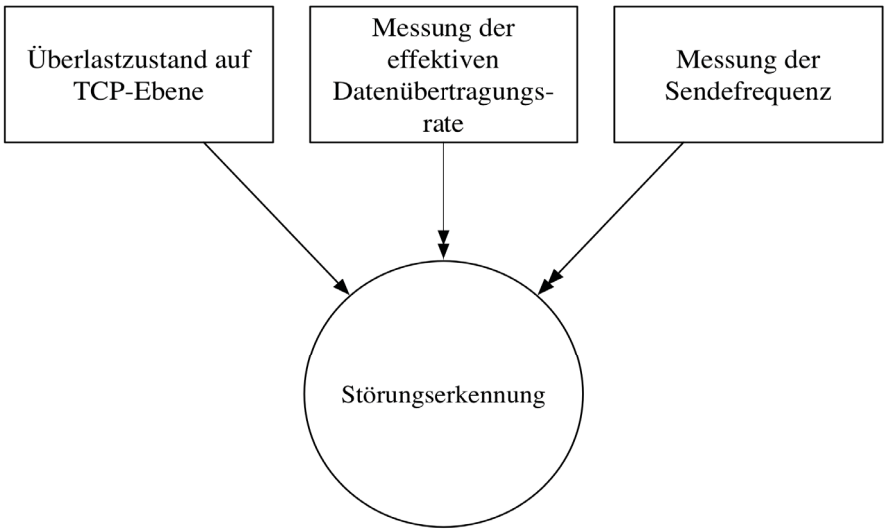

Abbildung 6.23: Kontextdiagram der Störungserkennung

Weitere Methoden sind denkbar, würden jedoch tiefergehende Studien benötigen, um die Effizienz, die Genauigkeit und vor allem die Geschwindigkeit der Erkennung zu bestimmen. Im Rahmen dieser Arbeit wird stattdessen eine Version implementiert, in der die drei beschriebenen Methoden kombiniert und deren Resultate studiert werden. Bei der Entwicklung einer Störungserkennung hat die Abschlussarbeit [Verbunt, 2017] unterstützend mitgewirkt.

6.5.1 Zustände der Störungserkennung

Um die zustandsabhängige Aktivierung der in diesem Kapitel entwickelten Algorithmen zu ermöglichen, wird das folgende Zustandsdiagramm in Abbildung 6.24 umgesetzt. Die Definition von Zuständen ermöglicht es, Übergänge in Störungszustände und die Rückkehr daraus genau zu bestimmen. Jeder Zustand erwirkt definierte Vorgehensweisen.

Beim Zustandsdiagramm werden vier Zustände unterschieden. Jeder Zustand steht für eine bestimmte Netzwerksituation, die unterschiedliche Algorithmen erfordert.

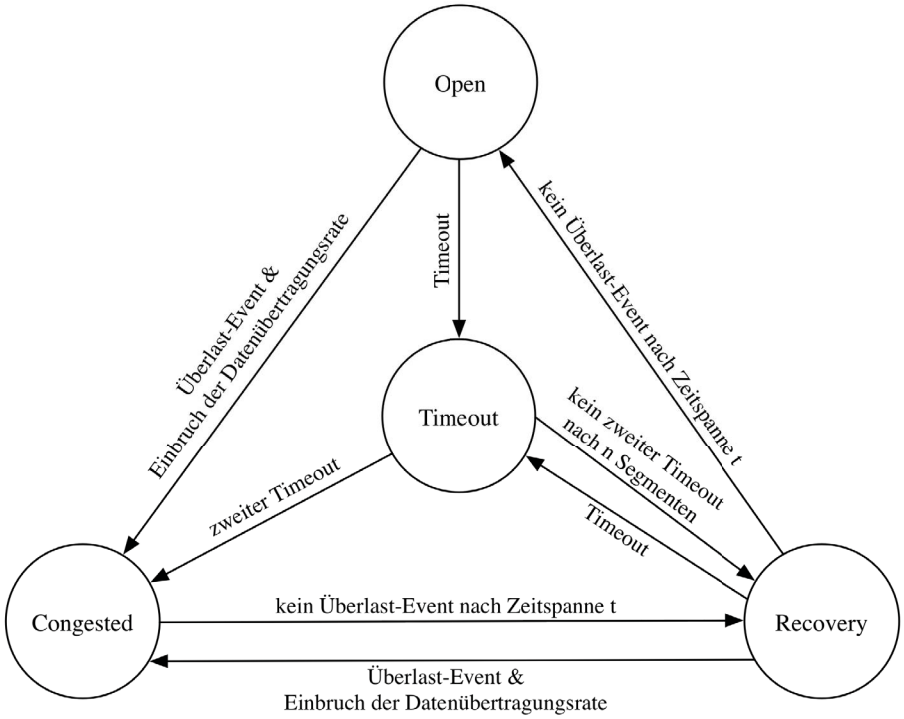

Abbildung 6.24: rMPTCP Zustandsdiagramm der Störungserkennung

*Open (*Abbildung 6.25*):* rMPTCP operiert im Normalzustand. Die Verbindung ist fehlerfrei, die gewünschte Datenübertragungsrate mit den definierten leistungsbezogenen Redundanzansprüchen oder darüber hinaus kann erfüllt werden. Ein Wechsel geschieht, wenn ein Subflow eine bestimmte Zeit nicht senden kann oder wenn die Datenübertragungsrate sinkt.

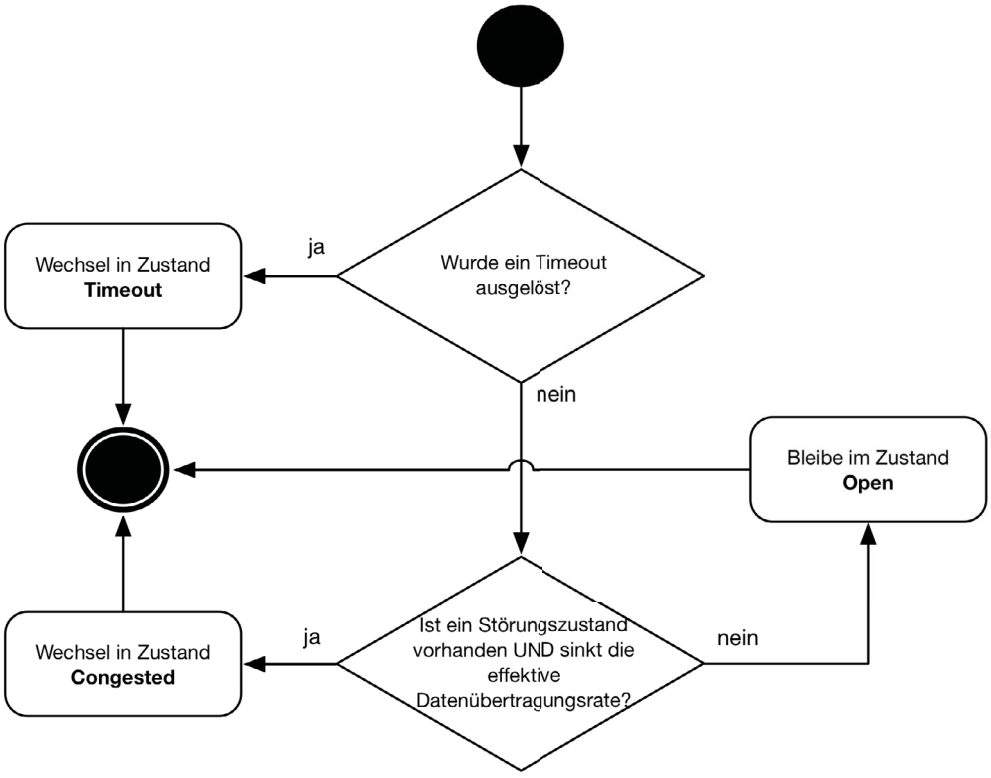

Abbildung 6.25: Aktivitätsdiagramm des rMPTCP Störungszustands „Open"

*Congested(*Abbildung 6.26*):* Einer oder mehrere Subflows sind von Überlast betroffen und können den gesetzten Ansprüchen an die Datenübertragungsrate nicht mehr genügen. Dieser Zustand wird durch mehrere Übergänge erreicht, die einen Störungsfall prognostizieren: Ein Störungsfall wird durch die Störungserkennungsalgorithmen von rMPTCP erkannt, d.h. ein Störungszustand ist auf dem Subflow vorhanden und die Datenübertragungsrate konnte nicht wie geplant erfüllt werden sowie zwei hintereinander folgende Timeouts werden durch ein zu langsames Senden der Daten ausgelöst. Bei diesem Zustand werden die adaptiven rMPTCP-Algorithmen zur Pfad- und leistungsbezogenen Redundanzanpassung aktiviert. Der *Recovery*-Zustand wird ausgelöst, wenn sich die Situation verbessert.

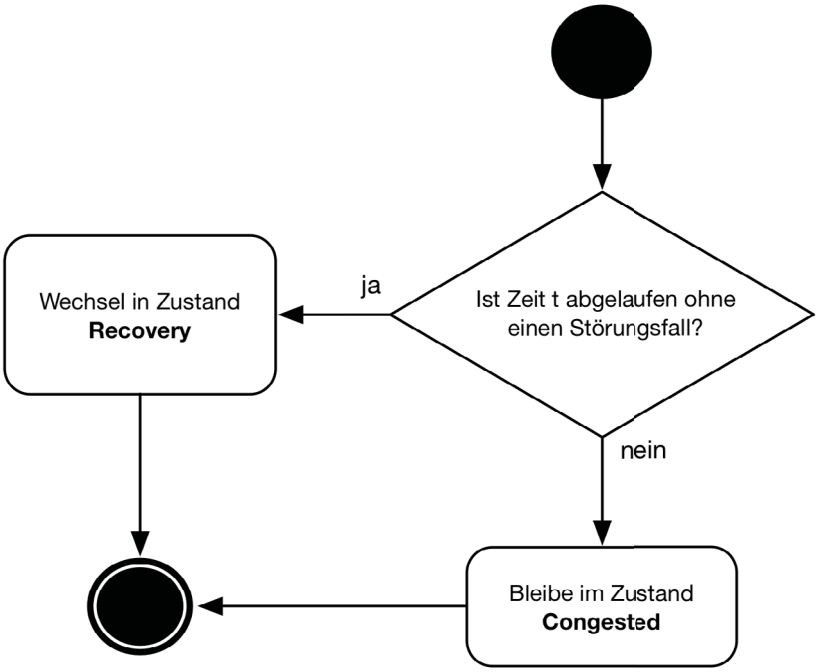

Abbildung 6.26: Aktivitätsdiagramm des rMPTCP Störungszustands „Congested"

*Recovery (*Abbildung 6.27*)*: Nachdem ein Überlastzustand erreicht wurde und sich die zugrundeliegende Verbindung wieder erholt hat, wird zunächst nach einer festgelegten Zeitspanne in den *Recovery*-Zustand gewechselt. In diesem Zustand müssen die adaptiven Algorithmen von rMPTCP wieder in den Normalzustand wechseln. Nach einer weiteren vordefinierten Zeitspanne wird wieder in den *Open*-Zustand gewechselt. Das Auftreten eines Störungsfalls führt zum Wechsel in den *Congested*-Zustand.

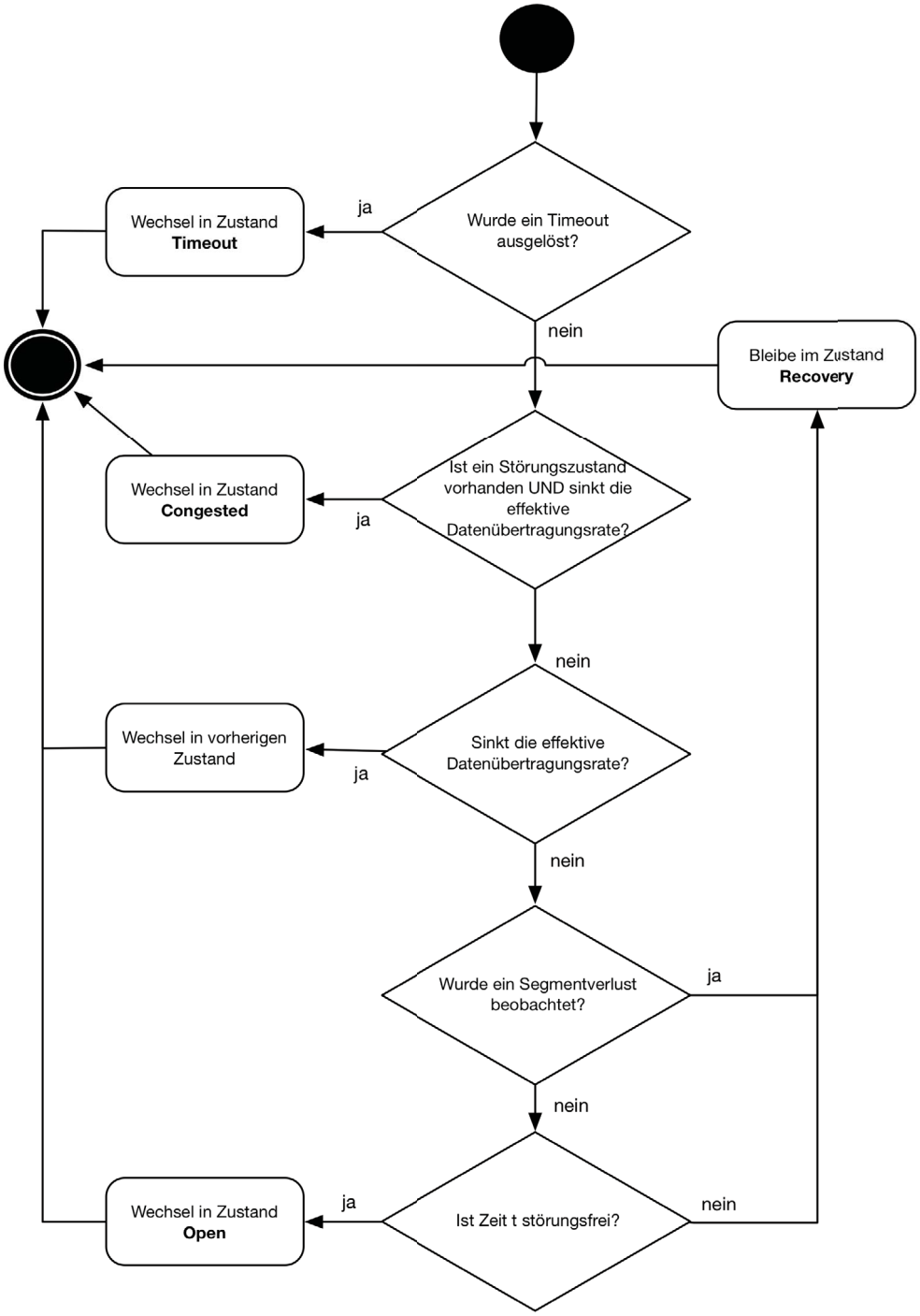

Abbildung 6.27: Aktivitätsdiagramm des rMPTCP Störungszustands „Recovery"

*Timeout (*Abbildung 6.28*)*: Ein Timeout tritt auf, wenn die zu sendenden Datensegmente nicht ausreichend auf den Pfaden verteilt werden konnten und die Zeit des zu sendenden Segments einen bestimmten Schwellwert überschreitet. Tritt dieses Timeout erneut auf, wird in den Zustand *Congested* gewechselt. Entsteht kein erneutes Timeout nach einer bestimmten Anzahl an Datensegmenten (hier $n = 20$), wird in den Zustand *Recovery* gewechselt.

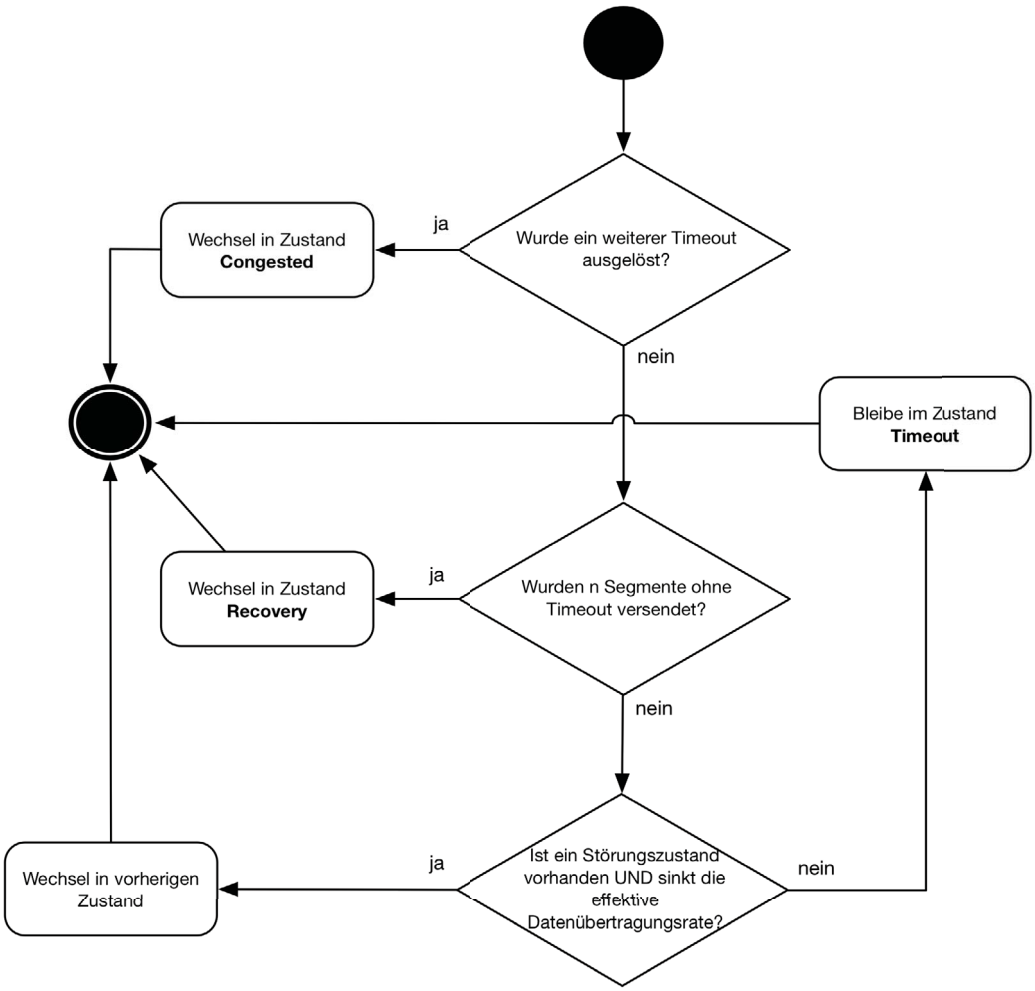

Abbildung 6.28: Aktivitätsdiagramm des rMPTCP Störungszustands „Timeout"

Um den Störungszustand zu verlassen, muss der Scheduler erkennen können, ob und wie der Subflow wieder einsatzbereit ist. Eine Messung kann nicht durchgeführt werden, solange der Subflow keinerlei oder nur unregelmäßig Daten überträgt. Wenn der Subflow den Störungszustand verlässt, ist es möglich, dass er andere Pfadeigenschaften besitzt, als dies vorher der Fall war.

Ein störungsbehafteter Subflow bleibt aktiv und wird weiterhin für das Senden von redundanten Daten benutzt. Somit kann der Überlast-Status der TCP-Überlastkontrolle beobachtet und ein wiederhergestellter Zustand erkannt werden. Der Störungszustand des Subflows wird dann verlassen, wenn der Überlast-Status als fehlerfrei gilt und für eine bestimmte Zeitdauer beibehalten wird.

6.5.2 Störungserkennung durch Beobachtung der Sendeperiode

Die Beobachtung der Sendeperiode ist eine Möglichkeit ein verändertes Sendeverhalten im Störungsfall zu erkennen. Hierfür wird die aktuelle Sendeperiode mit der durchschnittlichen Sendeperiode eines bestimmten Zeitraums verglichen. Wird eine Abweichung wahrgenommen, die einen bestimmten Schwellwert überschreitet, kann dies als Störungsfall interpretiert werden. Diese Möglichkeit der Störungserkennung ist vor allem bei Massendatenübertragung oder bei regelmäßigen Übertragungsarten wie Telemetrie sinnvoll, da hier ein gegebener Abstand zwischen gesendeten Datensegmenten vorhanden ist.

Das normale Sendeverhalten einer TCP-Verbindung ist in der Regel bereits Schwankungen in der Sendeperiode unterworfen. Um eine geeignete Methode zur Berechnung eines zuverlässigen Parameters für Schwankungsbreiten zu entwickeln, sollten zunächst maximale Sendeperioden ohne Störungsfall identifiziert werden, um außergewöhnlich hohe Sendeperioden erkennen zu können. Diese können mithilfe eines Tiefpassfilters erfasst werden. Die maximale Sendeperiode T_{max} lässt sich mithilfe der folgenden Formel berechnen:

$$T_{max} = T_{max} + \frac{T_{mom} - T_{max}}{\beta}, \qquad \beta = 64 \tag{6.35}$$

$$wenn \quad T_{max} \cdot \frac{100+var}{100} > T_{mom} > T_{max} \cdot \alpha, \quad \alpha = 0{,}9\,. \tag{6.36}$$

Hierbei sind T_{mom} die momentan gemessene Sendeperiode, *var* ein festgelegter Wert für die mögliche Varianz der Sendeperiode, α das Minimum der Sendeperiode T_{max} sowie β der Glättungskoeffizient des Tiefpasses. Für jedes Datensegment wird die Sendeperiode gemessen und T_{max} entsprechend angepasst. Dies geschieht, wenn die momentane Sendeperiode T_{mom} in einem Bereich um T_{max} liegt, d.h. höher als *90 %* von T_{max} und kleiner als die prozentuale Erhöhung von T_{max} um *var* Prozent. Für die implementierte Version der Funktion wird für *var* ein Wert von *30 %* eingestellt.

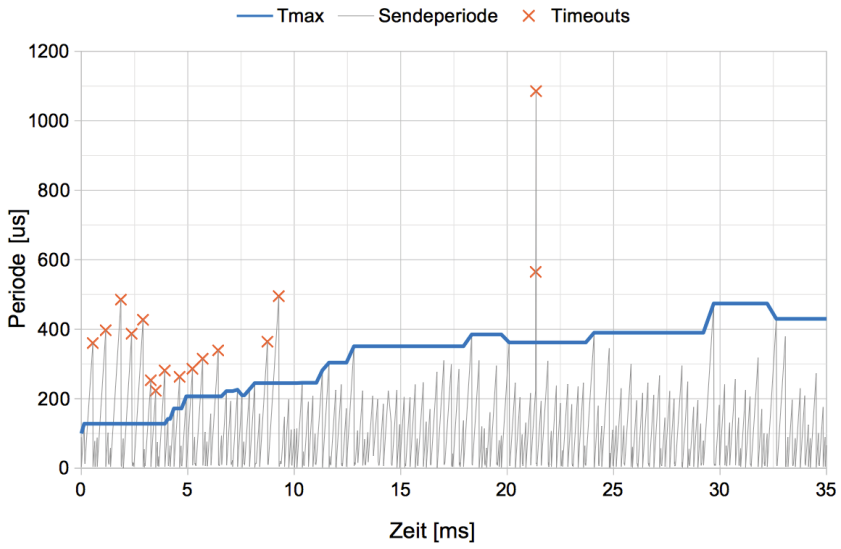

Abbildung 6.29: Störungserkennung mithilfe der Sendeperiode

Abbildung 6.29 zeigt die Messung der Sendeperioden einer Übertragung (grau) und die Berechnung der maximalen Sendeperiode T_{max} (blau) unter der Nutzung von Formel 6.35. Wie aus der Grafik ersichtlich wird, passt sich T_{max} der Sendeperiode an, solange sich die Abweichung im gesetzten Fenster befinden. Mithilfe der Formel werden kleine Erhöhungen der Sendeperiode ermöglicht, ohne dass ein Timeout erzeugt wird. Stärkere Schwankungen jedoch werden erkannt und führen zu einem Timeout. Ein Timeout wird durch das Überschreiten des oben angegebenen Wertes in der bedingenden Formel 6.36 ausgelöst. Dies ist zu Beginn der Übertragung der Fall, da der Wert noch nicht die nötige Anpassung erreicht hat. Timeoutauslöser müssen also in den ersten Millisekunden einer Übertragung ignoriert werden.

Werden mehrere Timeouts ausgelöst, d.h. werden Ausreißer der Sendeperiode erkannt, so gelangt das in Abbildung 6.24 angegebene Zustandsdiagramm in den Zustand *Timeout*. Weitere Timeouts sind nötig, um tatsächlich einen Störungsfall anzuzeigen und dann in den Zustand *Congestion* zu wechseln. Der Subflow wird für diesen Fall als überlastet markiert. Erholt sich die Verbindung, wird in den Zustand *Recovery* gewechselt.

6.5.3 Störungserkennung durch Beobachtung der effektiven Datenübertragungsrate und TCP-Überlastzustand

Eine Störungserkennung kann ebenfalls bei einem plötzlichen Abfall in der effektiven Datenübertragungsrate erfolgen. Wenn ein Subflow nicht mehr die vom Scheduler vorgesehene Datenmenge überträgt, verringert sich insgesamt die effektive Datenmenge, die eine Applikation ins Netzwerk einspeisen kann. Dieser Umstand kann dazu verwendet werden, Störungen zu erkennen.

Die Messung der Datenübertragungsrate muss über einen bestimmten Zeitraum geschehen und kann nicht segmentweise vorgenommen werden. Der untersuchte Zeitraum für die Übertragung eines einzelnen Segments wäre zu gering und eine Messung würde zu unrealistischen Werten führen. Bei einer durchgängigen Datenübertragung wäre eine gemessene Zeitspanne von mindestens *20 ms* erforderlich. Die Festlegung einer bestimmten Zeitspanne erfolgt in Abhängigkeit der verwendeten Anwendungen.

Um einen Störungsfall zu detektieren, wird die durchschnittliche effektive Datenübertragungsrate $E_{eff,mean}$[43] verwendet. Ein Störungsfall wird dann erkannt, wenn die folgende Gleichung zutrifft:

$$E_{eff} < BW_{threshold} \cdot E_{eff,mean} \tag{6.37}$$

Hierbei ist E_{eff} die momentan gemessene effektive Datenübertragungsrate und $BW_{threshold}$ die prozentuale Angabe, ab wann ein zusammengebrochener Subflow erkannt werden soll. Es wird ein typischer Wert von *80* bis *90 %* angesetzt. Wird ein Störungsfall erkannt, wird E_{mean} als $E_{meanBeforeCong}$ abgespeichert und kann für den in Kapitel 6.4.3 beschriebenen adaptiven Algorithmus verwendet werden.

[43] zu berechnen durch die effektive Datenübertragungsrate und einer Abwandlung von Formel 6.17

Der Algorithmus erkennt zwar, dass ein Störungsfall vorliegt, ist aber nicht in der Lage, den störungsbehafteten Subflow zu identifizieren, da lediglich die effektive Datenübertragungsrate der Applikation gemessen wird. Zu diesem Zweck und um die Richtigkeit eines Störungsfalls zu überprüfen, wird der Überlastungszustand des darunterliegenden TCP-Protokolls von jedem Subflow ausgelesen. Der Subflow mit dem höchsten Überlastungsgrad wird dann für die weitere Verwendung in den adaptiven Algorithmen markiert.

Der folgende Pseudocode 6.3 zeigt die Funktionsweise für die Markierung in der Störungserkennung durch Abfrage des TCP-Zustands, der im Störungsfall gleich oder höher als der *Recovery*-Zustand liegt. Der Subflow muss sich im Störungszustand befinden und *nutzbar* sein.

```
sk_event = 0;
FOR (i=0; i < Anzahl_Subflows; i++) {
  IF (TCP_state(i) >= TCP_CA_Recovery) {
    IF (TCP_state(i) > sk_event) AND (Subflow_state(i)=enabled)
    {
        sk_event = TCP_state(i);
    }
  }
}
```

Pseudoocode 6.3: Pseudocode für die Störungserkennung eines Subflows

7 Implementierung von rMPTCP und Anwendungen zur Nutzung und Auswertung

In diesem Kapitel wird ein Einblick in den strukturellen Aufbau der Implementierung gegeben und es werden Anwendungen beschrieben, die für eine Funktion und die weitere Untersuchungen benötigt werden. Hierzu gehören die grundlegende Implementierung von rMPTCP sowie eine Verwendung des Pfad-Managements durch eine externe Anwendung.

Der Einsatz eines entwickelten Gateways für die Verwendung von rMPTCP innerhalb des telemedizinischen Basisszenarios wird erläutert. Die Anwendungsschnittstelle von rMPTCP wird für eine volle Funktion und Evaluation benötigt und in diesem Kapitel offengelegt. Zum Schluss werden die entwickelte Steuerungsanwendung und deren Funktionsumfang für rMPTCP beschrieben.

Bei der Durchführung der Implementierungen haben die Abschlussarbeiten [Siahaan, 2015], [Zhang, 2016] und [Verbunt, 2017] unterstützend mitgewirkt.

7.1 Aufbau und Funktionen der rMPTCP Implementierung

Die Implementierung von rMPTCP erfolgt auf Basis der Implementierung von MPTCP, beschrieben in [Paasch & Barre, 2014], [Paasch & al., 2016], [Paasch, 2012] und [Paasch, 2014]. Abbildung 7.1 zeigt eine Übersicht über die Ordnerstruktur der Implementierung von rMPTCP und die verwendeten Programm- sowie Bibliotheksdateien.

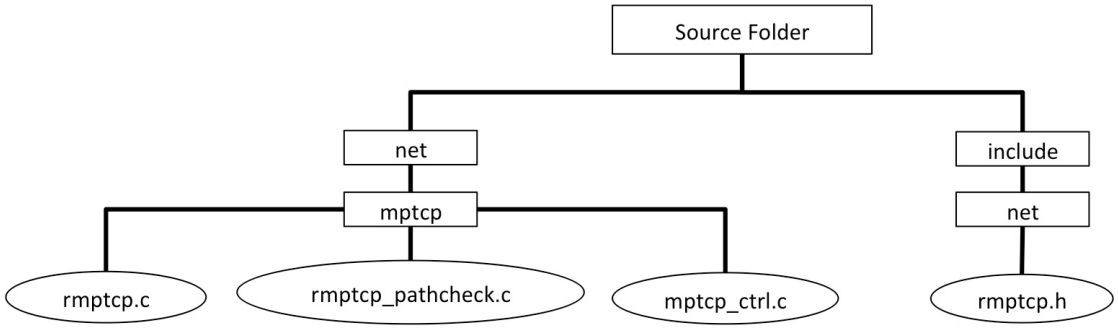

Abbildung 7.1: Hierarchischer Überblick der Ordner (Rechtecke) und Programmdateien (Ellipsen) der rMPTCP-Implementierung

Innerhalb der aufgelisteten Dateien werden die Funktionen des rMPTCP Schedulers umgesetzt. Die folgende Tabelle gibt Aufschluss über die Aufteilung der Funktionen in den Dateien.

Dateiname	Beschreibung
rmptcp.c	Hauptprogrammdatei für die Algorithmen des rMPTCP-Schedulers
rmptcp_pathcheck.c	Abrufen der benötigten Pfadkombinationen und Leistungsmessungen jeder Kombination
mptcp_ctrl.c	Original MPTCP-Programmdatei; Definition der SYSCTL-Variablen und Monitor Outputs für rMPTCP
rmptcp.h	Header-Bibliotheksdatei; Definition aller für die Funktionalität benötigter Strukturen, Variablen sowie der Subflow- und Störungszustände

Tabelle 7.1: Dateien der Implementierung

Die wichtigste Programmdatei ist hier *rmptcp.c*, welche die entwickelten rMPTCP-Algorithmen umsetzt. Die Programmdatei *mptcp_crtl.c* wird hauptsächlich für die Definition der SYSCTL-Variablen der API verwendet, die für eine rMPTCP-aware Anwendungen verwendet werden können und vom rMPTCP-Steuerungsprogramm genutzt wird. Bedeutend ist auch die Programmdatei *rmptcp_pathcheck.c* für die Pfadauswahl und Beurteilung der Wahl. Im Folgenden werden die beinhaltenden Funktionen der beiden Programmdatei *rmptcp.c* und *rmptcp_pathcheck.c* kurz erläutert.

rmptcp.c

Funktion	Beschreibung
is_available()	Überprüfen hinsichtlich der Verfügbarkeit eines Subflows zum Senden
insert_sk()	Geordnetes Einfügen eines Subflows in die Liste der vorhandenen Subflows
set_enable_flag()	Setzen einer *Verfügbar*-Markierung für die gegebene Pfadmaske
update_sk_list()	Suche und Abspeicherung von neuen Subflows
set_all_main()	Setzen aller *verfügbar* und *Ersatz*-Subflows als *Main*
set_all_uncong_main()	Setzen aller *störungsfreien*, *verfügbaren* und *Ersatz*-Subflows als *Main*
update_main_sk()	Setzen einer *Main*-Flag für die Out-of-Order Priorisierung
update_bw()	Messen der Auslastung aller Subflows (U_i, $U_{i,mean}$), der Gesamtauslastung (U_{tot}, $U_{tot,mean}$), der effektiven Datenübertragungsrate (E_{eff}, $E_{eff,mean}$)
regulateQ()	Regulieren der Redundanzquote während eines Störungszustands mit adaptiver Redundanz
statechange_cong()	Wechsel in den Störungszustand
statechange_timeout2cong()	Wechsel vom Timeout-Zustand in den Störungszustand
statechange_event2cong()	Wechsel in den Störungszustand, nachdem ein Subflowfehler detektiert wurde.
check_failure()	Detektion von Störungen und Setzen des Zustands

Funktion	Beschreibung
setQ()	Berechnen der Kontrollwerte $Q_{control}$ für die Redundanzquote
adaptR()	Anwenden des adaptiven Redundanzalgorithmus
adaptPath()	Anwenden des adaptiven Pfadalgorithmus
setQActual()	Messen der momentanen Redundanzquote Q_{actual}
check_avail_sk()	Überprüfen der verfügbaren Subflows auf die zu erfüllende Redundanzquote. Kernfunktion des Schedulers, um zu entscheiden, welche Subflows die anstehenden Segmente verschicken sollen, siehe Abbildung 6.7
calc_controlQ()	Anwenden der Kontrollwerte $Q_{control}$ in Hinblick auf Q_{actual}
calc_sendPeriod()	Messen der Sendefrequenz, um Timeouts und Störungen zu erkennen
send_skb()	Senden eines Segments
update_sysctl()	Anwenden der SYSCTL-Variablen und Umsetzung in die Implementierung
reset_meta()	Zurücksetzen aller Parameter im Falle einer Änderung im Meta-Socket
send_next_segment()	Einstiegspunkt und Hauptfunktion des rMPTCP-Schedulers, siehe Abbildung 7.2

Tabelle 7.2: Implementierte Hauptfunktionen

rmptcp_pathcheck.c

Funktion	Beschreibung
getAddresses()	Einmaliges Abrufen aller verfügbaren Adressen für den Pfad-Manager
getPathCombinations()	Bestimmen aller erlaubten Pfadkombinationen ohne doppelte Verwendung der Netzwerkschnittstellen
getMeasurements()	Abrufen und Abspeichern der Messinformationen der Pfadmessungen, die zur momentanen Pfadkombination gehören
pathChecker()	Einstiegspunkt der Pfadevaluation. Abrufen aller erlaubten Pfadkombinationen und Leistungsmessungen jeder Kombination

Tabelle 7.3: Implementierte Funktionen für die Pfadauswertung

Der Ablauf des Haupt-Algorithmus folgt dem Schema in Abbildung 6.6. Er wird in der Hauptfunktion *send_next_segment()* implementiert. Die nachfolgende Abbildung 7.2 zeigt das Aktivitätsdiagramm mit den eingefügten Funktionen. Dieser Ablauf wird für jedes zu sendende Datensegment durchgeführt. Hilfsfunktionen, die durch andere Funktionen aufgerufen werden, werden in diesem Aktivitätsdiagramm nicht berücksichtigt.

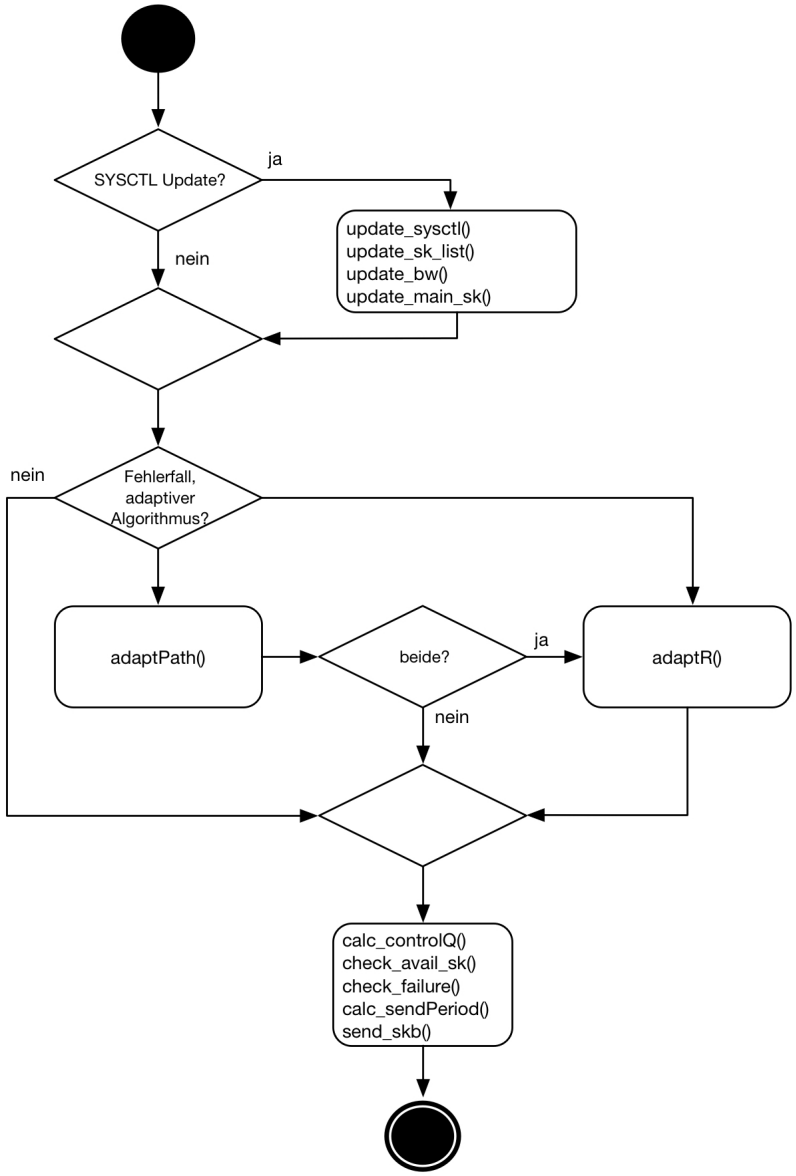

Abbildung 7.2: Aktivitätsdiagramm des rMPTCP-Schedulers der Funktion send_next_segment() in rmptcp.c

7.2 Manuelles Pfadmanagement

Die Entwicklung eines manuellen Pfadmanagements wird durch eine API realisiert, die es ermöglicht, Subflows auszuwählen und deren Eigenschaften durch eine Applikation zu beobachten. Der manuelle Pfadmanager funktioniert mithilfe einer Maske, die in Form eines Hexadezimalcodes an die API weitergeleitet wird. Hiermit können aus den realisierbaren Subflows diejenigen ausgewählt werden, die geschaltet werden sollen. Realisierbare Subflows sind zwischen allen vorhandenen Netzwerkinterfaces möglich. Netzwerkinterfaces auf der lokalen Seite werden mit den Netzwerkinterfaces auf der entfernten Seite in Kombination dargestellt. Die Maske gibt an, welche Subflows aus der Kombination von allen Netzwerk-Interfaces aktiviert und welche deaktiviert werden sollen. Die Maske errechnet sich wie folgt:

Jeder mögliche Subflow wird mit einer „1“ aktiviert und mit einer „0“ deaktiviert. Die Maske stellt bis zu sechzehn mögliche Subflows zur Verfügung, was einer Anzahl von vier Netzwerkinterfaces entspricht. Subflows, die gar nicht vorhanden sind, müssen ebenfalls auf „1“ geschaltet werden. Werden diese Bits in hexadezimal umgerechnet, so ergibt sich die Pfadmaske. Ein Beispiel zeigt die nachfolgende Tabelle 7.4 für die Aktivierung der Subflows 0, 2 und 6 – Subflows ab 9 und höher sind nicht vorhanden.

Subflow #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit	1	1	1	1	1	1	1	0	0	1	0	0	0	1	0	1
Hex	F				E				4				5			

Tabelle 7.4: Pfadmaske der Subflows in rMPTCP

Das Beispiel zeigt eine Eins-zu-Eins-Verbindung von drei Netzwerkinterfaces. Eine Einstellung der Subflows erfolgt über die ersten 9 Bits. Im Beispiel werden genau drei Subflows erstellt. Die sich daraus ergebende Maske ist „FE45“. Zum Zwecke einer besseren Handbarkeit der Funktionen wurde eine Kontroll-Applikation realisiert, die Zugriff auf die von rMPTCP bereitgestellte API besitzt. Abbildung 7.3 zeigt das oben dargestellte Beispiel unter der Benutzung der API mithilfe der Kontroll-Applikation.

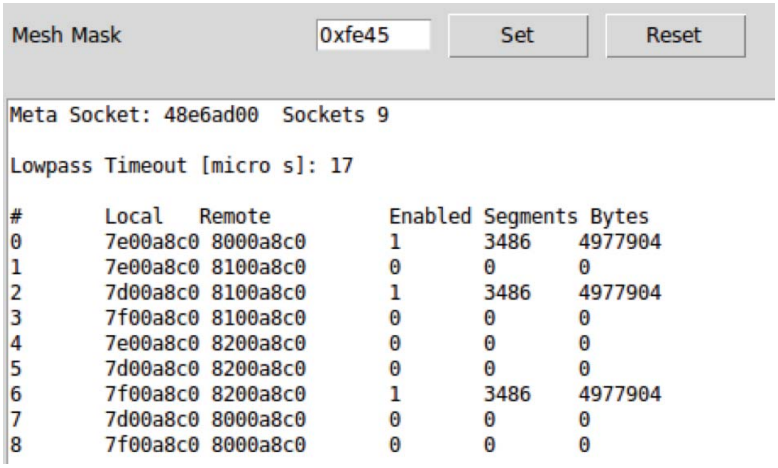

Abbildung 7.3: Benutzung der rMPTCP-Subflow-API

Im großen Textfeld sind zunächst alle denkbaren Subflows dargestellt. In der Spalte „Local“ sowie „Remote“ befindet sich die IP-Adresse des Netzwerkinterfaces in Form eines Hexadezimalcodes. Unter Eingabe der Pfadmaske unter „Mesh Mask“ können die spezifischen Subflows aktiviert und deaktiviert werden. Die in Kapitel 6.2.1 beschriebene Redundanzquote wirkt sich nur auf die aktivierten Subflows aus.

7.3 Automatische Pfadauswertung

Eine genaue Auswertung der Subflow-Eigenschaften zur optimalen Auswahl des Pfad-Managements ist nur durch Messung möglich. Für eine erschöpfende Messung der Datenübertragungsrate sollten möglichst viele Daten über den Subflow zum Empfänger gesendet werden. Bei einem nur zum Teil verwendeten Subflow, der z.B. Träger von Echtzeitdaten ist, kann keine Messung der vollen Datenübertragungsrate durchgeführt werden. Die Messung der RTT kann erfolgen, wenn Daten über den Subflow verschickt

werden. Während des regulären Betriebs ist eine passive Messung möglich. Diese kann neben der RTT lediglich die Auslastung eines Subflows feststellen, die von der Applikation selbst abhängig ist.

Um die oben diskutierten Funktionen zu implementieren, ist es von Vorteil, vor Beginn des regulären Betriebs eine *Pfad-Metrik* der verschiedenen Parameter einer Subflow-Kombination zusammenzustellen. Auf dieser Grundlage können Benutzer und das Pfad-Management ihre Entscheidungen treffen.

Eine Pfad-Metrik, die die Kriterien für eine Kombination aus möglichen Subflows abbildet, besteht aus den folgenden Parametern, die mit der oben genannten Methode zu bestimmen sind:

- *Minimale* $RTT_{i,min}$ ist die RTT des Subflows mit der geringsten RTT der Pfadkombination;
- *Gesamt* RTT_{tot} als die Summe aller Subflow RTTs, die als qualitativer Wert genutzt werden, um eine Pfadkombination insgesamt beurteilen zu können;
- *Gesamtauslastung der Subflows* U_{tot} gibt die Summe der momentanen Datenübertragungsrate aller Subflows an;
- *Minimale Auslastung eines Subflows* $U_{i,min}$ ist der Wert der Datenübertragungsrate des Subflows mit der geringsten ausgenutzten Datenübertragungsrate.

Die Überwachung und die Steuerung der Pfad-Metriken kann über ein Benutzerinterface bedient werden. Hierbei kann der Benutzer das Fenster für die durchschnittliche Messung eines Tests einstellen. Ein Beispiel für die Pfad-Metriken eines durchlaufenen Tests zeigt die folgende Tabelle:

Pfad	**Pfadmaskierung**	**$RTT_{i,min}$** [us]	**RTT_{tot}** [us]	**U_{tot}** [kbit/s]	**$U_{i,min}$** [kbit/s]
0	0x11	345223	435360	12341	8043
...	...	...	...	...	...

Tabelle 7.5: Beispiel einer Pfad-Metrik der automatischen Pfadauswertung

Um einen Ausgleich der Datenübertragungsrate zu schaffen, müssen vor allem die Werte U_{tot} und $U_{i,min}$ ermittelt werden:

- U_{tot}, um eine dynamische Redundanzanpassung zu erlauben
- $U_{i,min}$, um eine bestmögliche Kombination der Subflows zu gewährleisten, da der schwächste Subflow große Auswirkungen auf die gemeinsame Datenübertragungsrate hat

Daher wird ein Mechanismus entwickelt, der aus den zur Verfügung stehenden Adressen eine geeignete Kombination auswählt, diese evaluiert und für die Applikation abbildet. Die folgende Abbildung 7.4 zeigt ein Aktivitätsdiagramm mit den einzelnen Schritten für den Mechanismus der automatischen Pfadevaluation.

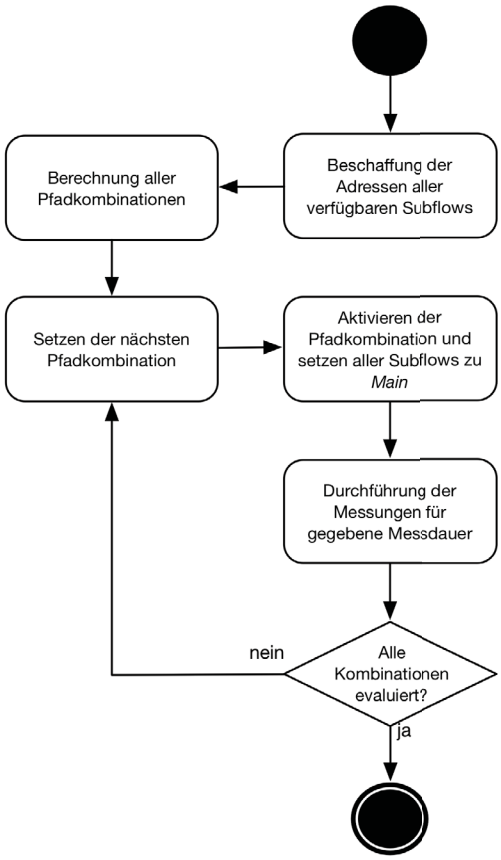

Abbildung 7.4: Aktivitätsdiagramm der automatischen Pfadauswertung

Zunächst werden alle verfügbaren Netzwerkendpunktadressen abgerufen und mögliche Pfadkombinationen bestimmt. Da die Subflows keine sich überschneidenden Pfade besitzen sollen, ist die Anzahl der nutzbaren Subflows beschränkt. Sie ist gleich der minimalen Anzahl *N* der Schnittstellen der miteinander verbundenen Endpunkte, sodass jedes Netzwerkinterface nur einmal benutzt wird:

$$N_{paths} = \min\left(N_{addr,src}, N_{addr,dst}\right) \tag{7.1}$$

Mit der errechneten Anzahl an Subflows werden alle Kombinationen der Pfadmaske auf ein Maximum an möglichen unabhängigen Subflows getestet. Ist die Anzahl der in der Pfadmaske enthaltenen Subflows gleich N_{paths} und wird in jedem Subflow ein Netzwerkinterface nur einmal verwendet, so wird die Pfadmaske abgespeichert.

```
FOR (i=0; i < Anzahl_Pfadmasken; i++) {
  IF Anzahl_Subflows(Pfadmaske(i)) = N_paths {
    IF Pfadmaske(i) contains no duplicate Interfaces
      SAVE Pfadmaske
} }
```

Pseudocode 7.1: Überprüfung der Pfadmaske auf die richtige Subflow-Kombination

Jede mögliche Kombination der gefundenen Pfadmasken wird nacheinander evaluiert. Eine Messung wird für eine festgelegte Dauer durchgeführt. Nur die durch die Pfadmaske spezifizierten Subflows werden für die Messung aktiviert und als Main-Subflow gesetzt, damit alle Subflows gleichberechtigt Daten übertragen können. Nach-

dem alle möglichen Pfadmasken evaluiert wurden, wird die frühere Pfadauswahl wiederhergestellt. Der Benutzer kann über die API eine Pfadkombination auswählen und aktivieren.

7.4 Gateway für Geräte ohne rMPTCP-Funktionen

Das in dieser Arbeit entwickelte Protokoll ist für den Einsatz in einem telemedizinischen Szenario als Backbone-Strategie zwischen zwei Standorten ausgelegt. Das telemedizinische Szenario enthält mehrere Geräte wie den ARTP, die miteinander vernetzt sind. Um das entwickelte Protokoll zum Einsatz zu bringen, wird ein Gateway entworfen, das die Datenübertragung von einem Standort zum anderen koordiniert und die Daten mehrerer Geräte über rMPTCP überträgt.

Durch eine einfache Netzwerkverbindung zum Gateway können beliebig viele Kleingeräte verwendet werden, um rMPTCP zu nutzen. Das Gateway bedeutet einen lokalen Einstiegspunkt für verschiedene Geräte und überbrückt die kritische Strecke. Zu diesem Zweck existiert an jedem Endpunkt jeweils ein Gateway.

Durch das Gateway wird mithilfe von rMPTCP ein Verfahren geboten, mit dem für netzwerkfähige Geräte jeglicher Art kostengünstig eine Latenzstabilisierung bei der Überbrückung kritischer Verbindungsstrecken ermöglicht wird. Das Gateway kann jede Anfrage eines Geräts empfangen und diese an ein weiteres Gateway weiterleiten. Es dient als Relais, um den normalen TCP-Verkehr als rMPTCP weiterzuleiten. Hiermit ist es möglich, die Einstellungen des rMPTCP-Schedulers an einem einzelnen Endpunkt durchzuführen, obwohl mehr Geräte involviert sind.

Eine einfache Aufstellung des Systems stellt Abbildung 7.5 dar. Zwei TCP-Endsysteme sind mit dem jeweiligen lokalen Gateway verbunden. Beide Gateways wiederum bilden eine rMPTCP-Verbindung über das Internet.

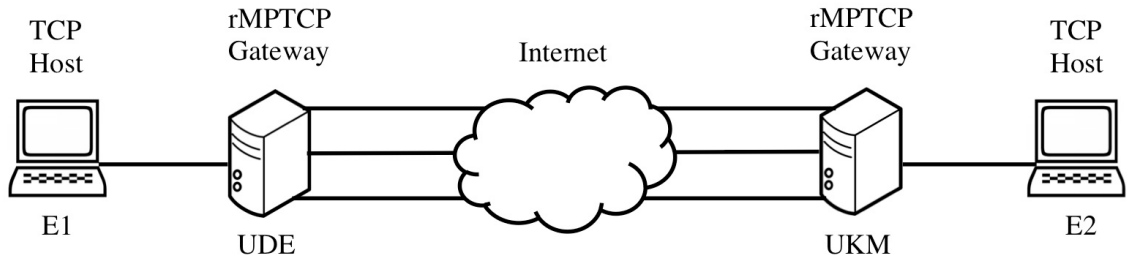

Abbildung 7.5: rMPTCP-Gateway-Aufstellung

Die Gateways UDE und UKM kommunizieren jeweils mit den Geräten E1 und E2 unter Verwendung des TCP-Protokolls. E1 und E2 verfügen über keinerlei Multihoming-Fähigkeiten und besitzen jeweils nur ein Netzwerkinterface.

Für eine Realisierung des Gateways auf der Anwendungsschicht wird ein Client/Server-Modell verwendet. Im Falle eines Datentransfers von Endgerät E1 nach Endgerät E2 wird hierfür das UDE-Gateway zunächst als Server von Endgerät E1 behandelt. E1 nimmt Kontakt zum UDE-Gateway auf, ohne Notiz davon zu nehmen, dass das UDE-Gateway nicht das tatsächliche Endgerät ist. Das Gateway bietet eine grafi-

sche Benutzerschnittstelle, mit der angeschlossene Geräte und Kernel-Priorisierungen für die Verarbeitung verwaltet werden können.

7.5 Application Programmable Interface von rMPTCP

Um Einstellungen und Prozesse von rMPTCP steuern und beobachten zu können, wird eine API benötigt, die von einer rMPTCP-fähigen Applikation, aber auch unabhängig von der genutzten Anwendung verwendet werden kann. Sie soll die Kontrolle über mehrere Funktionen und Parameter von rMPTCP ermöglichen.

In Abhängigkeit von den Anforderungen der Applikation können die oben beschriebenen Algorithmen mithilfe der API eingesetzt und von der Anwendung selbst gesteuert werden. rMPTCP-aware-Applikationen, also Anwendungen, die eigens für rMPTCP entwickelt oder dafür modifiziert wurden, können die API nutzen. Dies ermöglicht auch die Entwicklung einer Steuerungssoftware, die bei herkömmlichen TCP-Anwendungen für die Feinjustierung von rMPTCP eingesetzt werden kann. Eine Anwendung wäre dann ohne Modifikation rMPTCP-fähig.

Für weitere Untersuchungen von rMPTCP bedarf es zusätzlicher Monitor- und Einstellungsmöglichkeiten, die es erlauben, rMPTCP und die Auswirkungen auf den Datenstrom weiter zu untersuchen. Die folgenden Eigenschaften werden durch die API von rMPTCP realisiert:

Redundanzmanagement:

- **Redundanzgrad (Redundanzquote)**: Definiert, wie hoch die leistungsbezogene Redundanz hinsichtlich der benutzten Subflows ist. Der Standardwert liegt hier bei $Q_{target}=100$, d.h. alle Daten werden ohne leistungsbezogene Redundanz versendet.
  (Kernelparameter sysctl_rmptcp_quota)
- **Redundanz-Timeout:** Definiert den maximalen Zeitraum, für den rMPTCP ein Segment zurückhält, bis es über weniger Subflows gesendet wird, als es die Redundanzquote spezifiziert. Der Standardwert beträgt *50 ms.*
  (Kernelparameter sysctl_rmptcp_timeout)

Pfad-Management

- **Socket-Timeout:** Definiert die Zeit, die ein Subflow innerhalb des Registers bleibt, ohne *verfügbar* zu sein. Wird diese Zeit überschritten, so wird ein Socket-Timeout ausgelöst, womit ausgedrückt wird, dass der Subflow ausgefallen ist. Danach wird der Subflow entfernt. Die Standardzeit, bis ein Socket-Timeout ausgelöst wird, liegt bei *20.000 ms*.
  (Kernelparameter sysctl_rmptcp_maxnoshow)
- **Update-Periode**: Definiert den Zeitabschnitt, wie oft die Zusammenstellung der Subflow-Liste upgedatet wird. Der Standardwert beträgt hier *50 ms*.
  (Kernelparameter sysctl_rmptcp_skreg_update)

- **Pfad-Maske:** Definiert die Einstellungen der verwendeten Subflows, die Anzahl und die Vermaschung. Standardparameter ist *0xFFFF*, die Einstellung für eine Komplettaktivierung aller potenziellen Subflowkombinationen.
  (Kernelparameter sysctl_rmptcp_enablemask)
- **Entscheidungsfenster δ für zu identifizierende Main-Subflows:** Für die Out-of-Order-Vermeidung wird der Parameter δ für die Berechnung der benötigten Fähigkeiten eines Main-Subflows genutzt. Der Standardwert ist *2*. Damit liegt ein Main-Subflow unterhalb der doppelten niedrigsten gemessenen RTT und ist größer als die halbe höchstgemessene Datenübertragungsrate.
  (Kernelparameter sysctl_rmptcp_main_ratio)
- **Aktivierung der Pfadevaluation:** Durchführung einer Messung zur Pfadevaluation für eine Auswahl der in der Übertragung zur Verfügung stehenden Subflows. Der Standardwert liegt bei *0*, eine Deaktivierung der Messungen.
  (Kernelparameter sysctl_rmptcp_runpathcheck_duration)
- **Messdauer für die Pfadevaluation:** Definiert die Messdauer der Pfadevaluation für eine Auswahl der in der Übertragung zur Verfügung stehenden Subflows. Der Standardwert liegt bei *5 s*.
  (Kernelparameter sysctl_rmptcp_runpathcheck)
- **Aktivierung einer Pfadkombination:** Parameter zur Aktivierung bestimmter Subflow-Kombinationen. Standardmäßig sind alle Kombinationen deaktiviert und liegen damit bei *0*.
  (Kernelparameter sysctl_rmptcp_setPath)
- **Subflow-Entfernung:** Wenn mehr potenzielle Subflows erkannt wurden, als benötigt werden, werden diese deaktiviert. Diese Funktion kann mithilfe dieses Parameters aktiviert oder deaktiviert werden. Der Standard geht von einer Nutzung der Funktion aus und ist daher *1 (wahr)*.
  (Kernelparameter sysctl_rmptcp_enable_kick)

Störungserkennung

- **Erholung von Störungszuständen:** Definiert die Zeit, die verstreichen muss, bis vom *Congested*-Zustand oder vom *Recovery*-Zustand in den nächstniedrigeren Erholungszustand gewechselt wird. Die Standardeinstellung liegt hier bei *500 ms*.
  (Kernelparameter sysctl_rmptcp_CS_hold)
- **Varianz der Sendeperiode:** Der Parameter *var* wird genutzt um eine mögliche Varianz der Sendeperiode bei der Störungserkennung zu definieren. Der Standardwert beträgt *30 %*.
  (Kernelparameter sysctl_rmptcp_varperiod_th)
- **Timeout-Anzahl der Sendeperiode:** Die Spezifizierung der Anzahl aufeinanderfolgender Sendeperiode-Timeouts, bis das Wechseln in den nächsthöheren Störungszustand eingeleitet wird. Der Standardwert beträgt *20 Time-*

outs.
(Kernelparameter sysctl_rmptcp_numinTime)

- **Schwellwert der effektiven Datenübertragungsrate:** Der Parameter $BW_{threshold}$ steht für die Erkennung eines Zusammenbruchs in der effektiven Datenübertragungsrate. Der Standardwert liegt hier bei *90 %* der durchschnittlichen Datenübertragungsrate.
(Kernelparameter sysctl_rmptcp_collapse_th)
- **Messdauer für Datenübertragungsrate:** Zeitdauer für die Bestimmung der momentanen Datenübertragungsrate beim Messen der abgesendeten Datensegmente. Der Standardwert ist *20 ms*.
(Kernelparameter sysctl_rmptcp_minDeltaT)

Adaptive Algorithmen

- **Adaptive Redundanz:** Parameter zur Aktivierung/Deaktivierung des Algorithmus zur adaptiven Redundanz. Standardmäßig ist dieser Algorithmus aktiviert.
(Kernelparameter sysctl_rmptcp_adaptivR_enable)
- **Adaptives Pfad-Management:** Parameter zur Aktivierung/Deaktivierung des Algorithmus des adaptiven Pfad-Managements. Der Algorithmus ist standardmäßig aktiviert.
(Kernelparameter sysctl_rmptcp_adaptivPath_enable)

Zu diesen steuerbaren Parametern, die zur weiteren Untersuchung und Feinjustierung auf die anwendungsspezifischen Erfordernisse verwendet werden, kommt die Möglichkeit hinzu, verschiedene Parameter auch visuell zu beobachten. Bei Nutzung einer kritischen Anwendung ermöglicht dies, die Situation im Netzwerk zu beobachten und darauf zu reagieren. Zudem bietet diese Funktion die Möglichkeit Monitoringdaten abzuspeichern.

- **Datenübertragungsrate:** Monitoring der unterschiedlichen Datenübertragungsraten und effektiven Nutzwerte. Dazu gehören: Gesamte durchschnittliche Datenübertragungsrate, effektive Datenübertragungsrate, durchschnittliche effektive Datenübertragungsrate und die letzte effektive Datenübertragungsrate vor einer Überlastsituation.
- **Redundanz-Monitoring:** Das Monitoring ermöglicht das Auslesen aller Redundanzwerte, d.h. die gewünschte leistungsbezogene Redundanz, die adaptierte leistungsbezogene Redundanz und die über Subflows gesendete Segmentanzahl. Zusätzlich wird der derzeitige Zustand für die Überlastkontrolle der Störungserkennung ausgegeben.
- **Subflow-Monitoring:** Monitoring verschiedener Parameter der einzelnen Subflows. Dazu gehören: lokale IP Adresse des Subflows, entfernte IP Adresse des Subflows, Aktiv-Anzeige, Anzahl gesendeter Segmente, RTT, gesendete Bytes, Sende-/Empfangsfenstergröße, durchschnittlicher zeitlicher Abstand zwischen Segmenten.

- **Pfad-Monitoring:** Monitoring verschiedener Parameter hinsichtlich der einzelnen Pfadkombinationen und ihrer Eigenschaften. Dazu gehören: Anzahl der bekannten Pfadkombinationen, Verfügbarkeit, Datenübertragungsrate und RTT.

7.6 Steuerungsapplikation für rMPTCP

Es wurde eine Steuerungsapplikation entworfen, mit der verschiedene Parameter in rMPTCP gesteuert und kontrolliert werden können. Für telemedizinische Applikationen bietet sie die Möglichkeit, den Datenfluss auf die Bedürfnisse der jeweiligen Applikation sowie des Szenarios unabhängig von allen rMPTCP-unaware-Applikationen einzustellen. Die im vorherigen Abschnitt implementierten Optionen und Parameter, die sich über die API kontrollieren und beobachten lassen, werden innerhalb der Steuerungsapplikation berücksichtigt. Damit sind weitere Möglichkeiten gegeben, rMPTCP besser zu erforschen und zu erweitern.

Das Programm wurde mit Python 3 [Python, 2017] erstellt und bietet eine grafische Benutzeroberfläche für die Verwendung unter Linux, auf der sich die einzelnen API-Parameter einstellen und auslesen lassen. In der rechten Hälfte sind die Parameter einzustellen. Die linke Hälfte zeigt die Übersicht der Monitoringdaten der Verbindung, der Subflows und der Pfadkombinationen.

Eine Abbildung der Steuerungsapplikation befindet sich im Anhang.

8 Simulation und Auswertung des entwickelten Systems

In diesem Kapitel wird eine Auswertung der im Rahmen dieser Arbeit entwickelten Verfahren und Bestandteile des rMPTCP-Protokolls beschrieben. Das Protokoll wurde auf mehreren Rechnern des Instituts installiert und es wurde eine geeignete Simulationsumgebung im Labor geschaffen, mit der verschiedene Situationen des Netzwerks nachgebildet werden konnten. Das Verhalten der verschiedenen Protokoll-Funktionen wurde im Rahmen mehrerer Messreihen erprobt und dessen Nutzen und Grenzen eingefasst.

Zu Anfang wird auf die bei den Messungen verwendeten Methoden und Werkzeuge eingegangen und die Simulationsumgebung veranschaulicht. Die verschiedenen Funktionen und Bestandteile des Protokolls werden daraufhin mithilfe verschiedener Szenarien überprüft und bewertet.

Einige Inhalte der hier dargestellten Ergebnissen wurden gleichfalls in den Veröffentlichungen [Hunger & Klein, 2016] und [Hunger et al., 2016] dargelegt. Bei der Durchführung der Auswertung haben unterstützend die Abschlussarbeiten [Zhang, 2016] und [Verbunt, 2017] mitgewirkt.

8.1 Simulationsumgebung, Methoden, Werkzeuge

Für die Simulation kommen zwei Rechner zum Einsatz, die mit einer rMPTCP Kernel-Implementierung ausgestattet sind. Die beiden verwendeten Rechner besitzen die folgenden technischen Merkmale:

PC 1	Marke und Typ	Dell OptiPlex 990
	Prozessor	Intel Core i3-2100 CPU @ 3,1 GHz x4
	Arbeitsspeicher	4 GB
	Betriebssystem	Ubuntu 14.04.2 LTS, Linux 4.4.0-75-generic x86_64
	Netzwerkkarte	Intel 82579LM Ethernet, 10/100/1000 Mbit
	Netzwerkkarte	3x Realtek RTL8152 Based USB 2.0, 100 Mbit
PC 2	Marke und Typ	Dell OptiPlex 755
	Prozessor	Intel Core 2 Duo CPU @ 2,66 GHz x2
	Arbeitsspeicher	5GB
	Betriebssystem	Ubuntu 14.04.2 LTS, Linux 4.4.0-75-generic x86_64
	Netzwerkkarte	Intel 82566DM Gigabit LAN 10/100/1000 Mbit
	Netzwerkkarte	3x Realtek RTL8152 Based USB 2.0, 100 Mbit

Tabelle 8.1: Technische Daten der Rechner für die rMPTCP-Auswertung

Jeder Rechner besitzt drei identische Netzwerkadapter für eine einheitliche Ansprechzeit, die jeweils eine eigene Verbindung mit ihrem Gegenstück auf der anderen Seite bilden. Zusätzlich ist jeder Rechner mit einer On-Board-Netzwerkkarte ausgestattet, um als Gateway-Rechner fungieren zu können. Der Versuchsaufbau folgt dem in Abbildung 8.1 dargestellten Modell mit drei unabhängigen Netzwerkverbindungen.

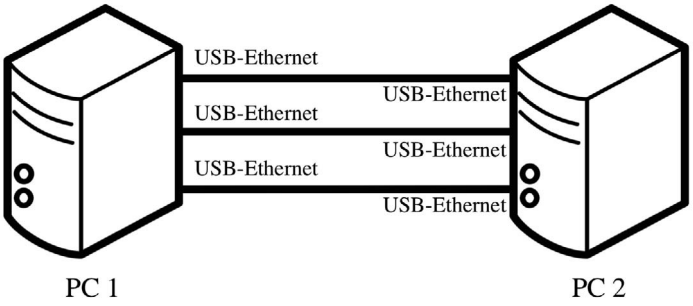

Abbildung 8.1: PC im rMPTCP-Versuchsaufbau

Um einen Pfadwechsel mit Ersatz-Subflows vorzunehmen, wird ein Switch zwischen die Verbindungen geschaltet. Über diesen können zwei oder drei Pfade verbunden werden. Die angeschlossenen Pfade sind dann nicht mehr voneinander unabhängig. Ein Switch wird beim Testen des adaptiven Pfadmanagements eingesetzt. Er besitzt die folgenden Merkmale:

Netgear GS108T	
Unterstützte Datenübertragungsrate	10/100/1000 Mbit
Ports	8

Tabelle 8.2: Switch im rMPTCP-Versuchsaufbau

Für die Nachahmung bestimmter Netzwerkeigenschaften wird die Software *netem* [netem, 2017] verwendet. *netem* ist ein kommandozeilenbasiertes Programm, mit dem verschiedene Eigenschaften, Störungen und realitätsnahes Verhalten eines WANs emuliert und innerhalb der Simulation eingesetzt werden können. Hiermit können für jeden Netzwerkadapter unter anderem die folgenden Eigenschaften gesetzt werden:

- Latenzzeiten [ms]
- Maximale Datenübertragungsrate [kbit/s]
- Jitter [ms]
- Datensegmentverluste [%]

Zum Erfassen der gesendeten Pakete an den jeweiligen Netzwerkinterfaces werden die Programme *tshark* sowie *Wireshark* verwendet. Für die Auswertung der Daten kommen das Statistikprogramm *R* und *MS-Excel* zur Anwendung[44]. Für die Zeichnung der RDS-Diagramme wird das kommandozeilenbasierte Darstellungsprogramm *gnuplot* [Gnuplot, 2017] eingesetzt.

Zum Erzeugen von Datensegmenten und zum Messen der Datenübertragungsrate wird das Programm *iPerf* genutzt. Hinzu kommt eine selbst programmierte Software

[44] Diese wurden bereits in Kapitel 4.2 vorgestellt

unter *C*, mit deren Hilfe Datenpakete mit verschiedenen Eigenschaften erstellt und in regelmäßigen Abständen versendet werden können. Hierbei wird ein Zeitstempel auf Sender- und Empfängerseite erzeugt, der zur weiteren Auswertung genutzt wird. Die folgenden Parameter können hierbei justiert werden:

- Sendefrequenz
- Paketgröße
- Dauer der Sendung

8.2 Funktion des Redundanzquoten-Schedulers

Eine erste Implementierung des Schedulers erfolgte in Hinblick auf die rMPTCP Scheduler-Strategie 1.[45] Dabei wird eine garantierte leistungsbezogene Redundanz hinzugefügt, ohne auf die Nutzung der Datenübertragungsrate der Applikation Rücksicht zu nehmen. Jedes Segment wird gleichzeitig an alle Sockets versendet ohne darauf zu achten, ob der Subflow zum Senden bereit ist. Die Verfügbarkeit von Subflow-Sockets wird damit ignoriert. Durch unterschiedliche Pfad-Eigenschaften der Subflows (verschieden großes BDP) ist der Verlauf einer Datensendung nicht mehr synchron. Das folgende RDS zeigt den Verlauf einer einfachen Implementierung von rMPTCP unter Hinzunahme einer Pfadheterogenität:

- *eth0 (blau)*: 100 Mbit
- *eth1 (rot)*: 100 Mbit
- *eth2 (orange)*: 40 Mbit

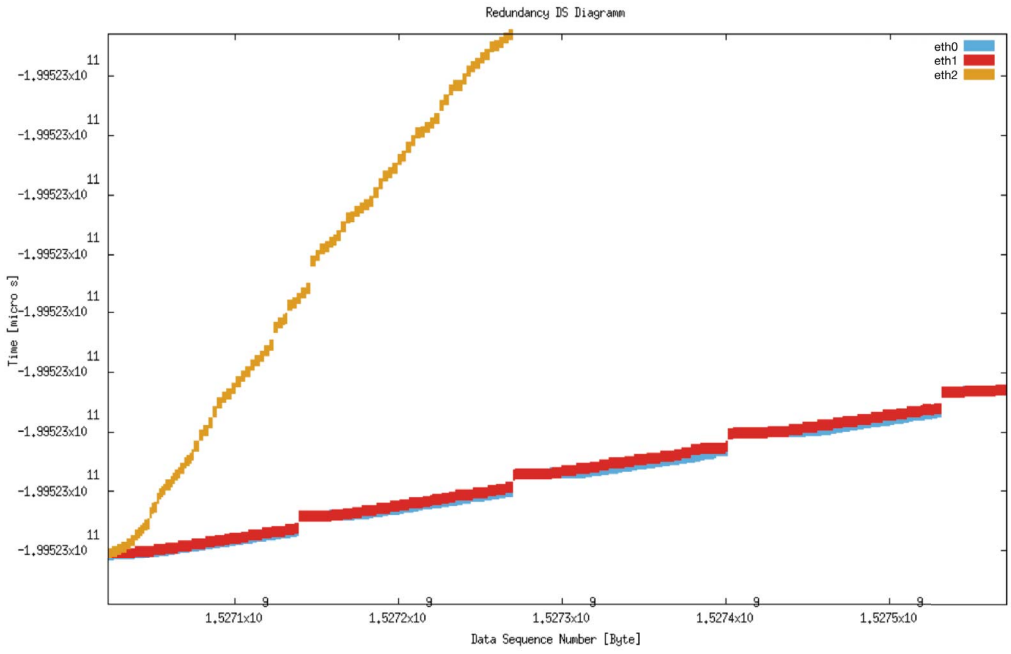

Abbildung 8.2: Redundanz-Datensequenzdiagramm eines nicht-synchronen Verlaufs in rMPTCP

[45] Vgl. Kapitel 6.2

Während *eth0* und *eth1* zeitlich synchron verlaufen, ist bei *eth2* eine starke Abweichung zu erkennen. Je weiter die Zeit fortschreitet, desto größer wird der zeitliche Abstand zwischen den Segmenten. Ab einem bestimmten Punkt ist die Latenzangleichungs-Grenze[46] erreicht und die Segmente auf *eth2* sind nicht mehr für eine Verbesserung nutzbar. Da *eth2* weniger Datenübertragungsrate besitzt als die anderen beiden Netzwerkinterfaces, können die Segmente nicht in ausreichender Form gesendet werden; die Verbindung „hinkt“ hinterher.

Unter Verwendung einer variabel angepassten Redundanzquote konnte das oben beschriebene Problem verhindert werden. Abbildung 8.3 zeigt rMPTCP mit einer variablen Verteilung und einer garantierten Redundanzquote von $Q = 2$. Hierbei wird zunächst eine primäre Verbindung über eth0 (blau) geschlossen. Nach erfolgtem Handshake beginnen ebenfalls die anderen Subflows eth1 und eth2 (rot, orange) mit dem Senden.

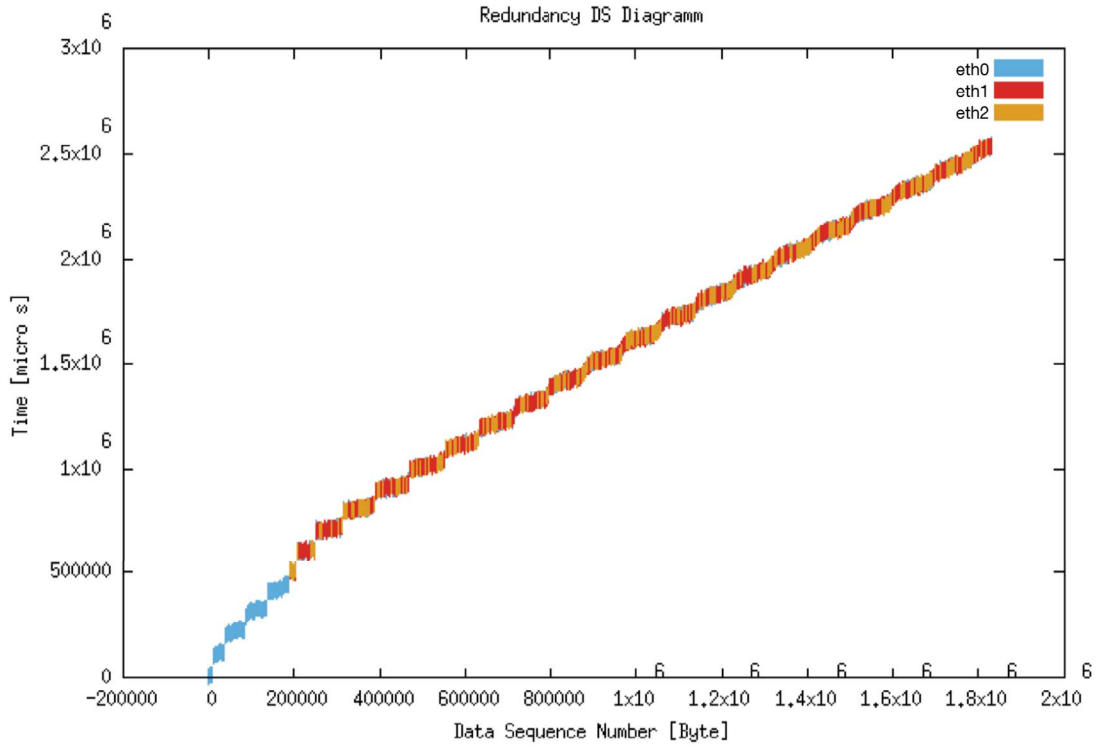

Abbildung 8.3: Redundanz-Datensequenzdiagramm mit einer variablen Redundanzquote in rMPTCP

Die stufenlose Anpassung der Redundanz an Gegebenheiten des Netzwerks wurde mit Hilfe der Formeln 6.19 und 6.20 modelliert.[47] Ziel war es hierbei, die effektive Datenübertragungsrate in Abhängigkeit von der Subflow-Anzahl, ihrer verfügbaren Datenübertragungsraten und der gewünschten Redundanzquote zu bestimmen.

Um die vollständige Datenübertragungsrate auszunutzen wurde mithilfe von *iPerf* eine Übertragung initiiert. Unter Berücksichtigung einer maximalen Netto-Datenrate von *93 Mbit/s* wurden mithilfe von *netem* die folgenden Dienstgütebeschränkungen eingestellt:

[46] Vgl. hierzu Ausführungen in Kapitel 6.4.1
[47] Vgl. hierzu Ausführungen in Kapitel 6.2.1

#	B_0 [Mbit/s]	B_1 [Mbit/s]	B_2 [Mbit/s]
1	93	93	93
2	93	93	-
3	93	48	23

Tabelle 8.3: Pfad-Beschränkungen für die Bestätigung der Formel 6.19 und 6.20

Die folgende Abbildung 8.4 stellt die Beziehung zwischen Redundanz und effektiver Datenübertragungsrate dar. Die Messwerte spiegeln sich wieder und die Formel konnte damit bestätigt werden.

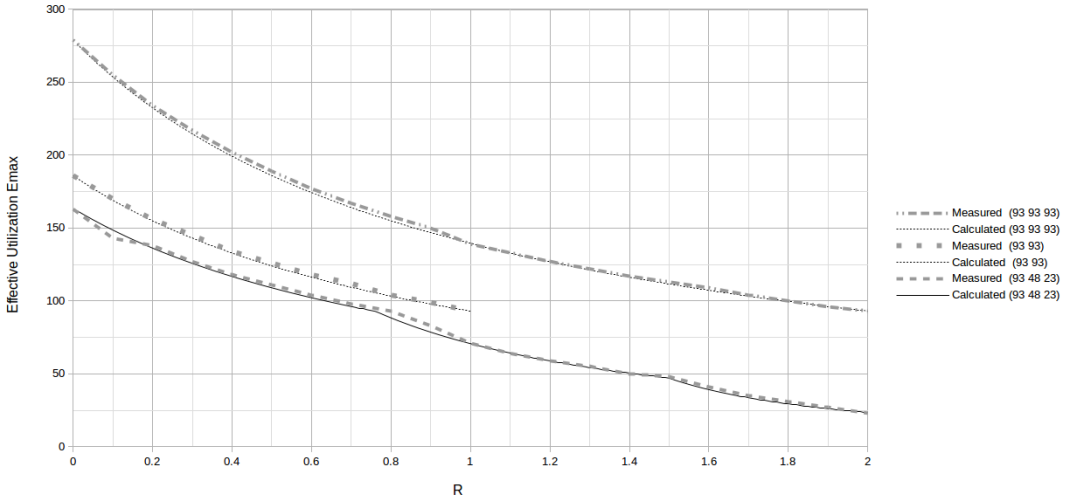

Abbildung 8.4: Validierung der Redundanz-Datenübertragungsrate-Beziehung, [Verbunt, 2017]

8.3 Out-of-Order-Vermeidung

Bei der Out-of-Order-Vermeidung wurden durch Priorisierung der besten Pfade alle Segmente zumindest einmal über einen priorisierten Subflow gesendet. Dies verhindert die Erzeugung von Out-of-Order-Datensegmenten, die durch einen Datenflusswechsel auf verschiedenen heterogenen Subflows entstehen. Ein „bester Pfad“ wird hierbei durch die größte Datenübertragungsrate und die kleinste RTT definiert. Die folgenden Dienstgütebeschränkungen wurden in diesem Szenario eingesetzt:

R	B_0 [Mbit/s]	B_1 [Mbit/s]	B_2 [Mbit/s]
1	48	93	93

Tabelle 8.4: Eingesetzte Pfad-Beschränkungen für die Out-of-Order-Vermeidung

Abbildung 8.5 zeigt eine RDS-Darstellung der Simulation aus Abbildung 6.15 mit aktivierter Out-of-Order-Vermeidung. Hierbei werden Subflow 0 (blau) und Subflow 1 (rot) als Support-Subflows erkannt sowie Subflow 3 (gelb) als priorisierter Main-Subflow.

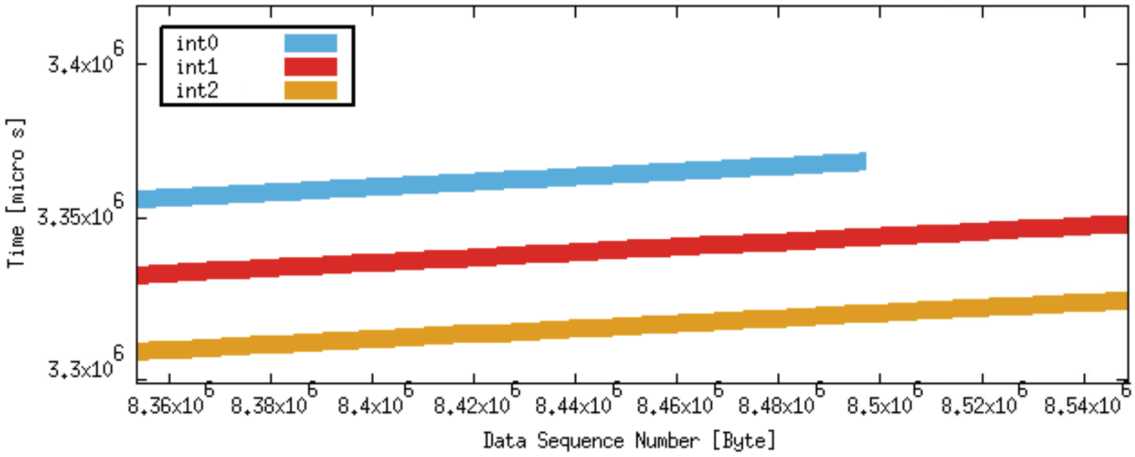

Abbildung 8.5: Out-of-Order-Prävention mit drei Subflows, [Verbunt, 2017]

Wie aus der Abbildung hervorgeht, wurden die Datenströme unter Nutzung der automatisierten Pfadevaluation so priorisiert, dass die Datenübertragungsrate von Subflow 2 (gelb) gehalten wird. Durch die voreingestellte Redundanzquote von $Q = 2$ wurde jedes Segment mindestens zweimal zur selben Zeit gesendet. Durch die Verfügbarkeit der beiden anderen Subflows resultierte eine kurzzeitige bessere Redundanzquote von $Q = 3$.

8.4 Empfangsverhalten in rMPTCP

Die Verwendung multipler Subflows und das einfache Annehmen des schnelleren Segments führte bei ähnlich homogenen Pfaden zu einem erratischen Verhalten beim Empfänger: Leicht unterschiedliche Timings zwischen den Segmenten wurden verstärkt, wobei der Empfänger immer zwischen mehreren Subflows hin und herspringt. Dieser *Inter-Subflow-Jitter* führt zu einem erhöhten Jitter anstatt zu einer Glättung.

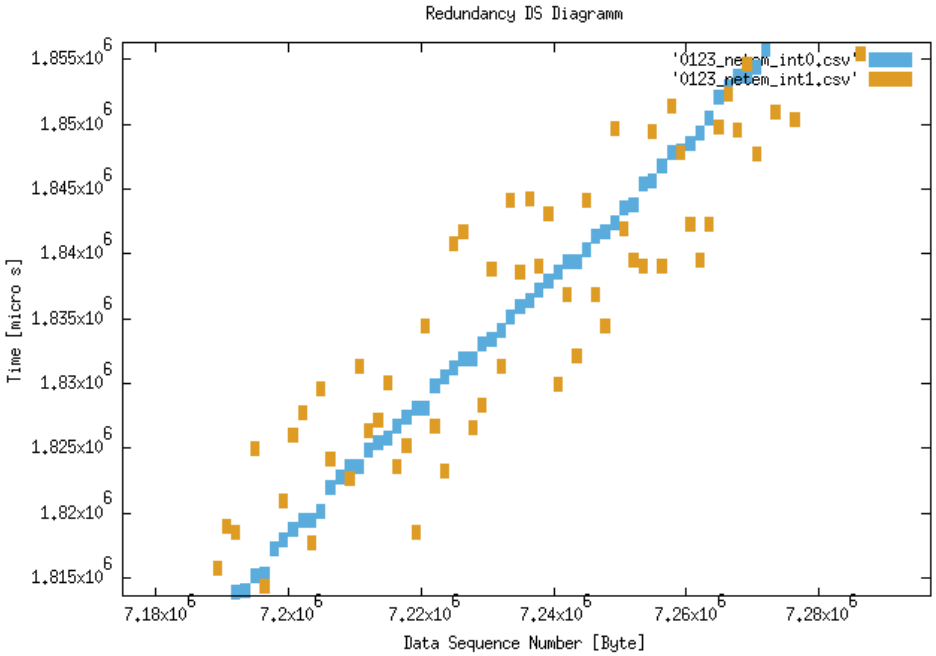

Abbildung 8.6: Redundanz-Datensegmentdiagramm ähnlich homogener Subflows, [Hunger et al., 2016]

Abbildung 8.6 stellt zwei fast homogene Subflows im Datensegmentdiagramm dar. Beide Subflows besitzen dieselbe durchschnittliche Latenzzeit, wobei einer der beiden Subflows einen durchschnittlichen Jitter von *+/- 10 ms* (gelb) aufweist. Bei der Applikation auf Empfängerseite wird in diesem Fall ein durchschnittlicher Jitter von *+/-5 ms* erzeugt, solange das zuerst empfangene Datensegment angenommen wird. Dies sind alle gelben Segmente unterhalb der blauen Reihung.

Für einen geglätteten Empfang auf Empfängerseite kann in diesem Fall eine Glättungsfunktion eingesetzt werden, die die hereinkommenden Pakete angleicht. Diese funktioniert, wenn die Datensegmente in regelmäßigen Abständen gesendet werden oder wenn innerhalb der TCP-Segmente ein Zeitstempel mitgesendet wird, aus dem die original abgesendeten Zeitabschnitte hervorgehen. Auf Empfängerseite wird eine Filterfunktion eingesetzt, die sich dem Sendemuster anpasst.

Für die aktuelle Implementierung von rMPTCP wurde zunächst eine einfache Filterfunktion eingesetzt, die die zeitliche Distanz zwischen den hereinkommenden Paketen adaptiert und einen Erwartungswert berechnet, um außergewöhnlich früh ankommende Segmente zwischen zu speichern. Die folgende Funktion beschreibt einen einfachen Tiefpassfilter, der die Ankunftszeit der Segmente glättet [Hunger et al., 2016]:

$$E_{new} = (1 - \alpha) \cdot E_{old} + \alpha \cdot C \tag{8.1}$$

Hierbei beschreibt E den Erwartungswert und C ist der gemessene Zeitstempel des hereinkommenden Segments. α steht für den Filterkoeffizienten, der als Einstellwert für die Stärke der Funktion verwendet wird. Dieser Wert wird am Anfang auf *0,1 (Lernrate 10 %)* eingestellt.

Für eine Evaluierung wurden zwei Pfade mittels *netem* mit leicht heterogenen Eigenschaften eingesetzt. Datenpakete wurden bei einer regelmäßigen Frequenz von *20 Hz* gesendet. Ein zusätzlicher Jitter von *+/-5 ms* an einem der beiden Subflows führte zu einem sprunghaften Verhalten am Empfänger, wie in Abbildung 8.6 dargestellt. Die folgende Abbildung 8.7 stellt den Verlauf auf Empfängerseite dar. Die Aktivierung des Tiefpassfilters führte zu einem Verlauf, der in Abbildung 8.8 dargestellt ist.

In Abbildung 8.8 ist ein Jitter von etwa *1 ms* zu erkennen, der innerhalb der möglichen Messfehler liegt. Am Beginn der Übertragung (bis Datensegmentnummer 50) ist eine anfängliche Lernphase zu erkennen. Diese Lernphase kann durch höher entwickelte Tiefpassfilter weiter verbessert werden. Die Standardabweichung der Messungen ist *0,65 ms* mit aktivem Tiefpass. Im Vergleich hierzu besteht eine Standardabweichung von *1,4 ms* ohne Tiefpass.

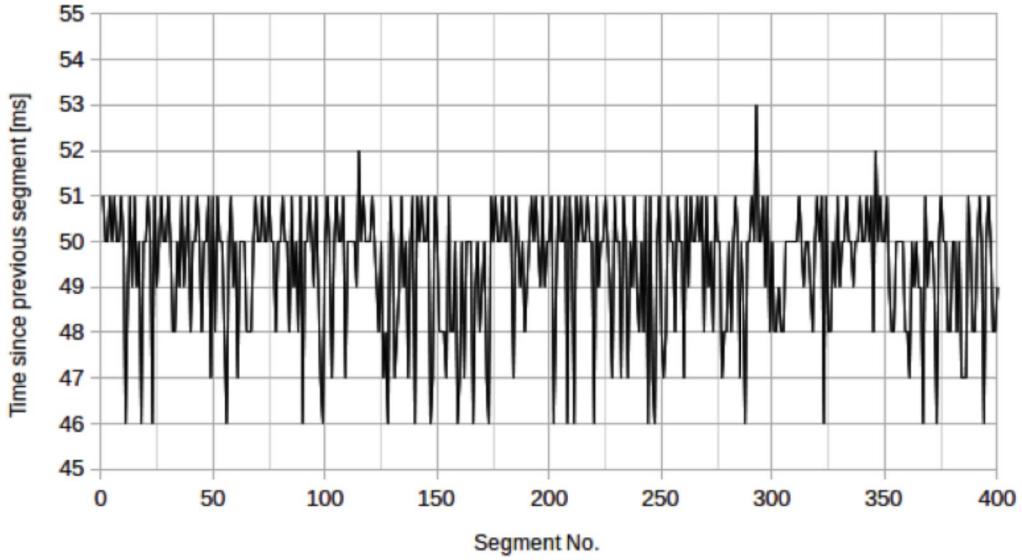

Abbildung 8.7: rMPTCP-Empfangsverhalten bei homogenen Subflows ohne Jitter-Kompensierung, [Hunger et al., 2016]

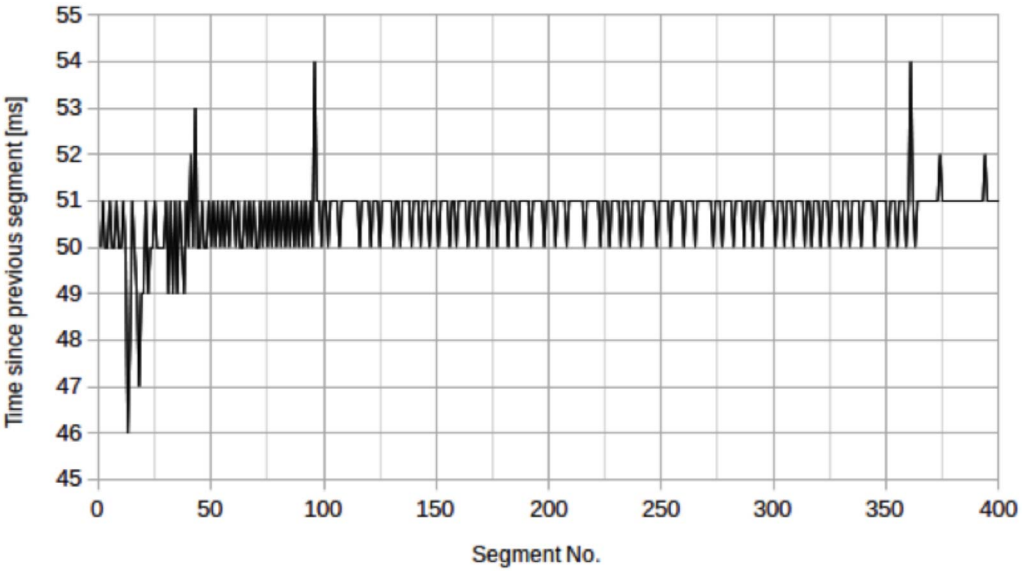

Abbildung 8.8: rMPTCP-Empfangsverhalten mit aktivierter Jitter-Kompensierung, [Hunger et al., 2016]

Weitere Messreihen sind in der nachfolgenden Tabelle 8.5 dargestellt. Es wurden folgende Einstellungen vorgenommen:

- Lernrate: 50 %
- Ausreißer wurden entfernt
- ca. 400 Segmente pro Versuch
- Datensegment-Payload: 13 Bytes

Sende-frequenz [Hz]	Sende-periode [ms]	Delay ± Jitter Interface 1 [ms]	Delay ± Jitter Interface 2 [ms]	Std.-Abw. Tiefpass [ms]	Std.-Abw. ohne Tiefpass [ms]
10	100	0 ± 0	0 ± 0	0,0	0,36
25	40	0 ± 0	0 ± 0	0,68	0,46
50	20	0 ± 0	0 ± 0	0,0	0,47
100	10	0 ± 0	0 ± 0	0,3	0,28
10	100	10 ± 0	10 ± 2	0,22	0,76
25	40	10 ± 0	10 ± 2	0,23	0,70
50	20	10 ± 0	10 ± 2	0,47	0,87
100	10	10 ± 0	10 ± 2	0,34	0,62
10	100	50 ± 0	50 ± 5	0,59	1,94
25	40	50 ± 0	50 ± 5	1,84	1,28
50	20	50 ± 0	50 ± 5	0,65	1,45
100	10	50 ± 0	50 ± 5	0,6	1,35
10	100	100 ± 0	100 ± 10	0,75	4,05
25	40	100 ± 0	100 ± 10	0,9	3,05
50	20	100 ± 0	100 ± 10	0,65	2,82
100	10	100 ± 0	100 ± 10	1,15	2,91

Tabelle 8.5: rMPTCP-Tiefpass-Messreihe

Die Tabelle zeigt eine Reduzierung des Jitters von > *50 %*. Der Empfangsfilter kann bei regelmäßigen Sendeformen eingesetzt werden. Dies könnte ein Telemetriesystem oder eine Telepointersteuerung sein, die aus einer Datensendung mit regelmäßigen Segmenten besteht.

In der Realität tritt dieser Problemfall äußerst selten auf, da homogene Pfade mit nahezu gleicher Latenz und einem leicht unterschiedlichem Jitter hierfür Voraussetzung sind. Für die nachfolgenden Messungen wird daher keine Empfangsfilterung eingesetzt.

8.5 Störungserkennung

Die Erkennung einer signifikanten Störung auf einem Subflow wurde mit den folgenden zwei Methoden realisiert:

1. durch Beobachtung der Sendeperiode
2. durch Beobachtung der effektiven Datenübertragungsrate unter Bestätigung durch die TCP-Überlastkontrolle

Bei der Auswahl der anzuwendenden Methode ist für eine Evaluation dieser Funktion die Reaktionszeit eine signifikante Größe. Sie gibt an, wie lang die Zeitspanne ist, die zwischen der eigentlichen Störung und ihrer Erkennung liegt. Nach einer erfolgreichen Erkennung können die Maßnahmen zur Ausfallsicherung greifen.

Zur Messung der Reaktionszeit wurden Daten mithilfe von *iPerf* und unter Nutzung von drei unabhängigen Subflows übertragen. Um einen Störungszustand hervorzurufen, wurde die Verbindung eines einzelnen Subflows mithilfe von *netem* und eines erhöhten Datenpaketverlusts unbrauchbar gemacht.

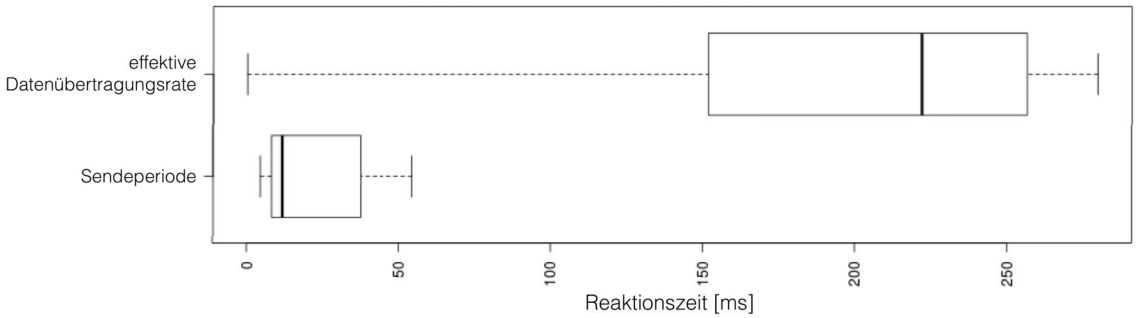

Abbildung 8.9: Vergleich der Reaktionszeit der Ausfallerkennung

Abbildung 8.9 gibt eine Übersicht über die gemessenen Reaktionszeiten. Hierbei fällt ein deutlicher Unterschied zwischen den beiden Methoden auf: Die Reaktionszeit der Sendeperiode besitzt eine sehr viel kürzere Erkennungszeit als jene über die effektive Datenübertragungsrate. Hinzu kommt eine weitaus geringere Breite der Boxplots bei der Sendeperiode.

In einigen Fällen greift der Algorithmus der Sendeperiode jedoch nicht bzw. entwickelt sich etwas träger. Dies ist der Verlängerung des wiederholten Sendens bei einem Timeout geschuldet: Eine Messung der Sendeperiode kann nur vorgenommen werden, wenn ein Senden erfolgt. Durch Ausdehnung des Warte-Timers nach einem Timeout unter TCP ist eine Erkennung erst nach der Beendigung des Warte-Timers möglich, wenn ein neues Datensegment versendet wird. Dies erhöht zwar die Erkennungszeit, liegt aber immer noch signifikant unter der Erkennungszeit der effektiven Datenübertragungsrate.

Andererseits ist die Erkennung einer Störung mithilfe der Sendeperiode nicht in allen Fällen gewährleistet. Bei unterschiedlichen Gegebenheiten konnte die Sendeperiode nur einen Ausfallerkennungswert von *46 %* erzeugen. Im Vergleich hierzu lag die Erkennung mithilfe der Beobachtung der effektiven Datenübertragungsrate stets bei *100 %*. Eine nähere Untersuchung des Problems erbrachte, dass ein störungsbedingter Anstieg der Sendeperiode in einigen Fällen eines Ausfalls zu langsam geschah. Der maximale Wert der Sendeperiode wird nicht schnell genug überschritten. Dies ist dem Überlastprotokoll von TCP geschuldet, das ein langsames Herantasten an die mögliche Verbindungsqualität bewirkt.

Durch Nutzung beider Methoden wird eine Erkennung sichergestellt. Je nachdem, welche Methode zuerst greift, entscheidet dies über die Reaktion einer Störungserkennung.

8.6 Adaptive Redundanz

Zur Anpassung der leistungsbezogenen Redundanz sowie der Datenübertragungsrate an einen bevorstehenden Subflow-Ausfall wurde der Algorithmus der adaptiven Redundanz entwickelt. Für eine Evaluation des Algorithmus wurden Tests einer Übertragung mit und ohne aktivierten Algorithmus durchgeführt.

Abbildung 8.10 zeigt eine Übertragung ohne adaptiven Algorithmus mit erzwungener Redundanz von mindestens $Q = 2$. In der Abbildung werden drei Phasen betrachtet.

1. In Phase I erfolgte eine Übertragung ohne Verbindungsprobleme über drei Subflows.
2. In Phase II kam es zu einer Abweichung der regulären Übertragung: Subflow int1 (rot) leidet unter mehreren Datensegmentverlusten. Die effektive Datenübertragungsrate sinkt. Durch die zusätzliche leistungsbezogene Redundanz kommen jedoch alle Datensegmente rechtzeitig beim Empfänger an. Die Segmente erfahren durch die geringer werdende Datenübertragungsrate eine leichte Verschiebung nach oben, d.h. sie kommen später an.
3. In Phase III ist die effektive Datenübertragungsrate soweit gesunken, dass die gewünschte Redundanz nicht mehr gehalten werden kann. Die Verbindung bricht ab. Ein großer Sprung von ca. *600 ms* entsteht durch die höhere Datenlast, die von zwei Subflows erfüllt werden muss, aber nicht unterstützt wird.

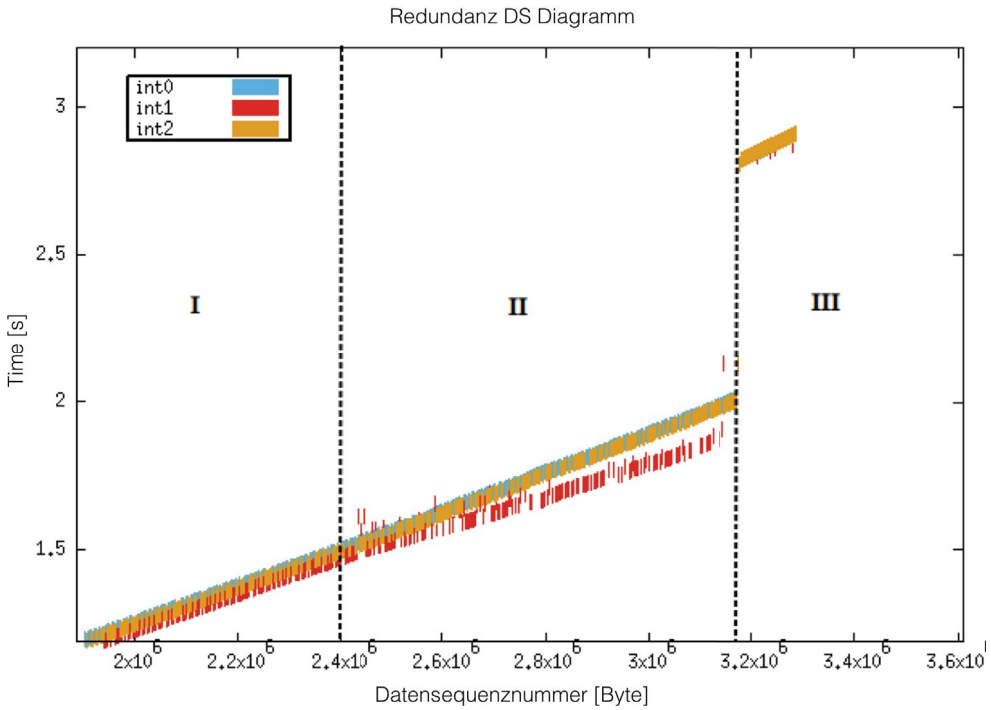

Abbildung 8.10: Datenübertragung einer gestörten Verbindung ohne adaptiven Algorithmus

In Abbildung 8.11 ist eine Datenübertragung mit aktiviertem adaptiven Redundanz-Algorithmus zu sehen. Diese besitzt ebenfalls eine garantierte leistungsbezogene Redundanz von mindestens $Q = 2$. Die Abbildung unterscheidet wiederum drei Phasen:

1. Phase I beschreibt eine funktionierende Verbindung über drei Subflows ohne Probleme.
2. In Phase II leidet Subflow int1 (rot) unter Segmentverlusten. Der zeitliche Abstand zwischen den Subflows vergrößert sich etwas durch die zusätzliche Last auf Subflows int0 (blau) und int2 (gelb).
3. Phase III beschreibt die Aktivierung des adaptiven Redundanz-Algorithmus. Die effektive Datenübertragungsrate sinkt soweit, dass der Schwellwert er-

reicht wird. Die leistungsbezogene Redundanz wird verringert, sodass die beiden anderen Subflows int0 (blau) und int2 (gelb) weiterhin mit der Last zurechtkommen und einen zeitlich geringer steigenden Verlauf aufweisen. Ein Verbindungsabbruch sowie größere Sprünge werden vermieden und eine optimale leistungsbezogene Redundanz wird angewandt.

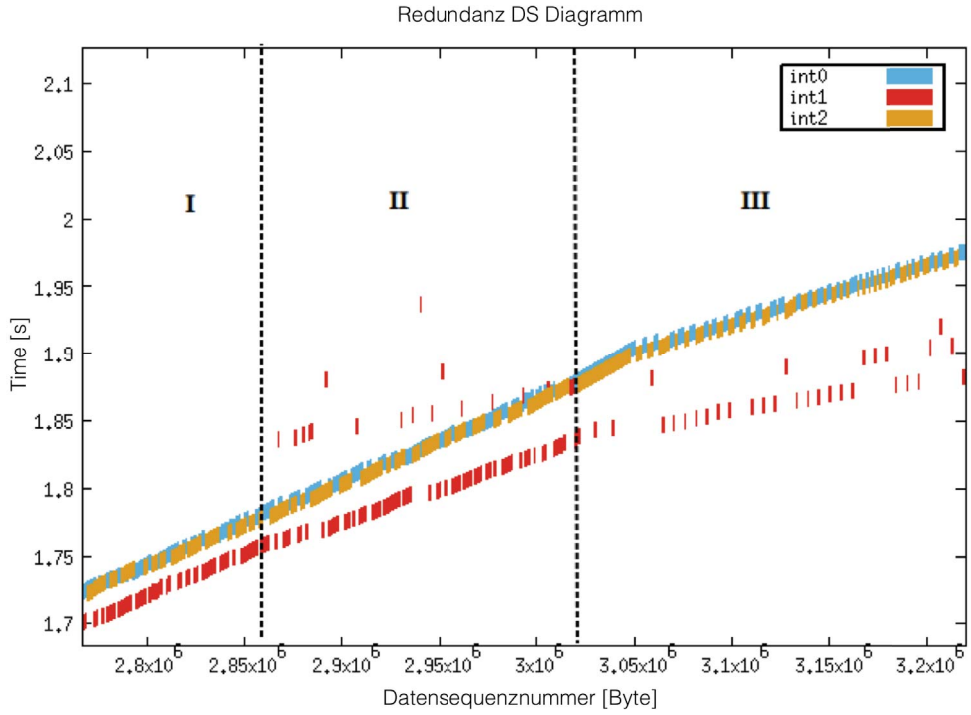

Abbildung 8.11: Datenübertragung einer ausfallenden Verbindung mit adaptiver Redundanz

Weitere Tests wurden mithilfe von *iPerf* durchgeführt, um die Datenübertragungsrate auf Seiten der Applikation zu überprüfen. Hierbei sollte festgestellt werden, inwieweit die Datenübertragungsrate gehalten werden kann, um die Ansprüche der Anwendung erfüllen zu können, während der adaptive Algorithmus in Funktion tritt. Der Test wurde mit der folgenden Metrik durchgeführt:

Parameter	Wert
Redundanz	Q = 2
Datenübertragungsrate Subflow 1	30 Mbit/s
Datenübertragungsrate Subflow 2	10 Mbit/s
Datenübertragungsrate Subflow 3	5 Mbit/s
Datenverlust Subflow 1	0 %
Datenverlust Subflow 2	20 % (5 s – 10 s); sonst 0 %
Datenverlust Subflow 3	0 %

Tabelle 8.6: Metriken für die Messung der effektiven Datenübertragungsrate

Abbildung 8.12 zeigt ein Liniendiagramm mit drei Testläufen bei gleichen Konditionen mit den oben genannten Metriken. Wie aus Tabelle 8.6 hervorgeht, erfuhr Subflow 2 ab dem Zeitpunkt $t = 5\ s$ einen Datenverlust von *20 %*. Die automatische Regulierung der Redundanz führte zu einer grundsätzlichen Anpassung. Aus der Aktivierung

des adaptiven Algorithmus resultierte eine leicht erhöhte Datenübertragungsrate in der zweiten Hälfte der Grafik. Dies war die Folge einer etwas niedrigeren leistungsbezogenen Redundanz, als tatsächlich benötigt wurde.

Eine weitere Abweichung zeigte Testlauf 3, der bei etwa *7 s* einen kurzzeitigen Abfall in der Datenübertragungsrate unterhalb des vorherigen Niveaus aufwies. Dieser Abfall trat sporadisch und nur kurzzeitig auf. Er wurde sofort durch eine Gegenkompensation nachreguliert, was wiederum zu einer höheren effektiven Datenübertragungsrate führte.

Die Datenübertragungsrate konnte grundsätzlich gehalten werden, obwohl einer der Subflows Datenverluste von *20 %* erlitt.

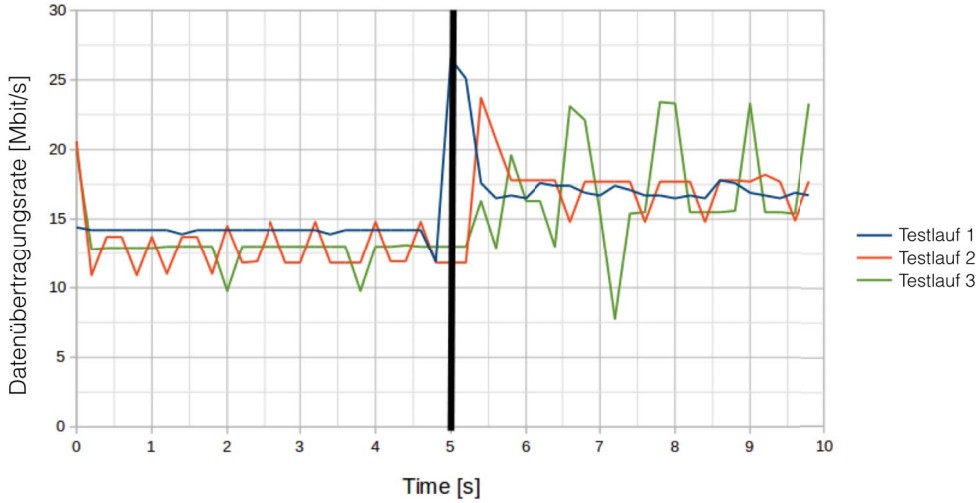

Abbildung 8.12: Liniendiagrammdarstellung der Datenübertragungsrate im Störungsfall mit adaptiver Redundanz

Ein weiterer Test maß die Datenübertragungsrate der einzelnen Subflows und der damit gebotenen leistungsbezogenen Redundanz. Ein Test wurde mit der folgenden Metrik in Tabelle 8.7 durchgeführt. Hierbei unterlag Subflow 3 zwischen *10 s* und *20 s* einem Datenverlust von *20 %*.

Parameter	Wert
Redundanz	Q = 2,3
Subflow-Versagen Schwellwert	90 %
Sendeperiode Schwellwert	300 %
Datenübertragungsrate Subflow 1	10 Mbit/s
Datenübertragungsrate Subflow 2	30 Mbit/s
Datenübertragungsrate Subflow 3	20 Mbit/s
Datenverlust Subflow 1	0 %
Datenverlust Subflow 2	0 %
Datenverlust Subflow 3	20 % (10 s – 20 s); sonst 0 %

Tabelle 8.7: Metriken für die Messung der Datenübertragungsrate/Subflow

Abbildung 8.13 zeigt eine Darstellung der Datenübertragungsrate der einzelnen Subflows ohne adaptiven Algorithmus. Zu sehen ist ein deutlicher Einbruch der Datenübertragungsrate, wenn eine garantierte leistungsbezogene Redundanz zum Tragen kommt.

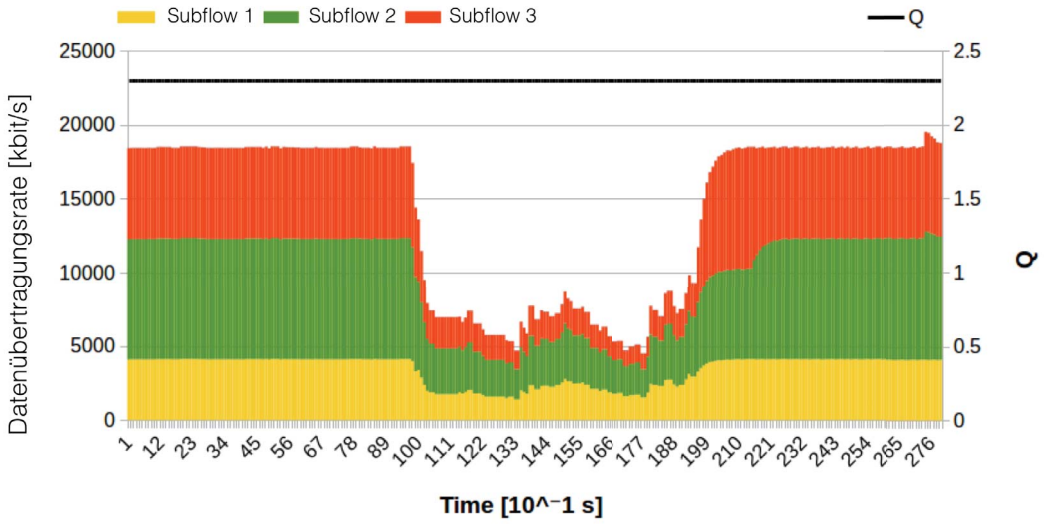

Abbildung 8.13: Messung der Datenübertragungsrate/Subflow ohne adaptiven Algorithmus

Abbildung 8.14 zeigt dasselbe Szenario mit eingeschaltetem adaptiven Algorithmus. Die Nachregulierung der Redundanzquote kann einen größeren Einbruch der effektiven Datenübertragungsrate verhindern. Auf Kosten der leistungsbezogenen Redundanz kann die Übertragung wie gehabt fortgesetzt werden, ohne dass die Anwendung hiervon beeinträchtigt wird. Sobald die Qualität von Subflow 3 sich bessert, wird die Redundanzquote wieder auf den Ursprungswert angehoben.

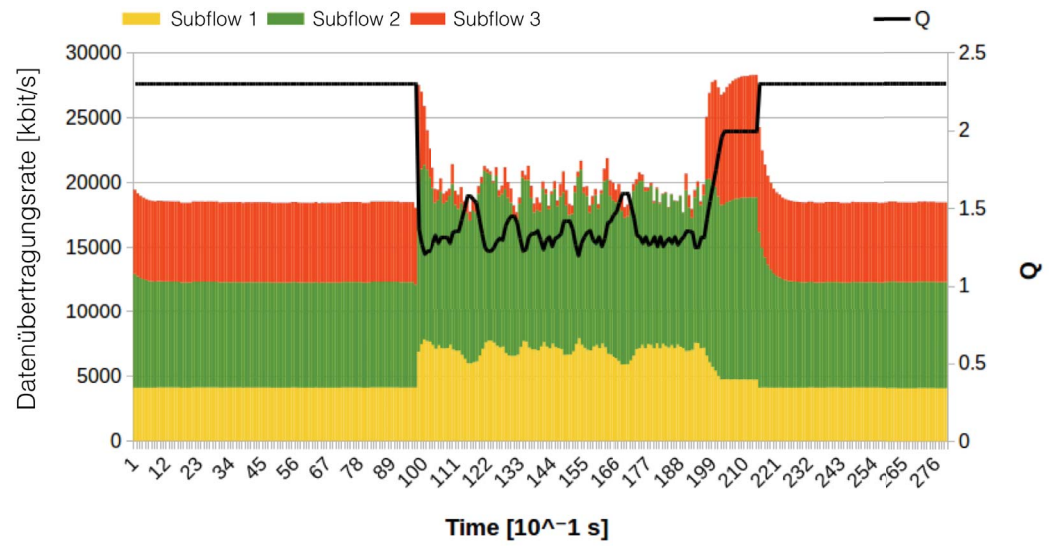

Abbildung 8.14: Messung der Datenübertragungsrate/Subflow mit adaptiver Redundanz

Die oben dargestellten Messungen haben ergeben, dass ein adaptiver Redundanz-Algorithmus wie geplant funktioniert. Die leistungsbezogene Redundanz wird im Fall einer Verschlechterung der Qualität oder bei einem Ausfall neu berechnet. Die Daten werden so verteilt, dass die Applikation hiervon nicht beeinträchtigt wird, aber ein Maximum an möglicher leistungsbezogener Redundanz, verteilt auf die verschiedenen Subflows, erhalten bleibt. Wenn die Störung sich wieder auflöst, kehrt die Verbindung zu ihren Ursprungswerten zurück.

8.7 Adaptives Pfadmanagement

Das adaptive Pfadmanagement erstellt zwei neue Subflows, wenn der Ausfall eines Subflows erkannt wird. Die Datenübertragungsraten werden neu verteilt, sodass eine fortlaufende Datenübertragungsrate ermöglicht wird. Für eine Messung wurden die Metriken aus Tabelle 8.6 verwendet.

Abbildung 8.15 zeigt das RDS für den Fall mit drei Subflows. Subflow 2 (blau) leidet unter einem Ausfall, der mithilfe von zwei neuerstellten Subflows (Subflow 3, grün und Subflow 4, orange) ausgeglichen wird. Subflow 0 und Subflow 1 funktionieren wie gehabt.

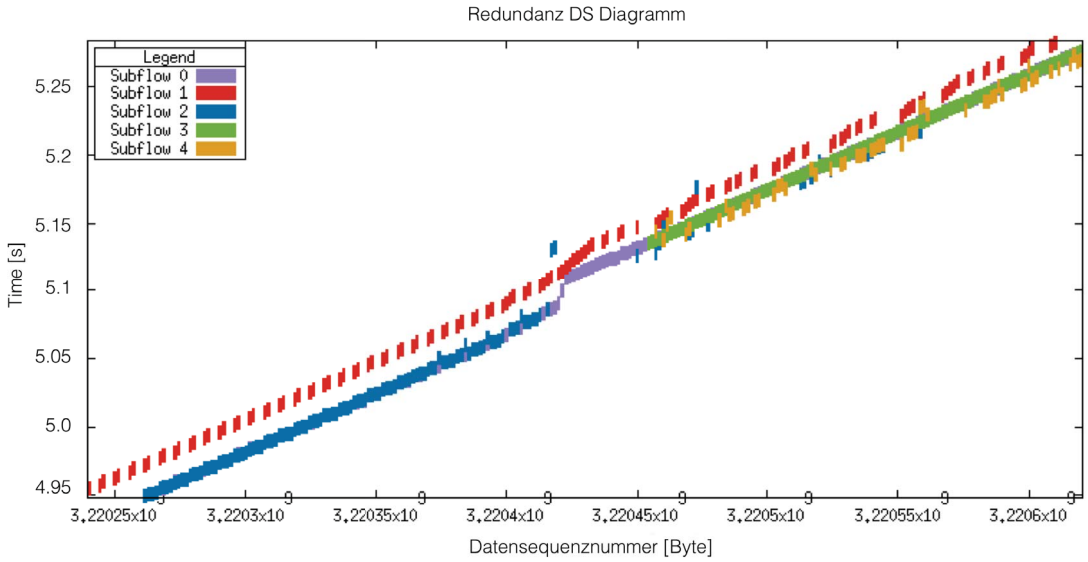

Abbildung 8.15: Datenübertragung einer ausfallenden Verbindung mit adaptivem Pfadmanagement

Eine weitere Messung wurde mithilfe von *iPerf* durchgeführt, um die Datenübertragungsrate der Anwendung bei einem Subflow-Wechsel zu beobachten. Abbildung 8.16 zeigt die erhaltende Datenübertragungsrate für denselben Testfall aus Tabelle 8.7 mit aktiviertem adaptiven Pfadmanagement. Wie bei der Betrachtung des Graphen deutlich wird, gelingt es dem adaptiven Pfadmanagement, die veranschlagte Datenübertragungsrate durch die Ersatz-Subflows zu halten.

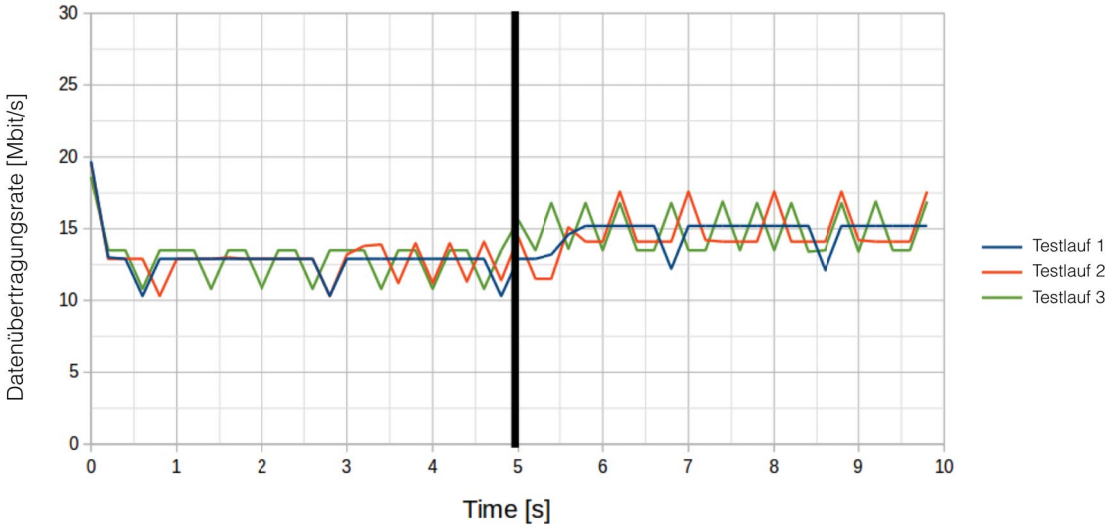

Abbildung 8.16: Liniendiagrammdarstellung der Datenübertragungsrate im Störungsfall mit adaptivem Pfadmanagement

Die Datenübertragungsrate der einzelnen Subflows wurde mit demselben Szenario aus Tabelle 8.7 auch für das adaptive Pfadmanagement überprüft. Abbildung 8.17 zeigt das Ergebnis der Datenübertragungsrate bei einem Ausfall von Subflow 3 und der Aktivierung von zwei Ersatz-Subflows (blau und braun). Die Messung bestätigte gleichfalls die Hypothese, dass einer der beiden Subflows die Störung umgehen kann. Nach *20 s* war der Original-Subflow wieder verfügbar, was sich in einer kurzzeitigen Erhöhung der Datenübertragungsrate bemerkbar machte. Während des Ausfalls von Subflow 3 konnte die vorherige Datenübertragungsrate nicht komplett gehalten werden, da Ersatz-Subflow 1 die Datenübertragungsrate, wie vorhergesehen, nur zum Teil stützen konnte. Ein größerer Abfall der Datenübertragungsrate wurde verhindert.

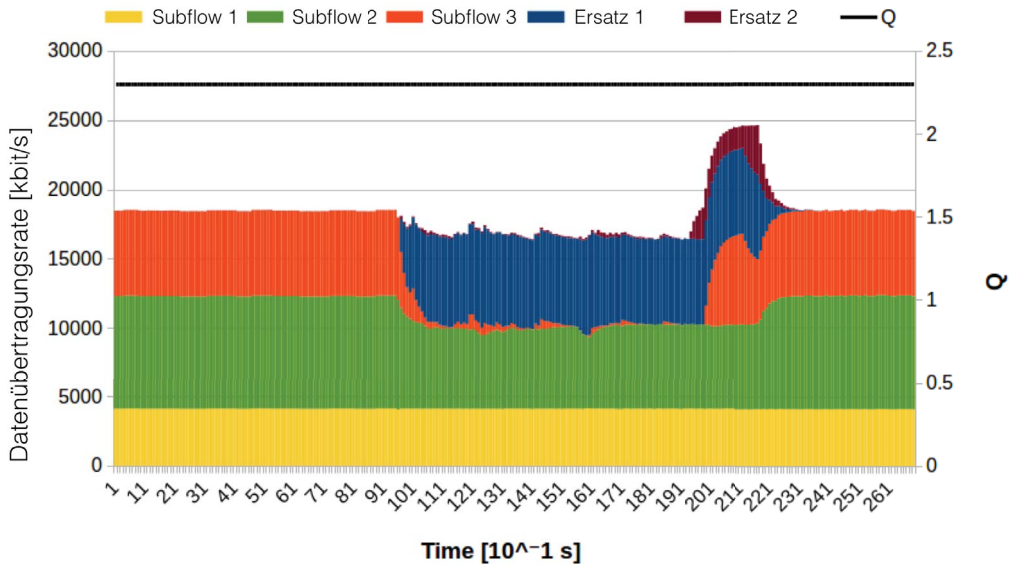

Abbildung 8.17: Messung der Datenübertragungsrate/Subflow mit adaptivem Pfadmanagement

Die in diesem Kapitel aufgezeigten Tests belegen die gewünschte Funktionalität des entwickelten Pfadmanagements. Ein größeres Abfallen der Datenübertragungsrate wurde für den Fall eines inszenierten Subflow-Ausfalls verhindert.

8.8 Adaptive Redundanz in Kombination mit Pfad-Management

Um die beiden Algorithmen kombiniert zu testen, wurde zunächst die Datenübertragungsrate in drei aufeinanderfolgenden Tests mit gleicher Metrik aus Tabelle 8.6 gemessen.

Abbildung 8.18 zeigt die Datenübertragungsrate für drei Testläufe. Grundsätzlich kann die Datenübertragungsrate gehalten werden. Für Testlauf 2 ergab sich ein kurzer Abfall der Datenübertagungsrate, da die Ausfallerkennung etwas länger benötigte. Eine kombinierte Nutzung der beiden Algorithmen hat keine größeren Probleme erkennen lassen.

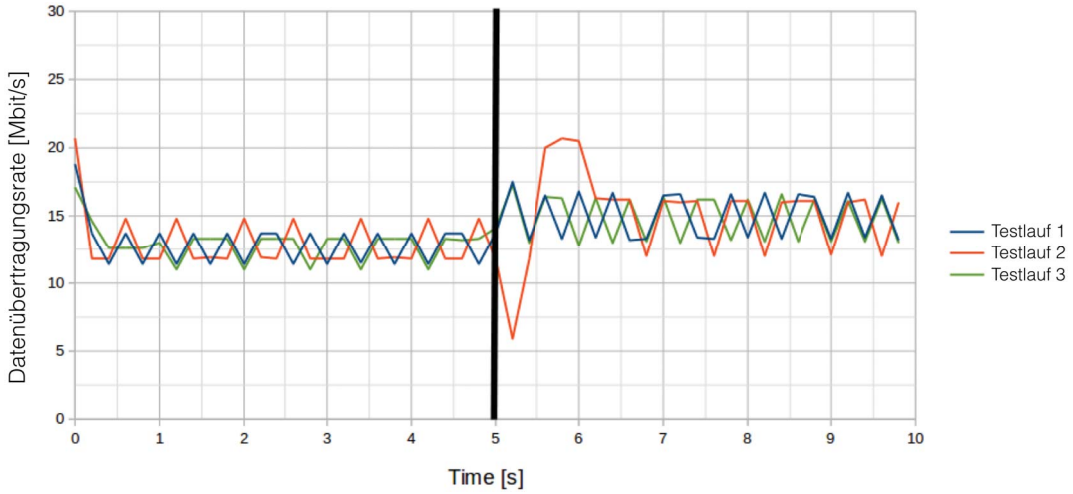

Abbildung 8.18: Liniendiagrammdarstellung der Datenübertragungsrate im Störungsfall mit adaptivem Pfadmanagement und adaptiver Redundanz

Abbildung 8.19 stellt die Datenübertragungsrate der einzelnen Subflows für das Szenario aus Tabelle 8.7 dar. Dem Ausfall von Subflow 3 folgte zunächst ein kurzzeitiger Einbruch der Redundanzquote. Nach Aktivwerden des Pfadmanagements und des Einspringens der Ersatz-Subflows konnte die Datenübertragungsrate gehalten werden. Nach kurzer Zeit wurde eine maximale Redundanzquote von $Q = 2$ erreicht. Subflow 3 wurde weiterhin für Sondierungszwecke verwendet.

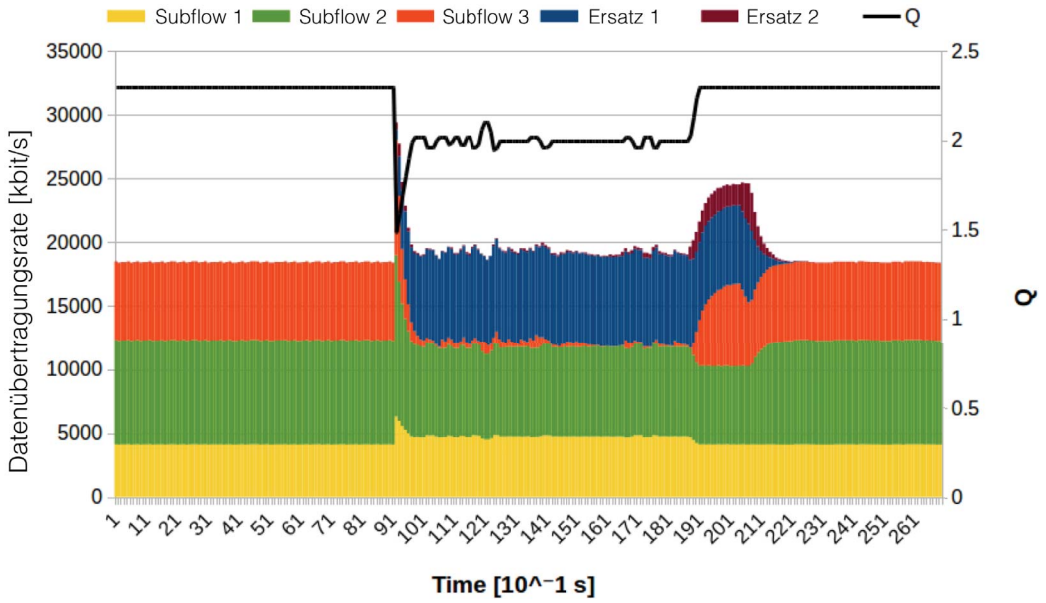

Abbildung 8.19: Datenübertragungsrate bei adaptivem Pfadmanagement und Redundanzausgleich

Das Ergebnis zeigte eine bessere Nutzung der verfügbaren Datenübertragungsraten unter Nutzung einer maximalen Redundanzquote, als dies bei der Nutzung der alleinigen adaptiven Algorithmen der Fall wäre.

Für einen Vergleich zwischen der Nutzung der redundanten Adaptierung mit und ohne Verwendung des adaptiven Pfadmanagements wird die folgende Metrik in Tabelle 8.8 für weitere Versuche erstellt.

	Subflow 1	**Subflow 2**	**Subflow 3**
Datenübertragungsrate [Mbit/s]	50	25	5

Tabelle 8.8: Metrik für Vergleich zwischen adaptiven Algorithmen mit/ohne adaptivem Pfadmanagement

Abbildung 8.20 zeigt einen direkten Vergleich der leistungsbezogenen Redundanz mit adaptivem Pfadmanagement und ohne diese, nachdem ein Subflow-Ausfall stattgefunden hat. Das Ergebnis zeigt eine deutliche Verbesserung bei der Verfügbarkeit von zusätzlicher Redundanz. Für viele Fälle ist damit eine Steigerung der Redundanz im Falle eines Ausfalls möglich.

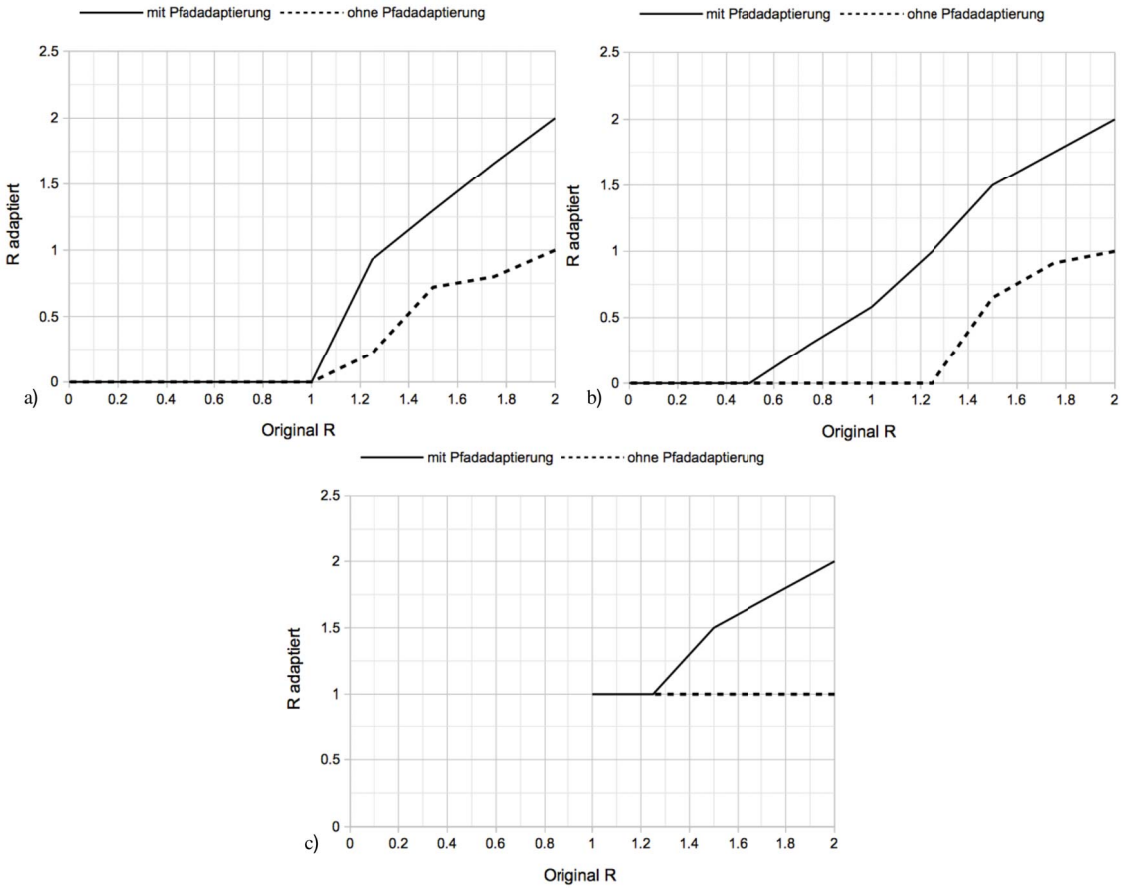

Abbildung 8.20: Messung bei Pfadausfalls mit/ohne adaptivem Algorithmus, a) Ausfall Subflow 1, b) Ausfall Subflow 2, c) Ausfall Subflow 3, [Verbunt, 2017]

8.9 Latenzevaluation unter rMPTCP

Die Evaluation des Latenz- und Jitter-Ausgleichs wurde mit dem selbstprogrammierten Paketgenerator durchgeführt. Hiermit können Segmente mit einer festen Frequenz gesendet und Veränderung der zeitlichen Abstände zwischen den Segmenten beobachtet werden. Daraus lassen sich Erkenntnisse hinsichtlich des Übertragungsmusters von rMPTCP ableiten. Unter Verwendung der Emulationssoftware *netem* werden bestimmte Netzwerkstörungen simuliert.

Eine typische Messung unter Jitter-Einfluss auf Empfangsseite wird durch den Boxplot in Abbildung 8.21 dargestellt. Hierbei wurden die folgenden Metriken verwendet, die den Werten beim Benutzen eines ARTPs auf der UDE/UKM-Verbindung folgen:

Parameter	Wert
Sendefrequenz	20 Hz
Segmentgröße	1 KB
Redundanzquote Q	3
Anzahl Subflows	3
Umlaufzeit	280 ms
Jitter	+/- 5 ms

Tabelle 8.9: Metrik für Jitter-Test

Die Abbildung 8.21 a) und b) stellen Boxplots der Übertragung unter rMPTCP dar und zwar einmal ohne adaptiven Redundanzalgorithmus und einmal mit. Dagegen zeigt Abbildung 8.21 c) eine herkömmliche TCP-Verbindung unter demselben Jitter-Einfluss. Bei der rMPTCP-Verbindung liegt ein deutlich kleinerer Median vor und die Quartile sowie die Whisker liegen sehr viel näher beieinander.

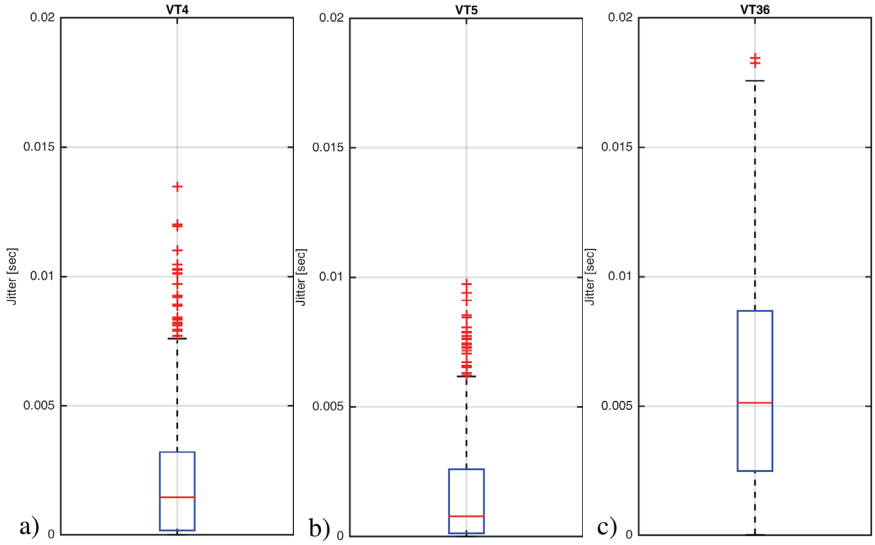

Abbildung 8.21: Jitter-Boxplot einer Übertragung unter Einfluss von +/- 5 ms Jitter: a) rMPTCP ohne adaptiven Redundanzalgorithmus, b) rMPTCP mit adaptivem Redundanzalgorithmus, c) herkömmliche TCP-Übertragung

In ergänzenden Tests wurde mithilfe von zusätzlichen Segmentverlusten eine weitere Verschlechterung der Verbindungsqualität realisiert. Hierbei wurden Segmentverluste zwischen *2 %* bis *4 %* bei einem Jitter von *+/- 5 ms* angewandt. Diese Konstellation spiegelt eine vergleichbare Situation wie die Daten-Verbindung zwischen UDE und UKM wider. Die Korrelation von Segmentverlusten, die auf einen vorherigen Segmentverlust folgen, wurde auf *25 %* eingestellt. Abbildung 8.22 zeigt Boxplots des auf Empfangsseite resultierenden Jitters bei einem Segmentverlust von *2 %*.

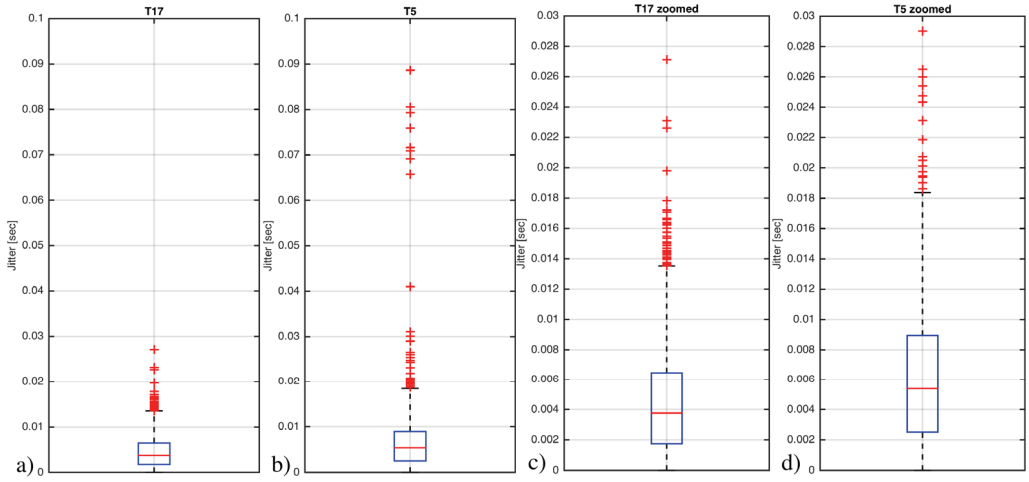

Abbildung 8.22: Jitter-Boxplot einer Übertragung unter Einfluss von +/- 5 ms Jitter bei einem Segmentverlust von 2 %, a) rMPTCP, b) TCP, c) rMPTCP vergrößert, d) TCP vergrößert

Deutlich zu sehen sind die vermehrt vorkommenden Ausreißer der TCP-Verbindung. Diese werden durch erneutes Senden von Datensegmentverlusten erzeugt. Sie konnten mithilfe von rMPTCP erfolgreich unterdrückt werden. Der Median unter rMPTCP wurde leicht erhöht.

Die nächste Abbildung 8.23 zeigt eine Verbindung unter Einfluss von *4 %* Segmentverlust. Hier konnten unter TCP massive Ausreißer beobachtet werden, die ebenfalls unter rMPTCP erfolgreich unterdrückt werden können.

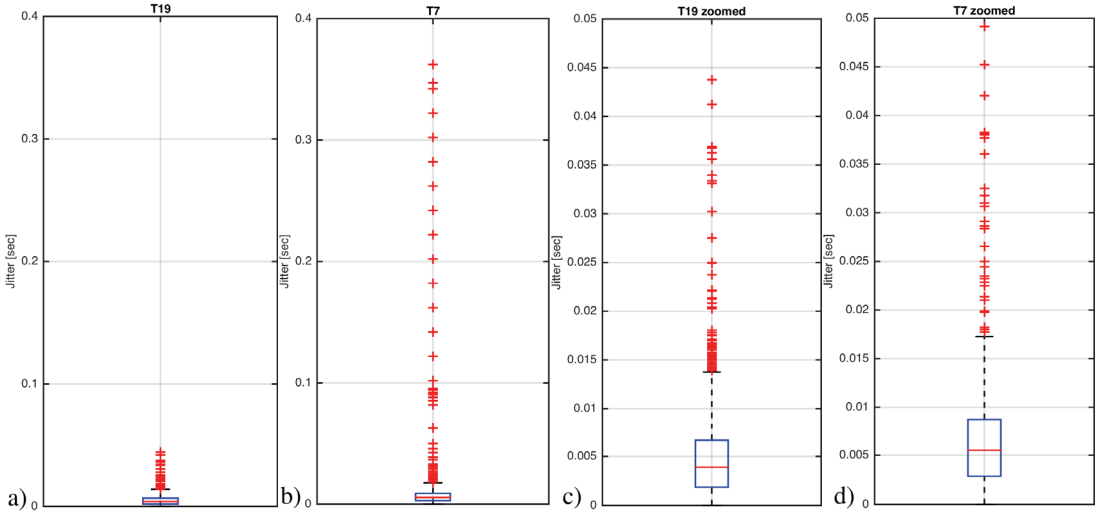

Abbildung 8.23: Jitter-Boxplot einer Übertragung unter Einfluss von +/- 5 ms Jitter bei einem Segmentverlust von 4 %, a) rMPTCP, b) TCP, c) rMPTCP vergrößert, d) TCP vergrößert

Unter Einfluss von nicht-korrelierten Segmentverlusten schwindet der Vorteil von rMPTCP unter höherwerdenden Verlusten. Bei extremen Bedingungen, wobei Segmentverluste keinem geordneten Muster folgen, werden die Subflows durch das in Kapitel 8.4 beschriebene Phänomen, dem Inter-Subflow-Jitter, beeinflusst. Obwohl eine allgemein kürzere Latenzzeit zwischen Sender und Empfänger zu beobachten ist, wird der Jitter durch ein Hin- und Herschwingen zwischen den Subflows erhöht. Dies ist unter extremen im Labor erzeugten Bedingungen zu beobachten, wenn alle Subflows nahezu homogenen, jedoch chaotischen Eigenschaften folgen – und damit keine Main-Subflows erkannt werden können. Abbildung 8.24 zeigt einen Fall, bei dem alle Subflows unter einem Jitter von *10 ms* und einem unkorrelierten Segmentverlust von *30 %* stehen.

Ein derartig hoher Segmentverlust deutet normalerweise den Verlust einer Verbindung an. In diesem Fall schaltet der Überlastalgorithmus von TCP, der zu einer starken Verlangsamung der betroffenen Verbindungen führt. Eine garantierte Redundanzquote ist in diesem Fall hinderlich und führt dazu, dass alle Subflows soweit gebremst werden, dass die Verbindung stark eingeschränkt wird. Eine rMPTCP-Verbindung sollte daher mit dem adaptiven Redundanzalgorithmus abgesichert werden, der die Redundanz absenkt und umverteilt. Der Algorithmus kann aber nur optimal funktionieren, wenn die Leitungsqualität einem Muster unterliegt und nicht in einem starken Maße auf

Zufall beruht, bei dem weder eine Verschlechterung noch eine Verbesserung der Leitungsqualität erkannt werden kann.

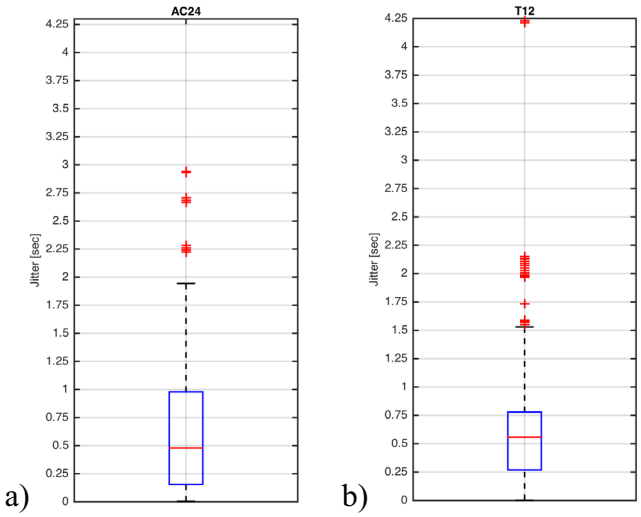

Abbildung 8.24: Jitter-Boxplot einer Übertragung unter Einfluss von +/- 10 ms Jitter bei einem Segmentverlust von 30 %: a) rMPTCP, b) TCP

Abbildung 8.24 zeigt eine Gegenüberstellung von a) rMPTCP gegenüber dem Verhalten von b) TCP unter extremen Bedingungen. Der Median von rMPTCP liegt zwar tiefer als bei TCP, jedoch besitzt der Boxplot eine größere Breite und es sind größere Ausreißer vorhanden. Die Überlastkontrolle ist im Fall von rMPTCP eher hinderlich, da sie es nicht zulässt, dass Segmente pünktlich versendet werden. Auf allen Subflows käme es zu einem gleichzeitigen Pausieren der Verbindung, damit alle vorher gepufferten Segmente nach der vorher eingesetzten Redundanzquote versendet werden können. Zeitliche Unterschiede zwischen den Subflows sind das Resultat, die zu einer Steigerung des Inter-Subflow-Jitters führen.

Der Inter-Subflow-Jitter führt durch das stetige Nutzen von einzelnen Segmenten auf jeweils unterschiedlichen Subflows zu einem hohen Jitter. Dennoch werden die Segmente unter rMPTCP allgemein etwas schneller versendet, wie der niedrigere Median zu erkennen gibt. Bei großen Latenzunterschieden bei den vorhandenen Verbindungen werden Segmente der schnelleren Verbindung bevorzugt.

8.10 Evaluation des rMPTCP Gateways

Ein Leistungstest der beschriebenen Applikation wurde mit Fokus auf die zusätzliche Latenz durchgeführt, die das Gateway benötigt, um die Daten weiterzuleiten. Abbildung 8.25 zeigt, an welcher Stelle eine Messung durchgeführt muss, um die Latenz der Applikation festzustellen.

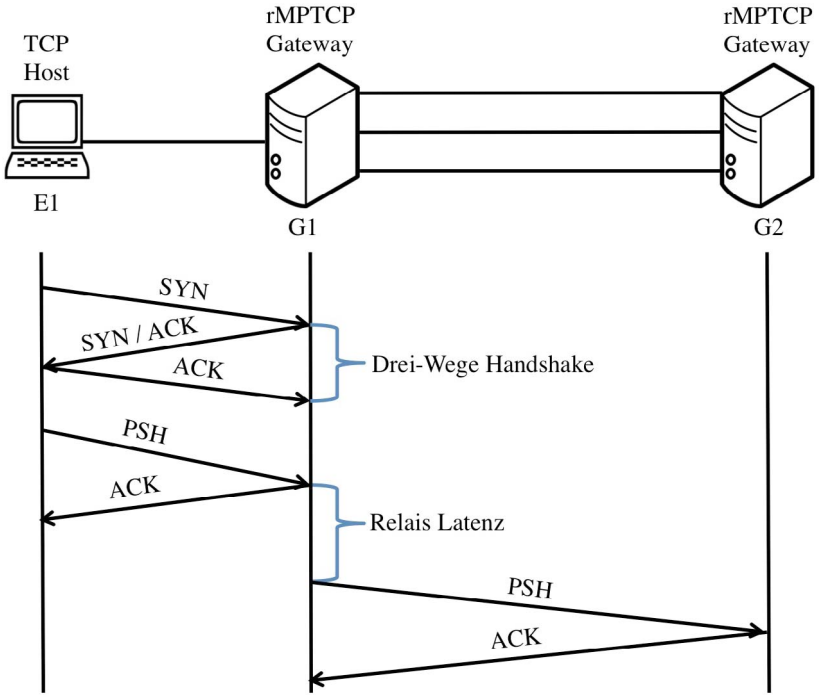

Abbildung 8.25: rMPTCP-Gateway-Latenz

Nach erfolgtem Drei-Wege-Handshake werden die eintreffenden Pakete weitergeleitet. Die Zeit vom Eintreffen des PSH-Pakets auf dem E1-zugewandten Netzwerkinterface bis zum Senden eines PSH-Pakets am G2-zugewandten Netzwerkinterface entsprach der Latenz des Gateways. Zur Messung des zeitlichen Abstands wurde das Programm *Wireshark* verwendet. Innerhalb einer einfachen Massendatensendung (Bulk Transfer) vom Rechner E1 zum Gateway-Rechner G1 wurden stichpunktartig 50 Paketsendungen mit verwandten PSH-Paketen der beiden Netzwerkinterfaces entnommen und ausgewertet.

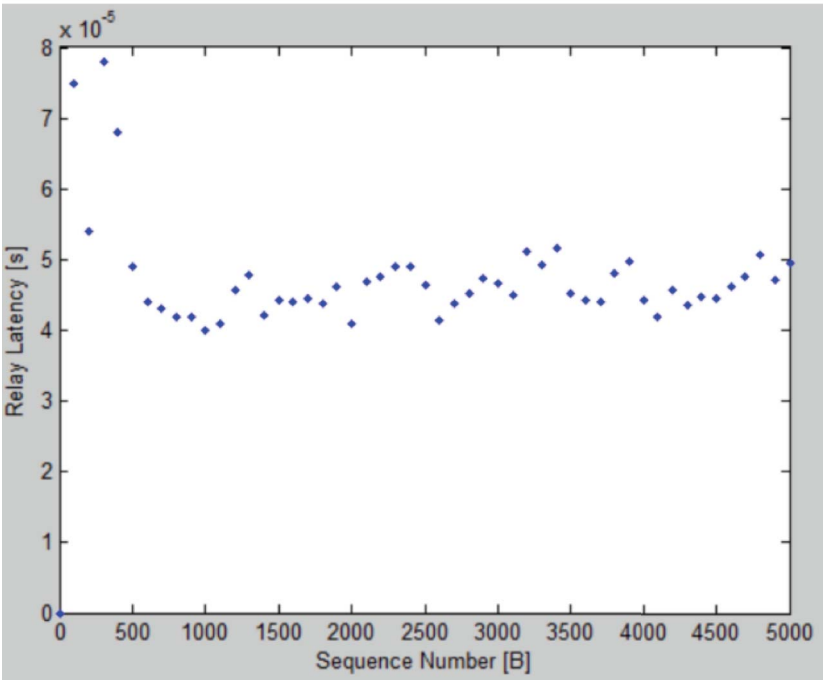

Abbildung 8.26: Relais-Latenz des rMPTCP-Gateways, [Zhang, 2016]

Die Messungen ergaben eine durchschnittliche Latenz von 51,24 µs. Abbildung 8.26 zeigt die Latenz von 50 Paketen in zeitlicher Abfolge.

Am Anfang ist eine deutliche Erhöhung der Werte zu erkennen, die sich im weiteren Verlauf der Übertragung stabilisiert. Wiederholungen der Versuche mit unterschiedlichen Sendungsarten der Daten führten zu ähnlichen Werten. Variationen in der Latenz lagen zwischen *40 µs* und *80 µs*. Unter Annahme, dass das zweite Gateway ähnliche Eigenschaften besitzt, kann von einer maximalen zusätzlichen Latenz von *160 µs* ausgegangen werden. Dies liegt oberhalb der in Kapitel 2.5.2 diskutierten Dienstgüte-Anforderungen in der Telemedizin.

9 Fazit und Ausblick

Im Rahmen dieser Arbeit wurde ein Scheduler für MPTCP entwickelt, der durch redundante Nutzung mehrerer Verbindungen Latenzzeiten minimiert und Datensegmentverluste ausgleicht. Die Entwicklung wurde mithilfe einer mathematischen Herangehensweise realisiert, die die Entwicklung eines neuen adaptiven Algorithmus auf Ende-zu-Ende-Ebene ermöglichte.

Die Arbeit entstand im Kontext einer Kooperation zwischen zwei Universitätsstandorten, die sich auf verschiedenen Kontinenten befinden. Um die begrifflichen Grundlagen zu schaffen, wurden Definitionen der Telemedizin inklusive ihrer Aspekte und Normen erörtert sowie verschiedene Dienstgüteklassen und die entsprechenden Dienstgüteparameter ausgearbeitet.

Die Grundlage von Dienstgüte im Internet folgt bestimmten Mechanismen, die im Rahmen dieser Arbeit betrachtet wurden. Die entsprechenden Funktionen, Protokolle und Eigenschaften, die im Internet die Basis für eine gleichberechtigte Behandlung des Datenverkehrs stellen, wurden dargestellt. Sie konnten als Ursachen für mehrere Probleme bei Anwendungen mit speziellen Dienstgüte-Anforderungen identifiziert werden.

Für eine Analyse der Verbindung zwischen UDE und UKM wurde ihr grundsätzlicher Aufbau beschrieben, Schwachstellen aufgedeckt sowie die Dienstgüteeigenschaften dreier Verbindungen zwischen den beiden Standorten bestimmt. Diese Analyse ergab, dass die Pfade zu einem gewissen Teil unabhängig voneinander sind. Ihre gleichzeitige Nutzung kann trotz zum Teil großer Unterschiede in der Dienstgüte zu einer Verbesserung der Verbindungsqualität führen.

Daran anknüpfend wurden aktuelle Techniken und Herangehensweisen untersucht, die zu einer Verbesserung der Dienstgüte führen können. Diese waren in erster Linie Mehrwegtechniken und Methoden der Redundanz zum Störungsausgleich. Verschiedene Beiträge wurden beschrieben und untersucht, wobei der aktuelle Stand der Technik dargelegt wurde.

Den Hauptaspekt dieser Arbeit bildete die Entwicklung des redundanten Mehrwegprotokolls rMPTCP mit dem Ziel, die Dienstgüte für die anvisierten Anwendungen über die Strecke zwischen den beiden Standorten zu verbessern. Mithilfe verschiedener Zusatzanwendungen wie einem Gateway, der Auswertung von Pfadeigenschaften und -zusammenstellung sowie einem Steuerungsprogramm, das die Nutzung von beliebigen Anwendungen und die Veränderungen der eingesetzten Parameter ermöglicht, wurde die Nutzung des Protokolls getestet. Hierdurch konnte eine Basis für weitere Untersuchungen und Verbesserungen geschaffen werden.

Bei der Auswertung der entwickelten Algorithmen wurde speziell auf die Funktion der Redundanzquote, der Erkennung von Störungen und der verwendeten Algorithmen eingegangen. Es erfolgte eine Evaluation des Nutzens hinsichtlich der Dienstgüteverbesserung durch Ausgleich von Datensegmentverlusten und Latenzvariationen.

9.1 Ergebnisse und wissenschaftlicher Beitrag

Es wurde ein adaptiver Algorithmus entwickelt, der in der Lage ist,

- eine bestimmte Redundanzquote zu garantieren, d.h. es kann in jedem Fall eine festgelegte Segment-Replikation auf verschiedenen Pfaden durchgeführt werden,
- redundante Datensegmente auf Subflows in Abhängigkeit von der Redundanzquote, der optimalen Ausnutzung aller Subflows und unter Rücksichtnahme auf Out-of-Order-Vermeidung zu verteilen,
- die Redundanzquote adaptiv den gegebenen Umständen einer Verringerung der Datenübertragungsrate anzupassen,
- von Störungen oder Ausfall betroffene Pfade mithilfe eines Pfad-Managements auszugleichen, sodass trotz festgelegter Redundanzquote ein Halten der geforderten Datenübertragungsrate möglich ist.

Im Zuge der Entwicklung des adaptiven Algorithmus wurde die mathematische Basis realisiert, um die Abhängigkeiten zwischen den folgenden Einflussfaktoren zu untersuchen:

- Datensegmentverluste
- Anzahl der verfügbaren Pfade
- Redundanz-Größe (Quote)
- nutzbare Datenübertragungsrate

Die mathematischen Zusammenhänge wurden bei der Entwicklung der Algorithmen benutzt, um eine bessere Verteilung von redundant gesendeten Datensegmenten zu ermöglichen.

rMPTCP bietet eine Handhabe, das Dienstgüte-Niveau für kritische Anwendungen über das Internet zu verbessern. Bestehende Techniken leiden vor allem an der Korrelation zwischen Fehlern auf einem einzelnen Pfad. Mehrwegtechniken mit ähnlicher Herangehensweise sind meist auf den unteren ISO/OSI-Schichten implementiert, sodass sie sich nur für kleinere Netzwerke oder für die Nutzung durch NSPs eignen.

Die nachfolgende Abbildung 9.1 zeigt ein Schema des in dieser Arbeit entwickelten Systems. Zu sehen sind mehrere Bereiche, die in dieser Arbeit schwerpunktmäßig behandelt wurden. Die größte Aufmerksamkeit lag auf dem rMPTCP-Scheduler (1), der die Daten repliziert und auf die vorhandenen TCP-Verbindungen verteilt. Unter Nutzung fortlaufender Messungen auf den Verbindungen konnten Algorithmen-Strategien (2) realisiert werden, die im Falle einer Veränderung im Netzwerk die Parameter anpassen. Die Messungen wurden mithilfe einer Monitoringfunktion (4) realisiert, die eben-

falls durch die API (3) ausgelesen werden können. Mit der API ist es außerdem möglich, die in rMPTCP implementierten Zustände und Parameter zu kontrollieren. Auf Seite des Empfängers wurde ein vereinfachtes Buffering (5) in Betracht gezogen, um den Inter-Subflow-Jitter zu kompensieren. Die Bereiche der Überlastkontrolle (CC) unter TCP (6) und eines Pfadmanagements für unabhängige Pfade (7) wurden angeschnitten und können in weiteren Forschungsarbeiten in Betracht gezogen werden.

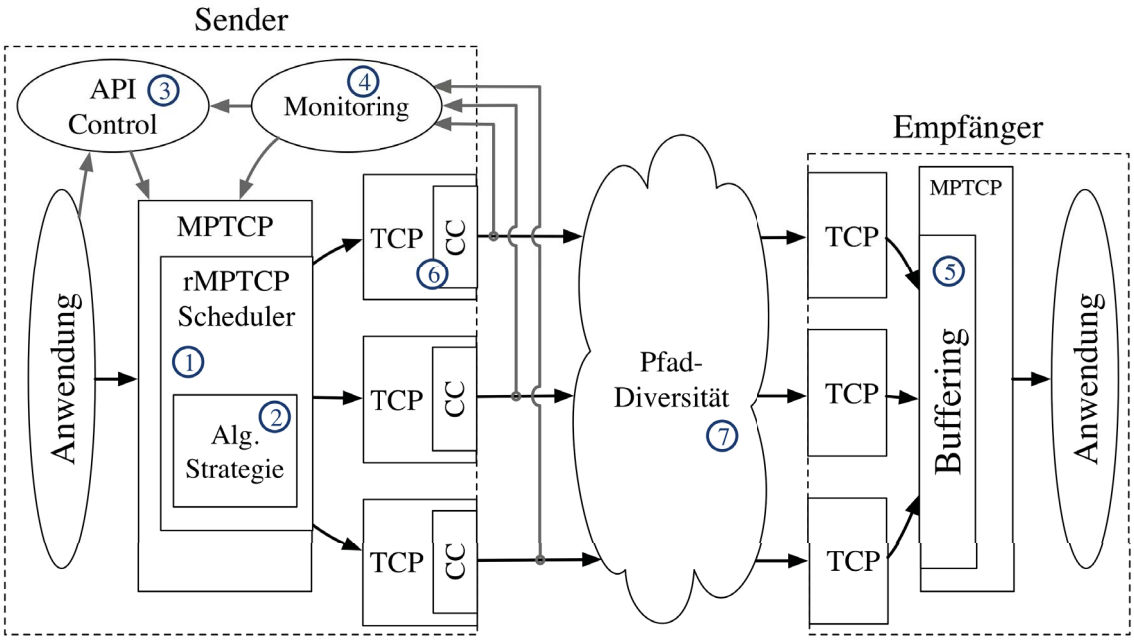

Abbildung 9.1: Schematische Darstellung von rMPTCP

Die Evaluation hat gezeigt, dass es möglich ist, eine bestmögliche Verteilung von redundant gesendeten Datensegmenten auf verschiedene Pfade vorzunehmen. Dies gelingt durch eine adaptive Verteilung der Datenübertragungsrate, die bei Minderung der Pfadqualität einen Ausgleich schaffen kann. Probleme von Out-of-Order-Datensegmenten, die bei einer beliebigen Verteilung der verschiedenen Datensegmente auf heterogene Pfade entstehen, werden mithilfe einer Priorisierung ausgeglichen. Die Evaluation ergab, dass die mathematischen Zusammenhänge zutreffen und ein nach diesem Schema entwickelter Algorithmus funktioniert.

Für die Anpassung an Pfadeigenschaften wurde ein Automat entworfen, der verschiedene Zustände nutzt, um Maßnahmen hinsichtlich einer Behandlung der Pfade zu treffen. Es wurden zwei verschiedene Erfassungsmethoden entwickelt, die eine relativ schnelle Erkennung von Störungen auf dem Pfad erlauben. Die Evaluation zeigt, dass diese für die getesteten Anwendungen und Fälle funktionieren, aber mit kleineren Störungen und kurzzeitigen Latenzerhöhungen einhergehen. Eine Feinjustierung der Parameter ist hier in Abhängigkeit der Anwendung sinnvoll. Durch die Natur der Sendemuster von verschiedenen Anwendungen benötigen Anwendungen mit davon abweichenden Sendeprofilen möglicherweise unterschiedliche Parameterjustierungen.

Eine Zusicherung der Dienstgüte auf Ende-zu-Ende-Ebene ist grundsätzlich nicht möglich, da Dienstgütebereitstellung vor allem auf Vermittlungsebene stattfindet und

damit nicht im Zugriff des Benutzers oder der Applikation liegt. Dies gilt für rMPTCP ebenso wie für andere Techniken auf Ende-zu-Ende-Ebene. Die hier entwickelte Technik erlaubt allerdings eine Verbesserung der Eigenschaften innerhalb eines Best-Effort-Ansatzes, die eine Qualitätssteigerung für bestimmte Anwendungen bewirkt. Durch Kombinationen mit darunterliegenden Vermittlungstechniken wie MPLS oder explizit vermittelten Datenverbindungen kann die Qualität weiter gesteigert werden. Dies gilt vor allem für die Zusicherung von unabhängigen Pfaden.

Als Problem hat sich der Inter-Subflow-Jitter bei verlustbehafteten Verbindungen herausgestellt, der bei gleichzeitig zunehmenden Störungen auf allen Subflows zu erhöhtem Jitter führt. Eine Grenze konnte bei Segmentverlusten von *4* bis *5 %* identifiziert werden. Höhere Verluste auf allen Subflows führen zu einem erhöhten Jitter. Trotzdem werden extreme Latenzspitzen weiterhin ausgeglichen und liegen sogar in Extremfällen von *30 %* Segmentverlust im Allgemeinen niedriger als bei TCP. Das Spektrum des Jitters ist jedoch breiter. Bleiben Segmentverluste unterhalb der Grenze von 5 %, so werden selbst starke Verzögerungsvarianzen ausgeglichen. Maßnahmen, diese Grenze zu erhöhen, betreffen eine alternative Überlastkontrolle in TCP, sowie ein Dejittering zwischen den Subflows auf Empfängerseite. Anwendungsspezifische Tests könnten ebenfalls zu besseren Ergebnissen führen.

Grundsätzlich bietet rMPTCP damit auf der Strecke UDE-UKM für verschiedene telemedizinische Anwendungstypen eine Verbesserung der Dienstgüte, um unter die in Tabelle 2.10 und Tabelle 2.11 beschriebenen Anforderungen zu fallen sowie das Übertragungsverhalten einer Anwendung über die Strecke zu stabilisieren. Die Anwendungstypen betreffen alle verbindungsorientierten Übertragungsmechanismen. Dazu gehören:

- Textkommunikation
- unkritische Maschinensteuerung bzw. ARTP-Steuerung
- Senden von Sensor- und Telemetriedaten
- Durchführung von Desktopkonferenzen
- Übertragungen von Massendaten mit weichen Echtzeitanforderungen

Für die Steuerung von Maschinen in hochkritischen Szenarien ist die Stabilisierung durch drei genutzte Pfade jedoch unzureichend, da nach wie vor die Wahrscheinlichkeit eines kurzzeitigen Anstiegs der Verzögerung auf allen Pfaden gegeben ist. Für die Nutzung in telemedizinischen Szenarien mit weniger kritischen Anforderungen, Szenarien der „weichen Echtzeit“ oder in Notsituationen kann durch die entwickelte Technik ein Nutzen erlangt werden. Dies betrifft die folgenden Anwendungsszenarien:

- Telekonsultation
- Telediagnostik
- Teletherapie
- Telemonitoring
- Notfalltelemedizin
- sowie ARTP-gestützter Teleausbildung und Tele-Operation.

Die hier entwickelte Technik gestattet mithilfe der implementierten Monitoring-funktionen ebenfalls eine Beobachtung des Verbindungszustands, der bei Abfall in kritische bzw. ungesicherte Zustände ein Eingreifen seitens des Benutzers ermöglicht. Für viele Anwendungen genügt diese Steigerung der Qualität insofern, dass sie den Einsatz in weniger kritischen telemedizinischen Situationen verbessert: Im Anwendungsfall des Telepointers führt dies zu einer verbesserten Kommunikation, einer Vermeidung von Missverständnissen und einer zielgerichteteren Arbeitsweise.

Gegeben durch zusätzliche Redundanz verwendet rMPTCP eine erhöhte Datenübertragungsrate, die vom Prinzip der Gleichberechtigung im Netz abweicht, indem mehr Daten versendet werden, als eigentlich für eine Anwendung vorgesehen sind. Es wird zusätzlicher Datenverkehr erzeugt, der zu weiterem Datenaufkommen innerhalb der Netzwerke führt. Durch den Erwerb zusätzlicher Bandbreite ist dies mit erhöhten Kosten verbunden, was den durch die vermeintlich ungerechte Nutzung der NSP-Ressourcen gegebenen Vorteil wieder ausgleicht.

Unter Einsatz mehrerer rMPTCP-Gateways kann eine Ende-zu-Ende-Infrastruktur aufgebaut werden, die mehrere Geräte ohne eigene rMPTCP-Fähigkeiten über kritische Entfernungen miteinander verbindet. Das Gateway stellt eine optimale Ergänzung für den in dieser Arbeit beschriebenen Einsatzfall dar.

9.2 Weiterführende Arbeiten

Die in dieser Arbeit entwickelte Technik liefert eine ausbaufähige Basis für weitere Forschungsarbeiten, um die Dienstgüte mit der dargelegten Technik weiter zu steigern. Hieraus entstanden weiterführende Ideen und Ansätze, die das Potenzial besitzen, in Kombination mit dem entwickelten Protokoll rMPTCP zu weiteren Verbesserungen führen. Diese umfassen die folgenden Themen:

- tiefergehende Untersuchungen der veränderbaren Parameter
- weitere Methoden zur Störungserkennung
- höher entwickeltes Empfangsfilter
- Untersuchungen anderer Überlastkontrollmechanismen unter TCP
- Anwendung im realen Einsatz
- Pfad-Priorisierung bezüglich Pfad-Diversität
- Vorhersehbarkeit der Dienstgüte auf der Strecke UDE-UKM

Tiefergehende Untersuchungen der veränderbaren Parameter

Die Evaluation von rMPTCP hat ergeben, dass eine Abhängigkeit zwischen den Scheduler-eigenen Parametern und dem Sendeverhalten der genutzten Anwendung besteht. Änderungen der Parameter können sich auf das Verhalten des Schedulers auswirken. Diese Parameter belaufen sich auf

- Mittelwertberechnungen,
- die Einstellungen für die Störungserkennung,
- die Timeouts beim Senden.

Für die getesteten Fälle konnten Einstellungen gefunden werden, die eine hinreichende Leistung von rMPTCP erlauben. Weitere Untersuchungen sind sinnvoll, in welchen das Sendeverhalten anderer Anwendungen genauer untersucht wird. Hierbei könnten Einstellungen gefunden werden, die ebenfalls einen Nutzen für extremere Fälle darstellen. Es wäre denkbar, einen Automaten zu entwickeln, der bestimmte Parametervorgaben einstellt, die eine Adaption auf das Sendeverhalten in Hinblick auf die Situation der Subflows (Anzahl, Datenübertragungsrate, RTT) vornimmt. Verschiedene Parametervorgaben könnten so automatisch eingesetzt werden.

Weitere Methoden zur Störungserkennung

Einen Arbeitsschwerpunkt von rMPTCP bildet der Bereich der Störungserkennung. Es wurden zwei Methoden eingesetzt, die eine Störungserkennung implementieren. Ergänzende Methoden zur Verbesserung sind denkbar, wie z.B. die Erfassung der Schwankungen der RTT. Diese wird kontinuierlich mithilfe der beim Sender empfangenen ACK-Segmente überwacht und kann dazu beitragen, eine Verschlechterung der Qualität auf einem Subflow zu erkennen. Vergrößert sich der zeitliche Abstand zwischen den gesendeten Datensegmenten und den zurückkommenden ACK-Segmenten, so bedeutet dies eine Verschlechterung der Dienstgüte. Weitere Methoden sind Bestandteil aktueller Forschungen, von deren Ergebnissen rMPTCP profitieren kann.

Höher entwickeltes Empfangsfilter

Die Analyse des rMPTCP-Protokolls hat ergeben, dass stark homogene Pfade zu einer Verschlechterung des Gesamt-Jitters beitragen können. Ein erratisches Verhalten wird durch das Hin- und Herspringen zwischen den Subflows hervorgerufen. In der Evaluation wurde zu diesem Zweck ein einfacher Tiefpassfilter eingesetzt, der periodisch gesendete Segmente angleicht. Höher entwickelte Empfangsfilter sind besser in der Lage, sich an weniger periodisches Sendeverhalten anzupassen. Diese Filterarten weiter zu untersuchen kann einen weiteren Forschungsschwerpunkt zukünftiger Forschungsarbeiten bilden.

Ein vorgeschaltetes Filter erzeugt zusätzlich Latenzzeiten durch die Vorpufferung. Als Erweiterung zum rMPTCP-Protokoll wäre daher eine mögliche Verwendung der Zeitstempelfunktion im TCP-Header sinnvoll. Mithilfe dieser Funktion könnte ein vorgeschaltetes adaptives Filter dazu beitragen, mit weniger zusätzlichen Latenzzeiten eine zeitgerechtere Zustellung des Segments an die Applikation zu gewährleisten.

Untersuchungen anderer Überlastkontrollmechanismen unter TCP

rMPTCP verwendet standardmäßig Überlastkontrollmechanismen von TCP. Diese wurden in Hinblick auf eine optimierte Übertragung zur maximalen Ausnutzung der möglichen Datenübertragungsrate entwickelt. Für rMPTCP kann diese Funktion jedoch hinderlich für bestimmte Anwendungen sein, da sie unter extremen Bedingungen dazu führt, dass die Verbindungen auf allen Subflows stark verlangsamt werden. Da unter rMPTCP mehrere Subflows zur Verfügung stehen, können abgesendete Segmente von

einer höhergelegenen Kontrollinstanz profitieren, die eine gleichzeitige Pufferung der zu sendenden Segmente auf allen Subflows verhindert. Hierfür ist eine Strategie vorteilhaft, die im Falle extremer Störungen Subflow-übergreifend eine Pufferung koordiniert. Für andere Überlastkontrollmechanismen, die für TCP entwickelt wurden, ist zu überprüfen, ob deren Optimierung ein verbessertes Verhalten von rMPTCP in diesen Situationen hervorruft.

Anwendung im realen Einsatz

Die vorgenommene Evaluierung von rMPTCP wurde unter Laborbedingungen durchgeführt, die die vorher vermessene Verbindung zwischen UDE und UKM widerspiegeln sollte. Hierbei wurden die Funktionen unter annähernd denselben Eigenschaften und Bedingungen wie in der real existierenden Verbindung sowie für davon abweichende extremere Fälle untersucht. Um noch exaktere Ergebnisse zu erlangen und damit die Nutzung weiter zu verbessern, sollten der Einsatz und die Feinjustierung des Protokolls noch im realen Einsatz durchgeführt werden. Eine weitere Untersuchung und Optimierung des entwickelten Protokolls sollte daher im Kontext der tatsächlichen Verbindung zwischen UDE und UKM stattfinden.

Pfad-Priorisierung bezüglich Pfad-Diversität

Das Sicherstellen von voneinander unabhängigen Pfaden ist eine Anforderung, die nur auf der Vermittlungsebene voll erfüllt werden kann. Eine Identifikation möglichst unabhängiger Pfade ist aber von Vorteil, um die gemeinsame Abhängigkeit von Fehlerquellen zu minimieren. Das in dieser Arbeit entwickelte Protokoll erlaubt verschiedene Pfadzusammenstellungen, die ebenfalls auf ihre Unabhängigkeit hin untersucht werden können. Eine Option ist hier die Nutzung von Formel 6.10 und entsprechender Modifikation für mehr als zwei Pfade, um verschiedene Pfadkombinationen bewerten zu können. Gemeinsame Knotenpunkte können mit Abtastung der verwendeten Routen durch Sondierung mit zum Beispiel dem ICMP-Protokoll identifiziert werden. Ein darauf spezialisiertes Pfad-Management erlaubt mithilfe der hierdurch gewonnen Daten eine automatisierte Priorisierung von bestimmten Pfaden. Dies wird durch das Einbinden der Routenabtastung in die Pfadauswahl und Pfadbeurteilung erreicht. Hierdurch kann ebenfalls eine Abschätzung der Zuverlässigkeit der Pfade und eine Beurteilung der gemeinsamen Fehlerwahrscheinlichkeiten verschiedener Subflow-Kombinationen ermöglicht werden.

Vorhersehbarkeit der Dienstgüte auf der Strecke UDE-UKM

Die Vorhersehbarkeit von Netzwerken ist ein Forschungsbereich, der seit Anbeginn der Entwicklung der Netzwerktechniken im Mittelpunkt der Aufmerksamkeit steht. Die Verbindung zwischen UDE und UKM weist auf der Strecke der Übertragung verschiedene Dienstgütezustände auf. Die Messungen erbrachten eine hinreichende Datensammlung der netzwerkeigenen Schwankungen. Sie bieten Ansätze für Erweiterungen des bereits im Rahmen dieser Arbeit implementierten Netzwerkmonitorings. So könnte eine

Markov-Modellierung zu einem Automaten führen, der verschiedene Verbesserungen hinsichtlich der adaptiven Funktionen und der Vorhersehbarkeit der Dienstgüte auf der Strecke UDE-UKM bewirkt.[48]

Abschließend kann zusammengefasst werden, dass eine Optimierung von Übertragungsprotokollen zur Verbesserung der Dienstgüte in telemedizinischen Echtzeitanwendungen trotz eines festgelegten Best-Effort-Regelwerks im Internet auf Ende-zu-Ende Ebene realisierbar ist. Es gibt jedoch noch oben genannte Bereiche, die weiterer Forschung bedürfen und weitere Vorteile hervorbringen würden.

[48] Eine weitere Ausführung dieser Methode befindet sich im Anhang unter 10.7.

Literaturverzeichnis

Adhari, H., Dreibholz, T., Becke, M. & Rathgeb, E.P., 2011. „Evaluation of Concurrent Multipath Transfer over Dissimilar Paths," *Conference on Advanced Information Networking and Applications (WAINA), Mar 22-25 Biopolis*, März, S. 708-714.

Aggarwal, H., 2012. „Managing Quality of Service in Computer Networks," *International Journal of Scientific & Engineering Research*, November, Band 3, Ausg. 11, S. 1-5.

Aidarous, S. & Plevyak, T., 2003. *Managing IP Networks – Challenges and Opportunities*, USA, IEEE Press, Wiley-Interscience.

Akella, A., Maggs, B., Seshan, S. & Anees, S., 2003. „A Measurement-Based Analysis of Multihoming," *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, S. 353-364.

Albrecht, M., Etgeton, S. & Ochmann, R.R., 2015. „Regionale Verteilung von Arztsitzen (Ärztedichte)," *Faktencheck Gesundheit*.

Allman, M., Paxson, V. & Blanton, E., 9/2009. *RFC 5681, TCP Congestion Control*, Internet Engineering Task Force.

Anästh Intensivmed (Hg.), 2016. [Online] Verfügbar unter: https://www.bda.de/files/Mrz_2016_-_Aus_der_Kommission_Telemedizin.pdf, [23.01.2017].

Ao, W.C., Chen, P.Y. & Chen, K.C., 2012. „Rate–Reliability–Delay Tradeoff of Multipath Transmission Using Network Coding," *IEEE Transactions on Vehicular Technology*, Juni, Band 61, Ausg. 5, S. 2336-2342.

Apache, *Software Foundation*, [Online], 2016, Verfügbar unter: http://www.apache.org/, [28.03.2016].

Apostolopoulos, J.G. & Trott, M.D., 2004. „Path Diversity for Enhanced Media Streaming," *IEEE Communications Magazine archive*, August, Band 42, Ausg. 8, S. 80-87.

Aubry, F., Lebrun, D., Deville, Y. & Bonaventure, O., 2015. „Traffic duplication through segmentable disjoint paths," *in Proceedings ALDB15, IIFIP Networking Conference (IFIP Networking), 20-22 May*, Mai.

Baidya, S.H. & Prakash, R., 2014. „Improving the performance of multipath TCP over heterogeneous paths using slow path adaptation," *2014 IEEE International Conference on Communications (ICC)*, S. 3222-3227.

Bannon, L.J., 2000. „Understanding Common Information Spaces in CSCW (Draft)," *Position paper for Workshop on Common Information Spaces, Copenhagen, August 23-25*.

Barré, S., 2011. *Implementation and assessment of Modern Host-based Multipath Solutions, PhD thesis*, Louvain, Belgien.

Barré, S., Paasch, C. & Bonaventure, O., 2011. „MultiPath TCP: From Theory to Practice," *10th International IFIP TC 6 Networking Conference, May 9-13*, S. 444-457.

Barré, S., Ronan, J. & Bonaventure, O., 2011. „Implementation and evaluation of the Shim6 protocol in the Linux Kernel," *Computer Communications, Elsevier B.V.*, September, Band 34, Ausg. 14, S. 1685-1695.

Bashshur, R.L., Shannon, G.W. & Reardon, T.G., 2000. „Telemedicine: A New Health Care Delivery System," *Annu. Rev. Public Health*, Band 21, S. 613–637.

Batsomboon, P. & Tosunoglu, S., 1996. „A Review of Teleoperation and Telesensation System," *1996 Florida Conference on Recent Advanced in Robotics, Florida Atlantic University, April 11-12*.

Berliner, B., Clark, B. & Hartono, A., o.D. „QoS Requirements of Multimedia Applications," *Department of Computer Science and Engineering, The Ohio State University*, S. 1-10.

Bettermann, S. & Rong, Y., 2011. „Effects of Fully Redundant Dispersity Routing on VoIP Quality," *IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, Mai, S. 1-6.

Bird, K., 1971. „Teleconsultation: A new Health Information Exchange System," *zitiert von Rashid L. Bashshur, Timothy G. Reardon, Gary W., Shannon, „Telemedicine: A New Health Care Delivery System," Annu. Rev. Public Health, 2000, Vol. 21, pp. 613–637*.

Blake, S. et al., 12/1998. *RFC 2475: An Architecture for Differentiated Services*, Internet Engineering Task Force.

Bolot, J.-C., 1993. „Characterizing End- to-End packet delay and loss in the Internet," *Journal of High Speed Networks*, Band 2, Ausg. 3, S. 305-323.

Bolot, J.-C., 1993. „End-to-End Packet Delay and Loss Behaviour in the Internet," *SIGCOMM'93*.

Bolot, J.-C., Fosse-Parisis, S. & Towsley, D., 1999. „Adaptive FEC-based error control for Internet telephony," *Proceedings of IEEE INFOCOM '99*, März, Band 3, S. 1453–1460.

Bolot, J.-C. & Garcia, A.V., 1996. „Control mechanisms for packet audio in the Internet," *INFOCOM'96 Proceedings of the Fifteenth annual joint conference of the IEEE computer and communications societies conference on The conference on computer communications*, März, Band 1, S. 232-239.

Bossert, M. & Breitbach, M., 1999. *Digitale Netze, Funktionsgruppen digitaler Netze und Systembeispiele*, Leipzig, BRD, B.G. Teubner Stuttgart.

Boyce, J.M. & Gaglianello, R.D., 1998. „Packet loss effects on MPEG video sent over the public Internet," *Proceedings of ACM MULTIMEDIA '98*, September, S. 181–190.

Brévart, C. et al., 2011. „Severe Vertex Epidural Hematoma in a Child: A Case Report of a Management without Expert Neurosurgical Care," *Case Reports in Surgery*, Band 2011, S. 1-3.

Braden, R., 10/1989. *RFC 1122, Requirements for Internet Hosts – Communication Layers*, Internet Engineering Task Force.

Braden, R. et al., 09/1997. *RFC 2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, Internet Engineering Task Force.

Brauns, H.-J. & Loos, W., 2015. „Telemedizin in Deutschland, Stand – Hemmnisse – Perspektiven," *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz*, September, Band 10, S. 1068-1073.

Briscoe, B. et al., 2014. „Reducing Internet Latency: A Survey of Techniques and their Merits," *IEEE Communications Surveys & Tutorials* , 26. November, Band 18, Ausg. 3, S. 2149-2196.

Brockhaus Enzyklopädie Online, 2017. [Online] Verfügbar unter: https://duisburg-essen-ub.brockhaus.de.

Bundesgesundheitsministerium (Hg.), 2016. [Online], Bundesgesundheitsministerium für Gesundheit, Verfügbar unter: https://www.bundesgesundheitsministerium.de/ministerium/meldungen/2016/big-data-anwendungen.html#c8154, [28.12.2016].

Bundesgesundheitsministerium (Hg.), *Glossar des Bundesministerium für Gesundheit, Telemedizin*, [Online], 2016, Verfügbar unter: www.bundesgesundheitsministerium.de/glossarbegriffe/t-u/telemedizin.html, [17.01.2016].

Bunz, M., 2009. *Vom Speicher zum Verteiler: Die Geschichte des Internet*, 2. Ausg., Berlin, BRD, Kulturverlag Kadmos.

Burns, A. & Wellings, A., 2001. *Real-time Systems and Programming Languages*, 4. Ausg., Addison-Wesley.

Callon, R., 12/1990. *RFC 1195: Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*, Internet Engineering Task Force.

Cerf, V.G. & Kahn, R.E., 1974. „A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications*, Mai, Band 22, Ausg. 5, S. 637-648.

Chen, Y., Farley, T. & Nong, Y., 2004. „QoS Requirements of Network Applications on the Internet," *IOS Press, Information, Knowledge, Systems Management*, Ausg. 4, S. 55-76.

Croteau, A.-M. & Vieru, D., 2002. „Telemedicine Adoption by Different Groups of Physicians," *International Conference on System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii*, Januar.

Cui, J. et al., 2003. „A Review of Teleoperation System Control," *Proceedings of the Florida Conference on Recent Advances in Robotics, FCRAR, May 8-9*, S. 1-12.

Davie, B. et al., 03/2002. *RFC 3246: An Expedited Forwarding PHB (Per-Hop Behavior)*, Internet Engineering Task Force.

Davie, B. & Rekhter, Y., 2000. *MPLS – Technology and Applications*, San Diego, CA, USA, Academic Press.

Davis, J.G., 1974. Contract NAS 9-13118 *Final report: Video Requirements for Remote Medical Diagnosis*, National Aeronautics and Space Adminsitration, Houston, Texas, USA.

DE-CIX, *DE-CIX Internetservice Node*, [Online], 2017, Verfügbar unter: https://www.de-cix.net, [08.01.2017].

Deering, S. & Hinden, R., 12/1998. *RFC 2460: Internet Protocol, Version 6 (IPv6)*, Internet Engieering Task Force.

Delrobaee, A. et al., 2006. „Design and Development of a Remote Medical Consultation System," *Acta Medica Iranica*, Band 44, Ausg. 1, S. 49-52.

Deter, G. & Markovski, G., 2011. „Aktueller Begriff: Telemedizin," *Wissenschaftliche Dienste, Deutscher Bundestag (Hg.)*, 11. Mai, Band 15, Ausg. 11.

Deutsches Institut für Normung (Hg.), 2006. *Medizinische elektrische Geräte; Teil 1: Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale*, DIN EN 60601-1; VDE 0750-1.

Deutsches Institut für Normung (Hg.), 2010. *Anwendung des Risikomanagements für IT-Netzwerke, die Medizinprodukte beinhalten - Teil 1: Aufgaben, Verantwortlichkeiten und Aktivitäten*, DIN EN 80001-1; VDE 0756-1:2011-11.

DFN, 2006. [Online] Verfügbar unter: https://www.dfn.de/fileadmin/5Presse/DFNMitteilungen/heft70.pdf, [28.2.2017].

DFN, *Deutsches Forschungsnetz Mitteilung Heft 89*, [Online], 2016, Verfügbar unter: https://www.dfn.de/fileadmin/5Presse/DFNMitteilungen/DFN_Mitteilungen_89.pdf, [08.01.2017].

DFN, *Weathermap des Deutschen Forschungsnetzes*, [Online], 2016, Verfügbar unter: http://www.win-labor.dfn.de/cgi-bin/hades/map.pl?config=win), [12.07.2016].

DFN, *Deutsches Forschungsnetz*, [Online], 2017, Verfügbar unter: http://www.dfn.de, [08.01.2017].

Dijkstra, E.W., 1959. „A note on two problems in connexion with graphs," *Numerische Mathematik 1*, S. 269–271.

Dittmar, R., Wohlgemuth, W.A. & Nagel, E., 2009. „Potenziale und Barrieren der Telemedizin in der Regelversorgung," *GGW*, November, Band 9, Ausg. 4, S. 16-26.

Dreibholz, T., Rathgeb, E.P., Rüngeler, I. & Seggel, R., 2011. „Stream Control Transmission Protocol – Past, Current, and Future Standardization Activities," *IEEE Communications Magazine*, April, Band 49, Ausg. 4, S. 82-88.

Duftschmid, G., Wrba, T., Dorda, W. & Pehamberger, H., o.D. [Online] Verfügbar unter: http://www.who.int/goe/policies/countries/aut_support_tele1.pdf.

Dugas, M. & Schmidt, K., 2003. *Medizinische Informatik und Bioinformatik*, Berlin Heidelberg, BRD, Springer-Verlag.

Dyck, J., Gutwin, C., Subramanian, S. & Fedak, C., 2004. „High-Performance Telepointers," *CSCW'04, Proceedings of the 2004 ACM conference on Computer supported cooperative work, November 6–10*.

Edwards, J., 2011. „Telepresence: Virtual Reality in the Real World," *IEEE Signal Processing Magazine*, November, Band 28, Ausg. 6, S. 9-12, 142.

Ellis, C.A., Gibbs, S.J. & Rein, G.L., 1991. „Groupware, some Issues and Experiences," *Communications of the ACM*, Januar, Band 34, Ausg. 1.

Eren, A.M., Peeters, W. & Farrow, J., 2007. „The Use of Space Technologies to Monitor and Respond to Earthquakes Economic Perspective," *3rd International Conference on Recent Advances in Space Technologies, 2007. RAST '07.*.

Ernst, A., 2015. „Funk-Übersicht, WLAN-Wissen für Gerätewahl und Fehlerbeseitigung," *c't*, Ausg. 15, S. 178–181.

Europäische Kommission, 1994. *Building the Information Society: The Telematics Applications Programme (1994 – 1998)*, Brüssel: European Commission DG XIII C/E.

Evensen, K. et al., 2009. „A network-layer proxy for bandwidth aggregation and reduction of IP packet reordering," *IEEE 34th Conference on Local Computer Networks, 2009. LCN 2009, 20-23 Oct.* , S. 585-592.

Fall, K.R. & Stevens, W.R., 2012. *TCP/IP Illustrated, Volume 1, The Protocols*, 2. Ausg., Michigan, USA, Pearson Education Inc.

Feussner, H., Etter, M. & Siewert, J.R., 1998. „Telekonsultation," *Der Chirurg*, Ausg. 69, S. 1129-1133.

Fiore, M., Casetti, C. & Galante, G., 2007. „Concurrent multipath communication for real-time traffic," *IEEE Computer Communications*, November, Band 30, Ausg. 17, S. 3307-3320.

Fischer, H. & Voges, U., 2011. „Medizinische Robotersysteme," In R. Kramme (Hg.) *Medizintechnik*, Berlin Heidelberg, Springer-Verlag. S.915-26.

Flach, T., Dukkipati, N., Terzis, A. & Raghavan, B., 2013. „Reducing web latency: the virtue of gentle aggression," *Proceedings of ACM SIGCOMM '13, August 12-16*, S. 159-170.

Fluckiger, F., 1995. *Understanding Network Multimedia, Applications and Technology*, Hertfordshire, UK, Prentice Hall.

Ford, L.R., 1956. *Network flow theory, Paper P-923*, Santa Monica, USA: The Rand Corporation,.

Ford, M., 2014. „Workshop Report: Reducing Internet Latency," *ACM SIGCOMM Computer Communication Review*, April, Band 44, Ausg. 2, S. 80-86.

Ford, A. et al., 03/2011. *RFC 6182, Architectural Guidelines for Multipath TCP Development*, Internet Engineering Task Force.

Ford, A., Raiciu, C., Handley, M. & Bonaventure, O., 2013. *RFC 6824: TCP Extensions for Multipath Operation with Multiple Addresses*, Internet Engineering Task Force.

Fortz, B., Rexford, J. & Thorup, M., 2002. „Traffic Engineering With Traditional IP Routing Protocols," *IEEE Communications Magazine*, Oktober, Band 40, Ausg. 10, S. 118 - 124.

Frömmgen, A. & Erbshäußer, T., *ReMPTCP Implementation*, [Online], 2015, Verfügbar unter: https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.91/net/mptcp/mptcp_redundant.c., [18.02.2017].

Frömmgen, A. et al., 2016. „ReMP TCP: Low Latency Multipath TCP," *IEEE ICC 2016 - Communication QoS, Reliabilty and Modeling Symposium*, Mai.

Frantti, T. & Majanen, M., 2013. „Real-Time Traffic Control for Multihomed Devices," *International Conference on Information Networking (ICOIN)*, S. 113-118.

GÉANT, *Europäisches Forschungsnetz*, [Online], 2016, Verfügbar unter: http://www.geant.org, [08.01.2017].

Gärtner, A., 2011. „Kommunizierende medizinische Systeme und Netzwerke, Kapitel 47," In (Hg.), R.K. *Medizintechnik*, Berlin Heidelberg, Springer-Verlag. S.773-80.

Gackowski, A. et al., 2011. „Development, implementation, and multicenter clinical validation of the TeleDICOM—advanced, interactive teleconsultation system," *Journal of Digital Imaging*, Band 24, S. 541–551.

Gärtner, A., 2006. *Telemedizin und computerunterstützte Medizin (Medizintechnik und Informationstechnologie)*, 1. Ausg., Köln, TÜV Media GmbH TÜV Rheinland Group.

Gärtner, A., 2012. „DIN EN 80001-1: Chancen und Potenziale für vernetzte Medizintechnik," *E-HEALTH-COM*, November, S. 1-17.

Gnuplot, *Gnuplot Project Page*, [Online], 2017, Verfügbar unter: http://gnuplot.info/, [14.05.2017].

Grönemeyer, D., *Süddeutsche Zeitung: Den Arzt online konsultieren*, [Online], 2015, Verfügbar unter: http://www.sueddeutsche.de/gesundheit/aussenansicht-naehe-schaffen-1.2586900, [25.06.2016].

Graf v.d. Schulenburg, J.-M. et al., 1995. *Ökonomische Evaluation telemedizinischer Projekte und Anwendungen*, 1. Ausg., Baden - Baden, BRD, Nomos (Gesundheitsökonomische Beiträge).

Gray, W. & Boehm-Davis, D., 2000. „Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior," *Journal of Experimental Psychology: Applied*, Band 6, Ausg. 4, S. 322-335.

Grinnemo, K.J. & Brunstrom, A., 2015. „A first study on using MPTCP to reduce latency for cloud based mobile applications," *2015 IEEE Symposium on Computers and Communication (ISCC)*, S. 64-69.

Gruber, J. & Strawczynski, L., 1985. „Subjective Effects of Variable Delay in Speech Clipping in Dynamically Managed Voice Systems," *IEEE Transactions on Communications*, August, Band 33, Ausg. 8.

Grudin, J., 1994. „CSCW: History and Focus," *IEEE Computer*, Band 27, Ausg. 5, S. 19-26.

Gutwin, C., 2001. „The Effects of Network Delays on Group Work in Real-Time Groupware," *Proceedings of the Seventh European Conference on Computer-Supported Cooperative Work, 16-20 September 2001*, S. 299-318.

Gutwin, C. & Penner, R., 2002. „Improving Interpretation of Remote Gestures with Telepointer Traces," *CSCW '02 Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 16. November, S. 49-57.

Hübscher, H. et al., 1999. *IT-Handbuch, IT-Systemelektroniker/-in, Fachinformatiker/-in*, 1. Ausg., Braunschweig, Westermann Schulbuchverlag GmbH (Hg.).

Haas, P., 2006. *Gesundheitstelematik - Grundlagen, Anwendungen, Potenziale*, 1. Ausg., Berlin Heidelberg, Springer-Verlag.

Hamann, K., 2002. *Möglichkeiten und Grenzen der Telepathologie in der Fetalpathologie*, Berlin, Medizinische Fakultät, Charité. Dissertation.

Hanly, E.J. & Broderick, T.J., 2005. „Telerobotic surgery," *Operative Techniques in General Surgery*, Band 7, Ausg. 4, S. 170–181.

Hardman, V., Sasse, A. & Watson, A., 1995. „Reliable audio for use over the Internet," *Proceedings of INET '95*, Juni, S. 171–178.

Heinanen, J., Baker, F., Weiss, W. & Wroclawski, J., 06/1999. *RFC 2597: Assured Forwarding PHB Group*, Internet Engineering Task Force.

Heller, M.A. & Schiff, W., 1991. „The Psychology of Touch," *Lawrence Erlbaum Associates*, S. 91–114.

Hensel, K., Schultz, C. & Gemünden, H.G., 2002. „Markteintrittstrategien und Netzwerkmanagement als kritische Erfolgsfaktoren telemedizinischer Dienstleistungen – erste empirische Bestätigungen," *Erfolgsfaktoren telemedizinischer Dienstleistungen, Ergebnisse der TU-Berlin im Auftrag BMBF*.

Hesmans, B. & Bonaventure, O., 2016. „An enhanced socket API for Multipath TCP," *ANRW '16 Proceedings of the 2016 Applied Networking Research Workshop*, Juli, S. 1-6.

Hesmans, B. et al., 2015. „SMAPP: Towards Smart Multipath TCP-enabled APPlications," *CoNEXT '15 Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, December 01-04*, S. 28.

Häcker, J., Reichwein, B. & Turad, N., 2008. *Telemedizin - Markt, Strategien, Unternehmensbewertung*, München, BRD, Oldenbourg Wissenschaftsverlag GmbH.

Honda, M., Nishida, Y., Raiciu, C. & Greenhalgh, A., 2011. „Is it Still Possible to Extend TCP?," *IMC '11 Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, Nov 02 - 04*, November, S. 181-194.

Hopps, C., 11/2000. *RFC 2992: Analysis of an Equal-Cost Multi-Path Algorithm*, Internet Engineering Task Force.

Horsch, A. & Handels, H., 2002. „Telematik im Gesundheitswesen," In Lehmann & Meyer zu Bexten *Handbuch der Medizinischen Informatik*, Hanser Fachbuch. S.569-606.

Hunger, A. & Klein, P., 2016. „Equalizing Latency Peaks Using a Redundant Multipath-TCP Scheme," *2016 International Conference on Information Networking (ICOIN)*, Januar, S. 184-189.

Hunger, A., Klein, P.A. & Verbunt, M., 2016. „rMPTCP, Evaluation of the Redundancy-Bandwidth Trade-off and Jitter Compensation in rMPTCP," *2016 8th IFIP International Conference on New Technology, Mobile & Security (NTMS)*.

Hunger, A., Shamsuddin, A.H. & Muchtar, A., 2013. „Over 10 Years of Cooperation between Universiti Kebangsaan Malaysia and University of Duisburg-Essen, Germany Case Study of the Development of A Fruitful Partnership," *6th International Forum on Engineering Education (IFEE 2012), Procedia Social Behavioral Sciences*, Band 102, S. 11-15.

Ibe, O.C., 2002. *Converged Network Architectures – Delivering Voice over IP, ATM and Frame Relay*, New York, USA, Wiley Computer Publishing.

IEEE, 1985. *802.3-1985 - IEEE Standards for Local Area Networks: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE.

IEEE, 2015. *IEEE Std 802.3: IEEE Standard for Ethernet*, New York, NY: IEEE.

IEEE, 2016. *IEEE Standard 8802-1Q-2016, ISO/IEC/IEEE International Standard, Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements, Part 1Q: Bridges and bridged networks*, The Institute of Electrical and Electronics Engineers.

IEEE, 5/2008. ISBN 1-55937-959-6 *IEEE Standard for Information technology - Telecommunications and information exchange between systems, Local and metropolitan area networks - Specific requirements, Part 2: Logical Link Control*, New York, USA: The Institute of Electrical and Electronics Engineers IEEE.

IETF, *Multipath TCP Deployments*, [Online], 2016, Verfügbar unter: https://www.ietfjournal.org/multipath-tcp-deployments/, [25.08.2016].

Information Sciences Institute, University of Southern California, 9/1981. *RFC 791: Internet Protocol*, Internet Engineering Task Force.

iPerf, *Project Homepage*, [Online], 2016, Verfügbar unter: https://iperf.fr/, [28.03.2016].

ITU, I.T.U., 05/2003. *ITU-T G.114: Transmission Systems and Media, Digital Systems and Networks, One-way transmission time*, International Telecommunication Union.

ITU, I.T.U., 07/1994. *ITU-T Rec. X.200: Data Networks and Open System Communications, Open Systems Interconnection - Model and Notation*, International Telecommunications Union.

ITU, I.T.U., 11/2001. *ITU-T G.1010: Transmission Systems and Media, Digital Systems and Networks, Quality of service and performance, End-user multimedia QoS categories*, International Telecommunication Union.

ITU, I.T.U., 12/2011. *ITU-T Y.1541: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks, Network performance objectives for IP-based services*, International Telecommunication Union.

ITU, *International Telecommunication Union*, [Online], 2017, Verfügbar unter: http://www.itu.int, [07.10.2016].

Jacobson, V., 1988. „Congestion Avoidance and Control," *SIGCOMM '88 Symposium proceedings on Communications architectures and protocols Aug 16-18*, Band 18, Ausg. 4, S. 314–329.

Jacobson, V., 1990. [Online] Verfügbar unter: ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt, [05.05.2017].

Jay, C. & Hubbold, R., 2006. „Quantifying the Effects of Latency on Sensory Feedback in Distributed Virtual Environments," *Proceedings of Virtual Images Seminar 2006, College de France*, Januar, S. 9-16.

Jäckel, A., 2004. „Chancen für eine Telematikplattform," In Jähn, K. & Nagel, E. *e-Health*, Berlin Heidelberg, Springer-Verlag. S.7-10.

Järvinen, I. et al., 2013. „Effect of Competing TCP Traffic on Interactive Real-Time Communication," In Roughan, & Chang, R. *Passive and Active Measurement, Lecture Notes in Computer Science*, Berlin Heidelberg, Springer-Verlag.

Järvinen, I., Chemmagate, B., Yi Ding, A. & Daniel, L., 2013. „Effect of Competing TCP Traffic on Interactive Real-Time Communication," *PAM 2013, LNCS 7799*, S. 94–103.

Johansen, R., 1991. „Teams for Tomorrow," *Proceedings of the Twenty-Fourth Annual Hawaii Intl. Conf. on System Sciences*, Band 3, S. 521-534.

Julsrud, P.R. et al., 1999. „Telemedicine consultations in congenital heart disease: assessment of advanced technical capabilities," *Mayo Clinic Proceedings*, Band 74, Ausg. 8, S. 758–763.

Kaidu, M. et al., 2004. „Development and evaluation of a teleradiology and videoconferencing system," *Journal of Telemedicine and Telecare*, Band 10, Ausg. 4, S. 214–218.

Karim, A. et al., 2013. „Telepointer technology in telemedicine: a review," *Biomedical Engineering Online*, Band 12, Ausg. 21, S. 1-19.

Khasawneh, F.A., BenMimoune, A., Kadoch, M. & Alom, A., 2015. „Multihoming admission and mobility management in wireless mesh network," *2015 International Conference on Computer, Information and Telecommunication Systems (CITS)*, Juli, S. 1-5.

Kim, J., Choi, W., Lim, H. & Par, K.-J., 2012. „Adaptive Selection of Multiple Paths for Delay-Sensitive Networked Control Systems," *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference, 19-22 Aug.*

Kirrmann, H., Hansson, M. & Müri, P., 2007. „IEC 62439 PRP: Bumpless Recovery for Highly Available, Hard Real-Time Industrial Networks," *IEEE, Emerging Technologies and Factory Automation, 2007. ETFA, Sep 25-28*.

Klar, R. & Pelikan, E., 2011. „Stand, Möglichkeiten und Grenzen der Telemedizin in Deutschland," In (Hg.), R.K. *Medizintechnik*, 4. Ausg., Berlin Heidelberg, BRD, Springer-Verlag GmbH. S.807-13.

Kochs, H.-D., 1984. *Zuverlässigkeit elektronischer Anlagen*, Berlin Heidelberg, Springer-Verlag.

Kurogi, T., Nakayama, M., Sato, K. & Kamuro, S., 2013. „Haptic Transmission System to Recognize Differences in Surface Textures of Objects for Telexistence," *Virtual Reality (VR), 2013 IEEE, 18-20 March*, März.

Kurose, J. & Ross, K., 2014. *Computernetzwerke, Der Top-Down-Ansatz*, 6. Ausg., Hallbergmoos, BRD, Pearson Deutschland GmbH.

Launois, C.d. & Bagnulo, M., 2006. „The Paths Towards IPv6 Multihoming," *IEEE Communications Surveys & Tutorials, 2nd Quarter 2006*, Band 8, Ausg. 2, S. 38-51.

Leffler, S.J., Joy, W.N., Fabry, R.S. & Karels , M.J., 1991. *Networking Implementation Notes 4.3BSD Edition, Technical Report*, Computer Systems Research Group, Computer Science Division Department of Electrical Engineering and Computer Science, University of California, Berkeley Berkeley, CA 94720.

Leon-Garcia, A. & Widjaja, I., 2004. *Communiation Networks – Fundamental Concepts and Key Architectures*, 2. Ausg., McGraw-Hill.

Liang, Y.J., Steinbach, E.G. & Girod, B., 2001. „Real-time Voice Communication over the Internet Using Packet Path Diversity," *MULTIMEDIA '01 Proceedings of the ninth ACM international conference on Multimedia, Sept 30 - Oct 05*, S. 431-440.

Liao, J., Wang, J. & Zhu, X., 2008. „A multi-path mechanism for reliable VoIP transmission over wireless networks," *Elsevier Computer Communications*, Band 52, Ausg. 13, S. 2450-2460.

Li, L. et al., 2015. „AMTCP: An Adaptive Multi-path Transmission Control Protocol," *CF '15 Proceedings of the 12th ACM International Conference on Computing Frontiers, May 18-21*, S. 29.

Liu, B. & Lu, Y., 2007. „A Scalable Peer-to-Peer Overlay for Applications with Time Constraints," *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 30 July-1 Aug, SNPD*, S. 662-667.

Llewellyn, C.H., 1995. „The Role of Telemedicine in Disaster Medicine," *Journal of Medical Systems*, Februar, Band 19, Ausg. 1, S. 29-34.

MacKenzie, I.S. & Jusoh, S., 2001. „An Evaluation of Two Input Devices for Remote Pointing," In M. Reed Little & L. Nigay (Hg.) *EHCI 2001, LNCS 2254*, Berlin-Heidelberg, Springer-Verlag. S.235-50.

Mahlous, A.R., Chaourar, B. & Mansour, M., 2009. „Performance Evaluation of Max Flow Multipath Protocol with Congestion Awareness," *International Conference on Advanced Information Networking and Applications Workshops*, S. 820-825.

Malindi, P., 2011. „QoS in Telemedicine, Telemedicine Techniques and Applications," In G. Graschew (Hg.) *Telemedicine Techniques and Applications*, Rijeka, Croatia, InTech.

Malkin, G., 11/1998. *RFC 2453: RIP Version 2*, Internet Engineering Task Force.

Marescaux, J. et al., 2001. „Transatlantic robot-assisted telesurgery," *Nature*, 27. September, Band 413, Ausg. 6854, S. 379–380.

Mathis, M. & Heffner, J., 03/2007. *RFC 4821, Packetization Layer Path MTU Discovery*, Internet Engineering Task Force.

Matusitz, J. & Breen, G.M., 2007. „Telemedicine: its effects on health communication," *Health Communication*, Band 21, Ausg. 1, S. 73-83.

Maxemchuk, N.F., 1975. „Dispersity Routing," *Proceedings of the IEEE ICC '75, June*, S. 41.10-41.13.

Maxis, *Internet Service Provider*, [Online], 2017, Verfügbar unter: http://www.maxis.com.my, [08.01.2017].

Minhas, T.N., Lagunas, O.G., Arlos, P. & Fiedler, M., 2012. „Mobile video sensitivity to packet loss and packet delay variation in terms of QoE," *Proceedings of 2012 IEEE 19th International Packet Video Workshop, May 10-11*, Mai, S. 83-88.

Ministry of Higher Education, M., *MyREN, Malaysian Research & Eduation Network*, [Online], 2017, Verfügbar unter: http://myren.net.my/, [08.01.2017].

Mogul, J. & Deering, S., 11/1990. *RFC 1191, Path MTU Discovery*, Internet Engineering Task Force.

Mohr, M.T.J., Schall, T. & Nerlich, M., 2004. „Telemedizin," In Jähn, K. & Nagel, E. *e-Health*, Berlin Heidelberg, BRD, Springer-Verlag. S.35-39.

MTR, *Github Fork Download*, [Online], 2013, Verfügbar unter: https://github.com/PofigNaNik/mtr, [12.01.2017].

Murthy, V.K. & Krishnamurthy, E.V., 2005. „Multimedia Computing Environment for Telemedical Applications," *In Encyclopedia of Information Science and Technology, IGI Global*, S. 2045–2050.

Muuss, M., *The Story of the PING Program*, [Online], o.D., Verfügbar unter: http://ftp.arl.mil/~mike/ping.html, [17.02.2017].

Nacenta, M.A., Pinelle, D., Stuckel, D. & Gutwin, C., 2007. „The Effects of Interaction Technique on Coordination in Tabletop Groupware," *Proceeding GI '07 Proceedings of Graphics Interface 2007*, S. 191-198.

Nagle, J., 01/1984. *RFC 896, Congestion Control in IP/TCP Internetworks*, Internet Engineering Task Force.

NASA (Hg.), 1985. „NASA satellite aids in Mexico City rescue effort.," *NASA News*, S. 85-133.

netem, *netem MAN-page*, [Online], 2017, Verfügbar unter: http://man7.org/linux/man-pages/man8/tc-netem.8.html, [14.05.2017].

Nguyen, S.C. & Nguyen, T.M.T., 2011. „Evaluation of Multipath TCP Load Sharing with Coupled Congestion Control Option in Heterogeneous Networks," *Global Information Infrastructure Symposium (GIIS)*, August, S. 1-5.

Nichols, K., Blake, S., Baker, F. & Black, D., 12/1998. *RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, Internet Engineering Task Force.

Nora, S. & Minc, A., 1978. *L'informatisation de la société: rapport à M. le Président de la République*, Paris : Seuil.

Nordmark, E. & Bagnulo, M., 06/2009. *RFC 5533: Shim6: Level 3 Multihoming Shim Protocol for IPv6*, Internet Engineering Task Force.

Oh, H., Rizo, C., Enkin, M. & Jadad, A., 2005. „What is eHealth?: a systematic review of published definitions," *World Hospitals and Health Services*, Band 41, Ausg. 1, S. 32-40.

Ohta, S. et al., 2006. „Remote support for emergency medicine using a remote-control laser pointer," *Journal of Telemededicine and Telecare*, Band 12, S. 44–48.

Ong, L. & Yoakum, J., 5/2002. *RFC 3286: An Introduction to the Stream Control Transmission Protocol (SCTP)*, Internet Engineering Task Force.

Oodan, A. et al., 2003. *Telecommunications Quality of Service Management, from legacy to emerging services*, London, UK, IEE telecommunications series.

Open Networking Foundation, 2012. [Online] Verfügbar unter: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf, [13.4.2017].

Opstad, B.R. et al., 2015. „Latency and fairness trade-off for thin streams using redundant data bundling in TCP," *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, S. 287-294.

Oran, D., 2/1990. *RFC 1142: OSI IS-IS Intra-domain Routing Protocol*, Internet Engineering Task Force.

Paasch, C., 2012. *MultiPath TCP: Linux Kernel implementation* [Online] Verfügbar unter: http://www.linuxplumbersconf.org/2012/wp-content/uploads/2012/08/christoph-paasch- networking.pdf, [16.06.2016].

Paasch, C., 2014. *Improving Multipath TCP, Doctor Thesis*, Louvain.

Paasch, C. & al., e., *Configure MPTCP*, [Online], 2016, Verfügbar unter: http://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP, [16.06.2016].

Paasch, C. & Barre, S., *Multipath TCP in the Linux Kernel*, [Online], 2014, Verfügbar unter: http://www.multipath-tcp.org, [05.04.2016].

Paasch, C., Ferlin, S., Alay, O. & Bonaventure, O., 2014. „Experimental Evaluation of Multipath TCP Schedulers," *CSWS'14 Proceedings of the ACM SIGCOMM workshop on Capacity sharing*, August, S. 27-32.

Paasch, C., Khalili, R. & Bonaventure, O., 2013. „On the Benefits of Applying Experimental Design to Improve Multipath TCP," *CoNEXT'13 Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, Dezember, S. 393-398.

Pala, M., Lorencik, D. & Sincak, P., 2012. „Towards the robotic teleoperation systems in education," *ICETA 2012, 10th IEEE International Conference on Emerging eLearning Technologies and Applications*, S. 241-246.

Palmer, M., Steffen, C., Iakovidis, I. & Giorgio, F., 2009. „European Commission perspective: Telemedicine for the benefit of patients, health care systems and society," *Eurohealth*, Band 15, Ausg. 1, S. 13-15.

Pavlovych, A. & Stuerzlinger, W., 2011. „Target following performance in the presence of latency, jitter, and signal dropouts," *Graphics Interface Conference 2011 25-27 May*, S. 33–40.

Paxson, V., 1997. *Measurements and Analysis of End-to-End Internet Dynamics, Ph.D. Thesis*, Berkeley, CA, USA.

Paxson, V. & Allman, M., 11/2000. *RFC 2988, Computing TCP's Retransmission Timer*, Internet Engineering Task Force.

Peng, Q., Walid, A. & Low, S.H., 2013. „Multipath TCP Algorithms: Theory and Design," *SIGMETRICS'13 Proceedings of the ACM SIGMETRICS/ international conference on Measurement and modeling of computer systems,* , S. 305-316.

Petlund, A., Evensen, K., Griwodz, C. & Halvorsen, P., 2008. „TCP mechanisms for improving the user experience for time-dependent thin-stream applications," *2008 33rd IEEE Conference on Local Computer Networks (LCN),* S. 176-183.

Plummer, D.C., 11/1982. *RFC 826, An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48 bit Ethernet Address for Transmission on Ethernet Hardware*, Internet Engineering Task Force.

Pohlmann, N. & Dierichs, S., *Heise - So funktioniert Internet-Routing*, [Online], 2008, Verfügbar unter: https://www.heise.de/ct/artikel/So-funktioniert-Internet-Routing-221495.html, [23.06.2017].

Polycom, *Videokonferenzen*, [Online], 2017, Verfügbar unter: http://www.polycom.de, [08.01.2017].

Postel, J., 09/1981. *RFC 793, Transmission Control Protocol*, Internet Engineering Task Force.

Postel, J., 11/1983. *RFC 879, The TCP Maximum Segment Size and Related Topics*, Internet Engieering Task Force.

Postel, J., 9/1981. *RFC 792, Internet Control Message Protocol*, Internet Engineering Task Force.

Python, *Projekt Homepage*, [Online], 2017, Verfügbar unter: http://www.python.org, [26.03.2017].

Qadir, J., Anwaar, A. & Yau, K.-L.A., 2015. „Exploiting the power of multiplicity: A holistic survey of network-layer multipath," *IEEE Communications Surveys & Tutorials*, Juli, Band 17, Ausg. 4, S. 2176-2213.

Raiciu, C., Handly, M. & Wischik, D., 10/2011. *RFC 6356, Coupled Congestion Control for Multipath Transport Protocols*, Internet Engineering Task Force.

Raiciu, C. et al., 2012. „How hard can it be? designing and implementing a deployable multipath TCP," *NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, Apr 25- 27*, April, S. 29-29.

Reiter, B., Jürgen, T. & Weidenfeld, W., 2011. „Telemedizin – Zukunftsgut im Gesundheitswesen, Gesundheitspolitik und Gesundheitsökonomie zwischen Markt und Staat," *Forschungsgruppe Zukunftsfragen, CAP Analyse*, Januar, Band 1.

Rekhter, Y., Li, T. & Hares, S., 01/2006. *RFC 4271: A Border Gateway Protocol 4 (BGP-4)*, Internet Engieering Task Force.

R, *Project for Statistical Computing*, [Online], o.D., Verfügbar unter: https://www.r-project.org/, [28.03.2016].

Rosen, E., Viswanathan, A. & Callon, R., 01/2001. *RFC 3031: Multiprotocol Label Switching Architecture*, Internet Engineering Task Force.

Sánchez, J.A., Strazzulla, D. & Paredes, R.G., 2008. „Enhancing Interaction and Collaboration in Multimedia Rooms with Multilayered Annotations and Telepointers," *IHC 2008 – VIII Simpósio Sobre Fatores Humanos em Sistemas Computacionais, October 21-24*, S. 117-125.

Saddik, A.E., Orozco, M., Eid, M. & Cha, J., 2011. *Haptics Technologies, Bringing Touch to Multimedia*, 1. Ausg., Berlin Heidelberg, BRD, Springer-Verlag.

Saltzer, J.H., Reed, D.P. & Clark, D.D., 1984. „End-To-End Arguments in System Design," *ACM Transmission on Computer Systems (TOCS)*, November, Band 2, Ausg. 4, S. 277-288.

Sathiaseelan, A. & Radzik, T., 2004. „Improving the Performance of TCP in the Case of Packet Reordering," *HSNMC 2004, LNCS 3079*, S. 63–73.

Savage, S., Collins, A., Hoffman, E. & Snell, J., 1999. „The End-to-End Effects of Internet Path Selection," *SIGCOMM '99 Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, Aug 30 - Sep 03*, Band 29, Ausg. 4, S. 289-299.

Scharf, M. & Ford, A., 03/2013. *RFC 6897, Multipath TCP (MPTCP) Application Interface Considerations*, Internet Engineering Task Force.

Schmid, A. & Steigner, C., 2002. „Avoiding Counting to Infinity in Distance Vector Routing," *Telecommunication Systems*, März, Band 19, Ausg. 3, S. 497-514.

Schmidt, K., 2002. „The Problem with 'Awareness': Introductory Remarks on 'Awareness in CSCW'," *Journal Computer Supported Cooperative Work*, Band 11, Ausg. 3, S. 285-298.

Schrempf, A. & Zauner, M., 2009. *Informatik in der Medizintechnik - Grundlagen, Software, Computergestützte Systeme*, Wien, Österreich, Springer-Verlag.

Schulzrinne, H., Casner, S., Frederick, R. & Jacobson, V., 7/2003. *RFC 3550: RTP: A Transport Protocol for Real-Time Applications*, Internet Engineering Task Force.

Seth, S. & Venkatesulu, M.A., 2008. *TCP/IP Architecture, Design and Implementation in Linux*, NJ, USA, Wiley & Sons Inc, IEEE.

Shacham, N. & McKenney, P., 1990. „Packet Recovery in High-Speed Networks Using Coding and Buffer Management," *INFOCOM '90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. 3.-7. June, San Francisco, The Multiple Facets of Integration, Proceedings*, Juni.

Shuminoski, T. & Janevski, T., 2016. „Lyapunov Optimization Framework for 5G Mobile Nodes With Multi-Homing," *IEEE Communications Letters*, Mai, Band 20, Ausg. 5, S. 1026-1029.

Siahaan, A., 2015. *Masterarbeit unter der Betreuung von Pascal A. Klein: Implementation and Evaluation of a Multipath TCP Simulation Environment for Tele-medical Applications*, Duisburg, technische Informatik, Universität Duisburg-Essen.

Singh, V., Ahsan, S. & Ott, J., 2013. „MPRTP: Multipath Considerations for Real-time Media," *Proceedings of the 4th ACM Multimedia Systems Conference, Feb 28 - Mar 01*, S. 190-201.

Singla, A., Chandrasekaran, B., Godfrey, P.B. & Maggs, B., 2014. „The Internet at the Speed of Light," *Proceeding HotNets-XIII Proceedings of the 13th ACM Workshop on Hot Topics in Networks, October 27–28,* S. 1.

Skorin-Kapov, L. & Matijasevic, M., 2010. „Analysis of QoS Requirements for e-Health Services and Mapping to Evolved Packet System QoS Classes," *International Journal of Telemedicine and Applications*.

Smithwick, M., 1995. „Network options for wide-area telesurgery," *Journal of Telemedicine and Telecare*, Band 1, Ausg. 3, S. 131-138.

Sood, S. et al., 2007. „What Is Telemedicine? A Collection of 104 Peer-Reviewed Perspectives and Theoretical Underpinnings," *Telemedicine and e-Health*, November, Band 13, Ausg. 5, S. 573-590.

Stefik, M. et al., 1987. „WYSIWIS Revised: Early Experiences with Multiuser Interfaces," *ACM Transactions on Office Information Systems*, April, Band 5, Ausg. 2, S. 147-167.

Stevens, W., 01/1997. *RFC 2001: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*, Internet Engineering Task Force.

Szymanski, T.H.u.G.D., 2010. „Provisioning Mission-Critical Telerobotic Control Systems over Internet Backbone Networks with Essentially-Perfect QoS," *IEEE Journal on selected areas in communications*, Juni, Band 28, Ausg. 5, S. 630-643.

Tanenbaum, A.S. & Wetherall, D.J., 2012. *Computernetzwerke*, 5. Ausg., München, BRD, Pearson Deutschland GmbH.

TeamViewer, *TeamViewer Fernwartungstool*, [Online], o.D., Verfügbar unter: https://www.teamviewer.com/de/, [05.05.2017].

TEIN3 (Hg.), 2013. *TEIN3 – The Research and Education Network for Asia-Pacific, Brochure* [Online] Verfügbar unter: https://www.tein3.net/Media_Centre/Documents/TEIN3%20Brochure.pdf, [28.06.2016].

Tesla, N., 1898. „Method of and Apparatus for Controlling Mechanism of Moving Vessels on Vehicles," *US-Patent 68,809*, 8. November, URL: http://www.google.com/patents/US613809.

Tobagi, F.A., 2005. „A Top Down View of Quality of Service," *ST Journal of Research, Network Media*, Band 2, Ausg. 1.

Trueöl, K., 1984. „Aufbau eines deutschen Forschungsnetzes, Stand der Realisierungen und Konzepte zum Betrieb," *Zentrale Projektleitung des DFN – Berlin, Ges. f. Mathematik und Datenverarbeitung in Informatik-Fachberichte 96, 6. KI Fachgespräch über Rechenzentren*, Dezember.

Tsai, T.-Y., Chung, Y.-L. & Tsai, Z., 2010. „Introduction to Packet Scheduling Algorithms for Communication Networks," In Peng, J. *Communications and Networking*, Rijeka, Croatia, InTech. S.263-88.

Tsai, J. & Moors, T., 2006. „A Review of Multipath Routing Protocols: From Wireless Ad Hoc to Mesh Networks," *ACoRN Early Career Researcher Workshop on Wireless Multihop Networking, 17-18 July*, Januar.

uMobile, *Communications*, [Online], 2017, Verfügbar unter: http://www.u.com.my, [08.01.2017].

Van, G.W.J., 2004. „Jitter compensation method for systems having wall clocks," *WO Patent App. PCT/IB2003/002,992*, 29. Januar, url: http://www.google.ch/patents/WO2004010670A1?cl=en.

VDI, V.D.I.(.)., 1990. *VDI 2860: Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole*, Verein Deutscher Ingenieure.

Verbunt, M., 2017. *Masterarbeit unter der Betreuung von Pascal A. Klein: Implementation and Enhancement of an Adaptive Redundant Multipath Transport Protocol*, Duisburg, technische Informatik, Universität Duisburg-Essen.

Vulmiri, A., Michel, O., Brighten Godfrey, P. & Shenker, S., 2012. „More is Less: Reducing Latency via Redundancy," *HotNets-XI Proceedings of the 11th ACM Workshop on Hot Topics in Networks, Oct 29 - 30*, S. 13-18.

W3C, *The Original HTTP as defined in 1991*, [Online], 1991, Verfügbar unter: https://www.w3.org/Protocols/HTTP/AsImplemented.html, [28.3.2016].

W3C, *HTML5, A vocabulary and associated APIs for HTML and XHTML*, [Online], 2014, Verfügbar unter: https://www.w3.org/TR/html5/single-page.html, [11.04.2017].

Wang, J., Zhou, M. & Li, Y., 2004. „Survey on the End-to-End Internet Delay Measurements," In Mammeri, Z. & Lorenz, P. *High Speed Networks and Multimedia Communications. HSNMC 2004, Lecture Notes in Computer Science*, Berlin-Heidelberg, BRD, Springer. S.155-66.

Wasem, J., Staudt, S., Matusiewicz, D. & (Hg.), 2013. *Medizinmanagement: Grundlagen und Praxis des Management in Gesundheitssystem*, 1. Ausg., Essen, Medizinisch Wissenschaftliche Verlagsgesellschaft.

WebEx, *Cisco WebEx Online-Meeting*, [Online], 2017, Verfügbar unter: https://www.webex.de/, [06.05.2017].

Wehrle, K. et al., 2002. *Linux Netzwerkarchitektur: Design und Implementierung von Netzwerkprotokollen im Linux-Kern*, 1. Ausg..

Werner, S., 2002. *Synchrone Groupware für die Software-Engineering-Ausbildung - Ein Beispiel für die Ableitung unterstützender Werkzeuge aus problemorientierter Sicht, Dissertation*, Duisburg, BRD, Universität Duisburg-Essen.

wget, *HTTP Kommandozeilentoll*, [Online], 2017, Verfügbar unter: https://www.gnu.org/software/wget/, [28.3.2016].

Williams, N. & Melnikov, A., 5/2008. *RFC 5178: Generic Security Service Application Program Interface (GSS-API), Internationalization and Domain-Based Service Names and Name Type*, Internet Engieering Task Force.

Willinger, W. & Doyle, J., *Robustness and the Internet: Design and evolution*, [Online], 2002, Verfügbar unter: https://pdfs.semanticscholar.org/bd84/3c1ae2601a1e6004ea7a6e76042ea21dac22.pdf, [08.11.2016].

Winther, M., 2006. [Online] Verfügbar unter: http://www.us.ntt.net/downloads/papers/IDC_Tier1_ISPs.pdf, [28.12.2016].

Wireshark, *Protocol Analyzer*, [Online], 2017, Verfügbar unter: https://www.wireshark.org/, [28.3.2016].

Wischik, D., Handley, M. & Braun, M.B., 2008. „The Resource Pooling Principle," *ACM SIGCOMM Computer Communication Review*, Oktober, Band 38, Ausg. 05, S. 47-52.

Wischik, D., Raiciu, C., Greenhalgh, A. & Handley, M., 2011. „Design, implementation and evaluation of congestion control for multipath TCP," *NSDI'11 Proceedings of the 8th USENIX conference on Networked systems design and implementation*, S. 99-112.

Wootton, R., 2008. „Telemedicine support for the developing world," *Journal of Telemedicine and Telecare*, Band 14, S. 109-114.

World Health Organization, 1998. *A Health Telematics Policy*, Geneva: Report of the WHO Group Consultation on Health Telematics, 1997, 11-16 Dec.

World Health Organization, 2015. *Third Global Survey on eHealth 2015: The use of eHealth in support of universal health coverage*, World Health Organization.

World Health Organization, 2016. *Atlas of eHealth country profiles - The use of eHealth in support of universal health coverage*, Genf, Schweiz, World Health Organization.

Xu, H. & Li, B., 2014. „RepFlow: Minimizing flow completion times with replicated flows in data centers," *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, S. 1581-1589.

Yamazaki, K., Yamazaki, A., Kuzuoka, H. & Oyama, S., 1999. „GestureLaser and GestureLaser Car: Development of an embodied Space to Support remote Instruction," *Proceeding ECSCW'99 Proceedings of the sixth conference on European Conference on Computer Supported Cooperative Work*, S. 239–258.

Yang, F., Wang, Q. & Amer, P.D., 2014. „Out-of-order Transmission for In- order Arrival Scheduling for Multipath TCP," *28th International Conference on Advanced Information Networking and Applications Workshops (WAINA),* S. 749-752.

Ye, N., 2002. „QoS-centric stateful resource management in information systems," *Information Systems Frontiers*, Band 4, Ausg. 2, S. 149-160.

Zhang, M., 2005. *Understanding Internet Routing Anomalies and Building Robust Transport Layer Protocols, Dissertation*, NJ, USA: Princeton University.

Zhang, J., 2016. *Masterarbeit unter der Betreuung von Pascal A. Klein: Development of an MPTCP proxy bridging environment as a gateway for Real Time applications without MPTCP capabilities*, Duisburg, technische Informatik, Universität Duisburg-Essen.

Zhang, M. et al., 2004. „A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths," *ATEC '04 Proceedings of the annual conference on USENIX Annual Technical Conference*, Juni, S. 8-8.

Zimmermann, H., 1980. „OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, April, Band COM-28, Ausg. 4.

Ziv, J. & Lempel, A., 1977. „A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, Mai, Band 23, Ausg. 3, S. 337-343.

Zuberbühler, H., Ruegg, S., Krueger, H. & Kündig, A., 2002. „Intermedia Synchronisation in Network Design: Using an Adaptive Psychophysical Method to Specify the Perceivable Audio-Visual Delay," *Proceedings of the Conference WWDU 2002 World Wide Work, May 22-25*, Januar, S. 107-109.

Zundel, K.M., 1996. „Telemedicine: History, Applications, and Impact on Librarianship," *Bulletin of the Medical Library Association*, Band 84, Ausg. 1, S. 71-79.

Auflistungen der Veröffentlichungen

Hunger, Axel; Werner, Stefan; Kärchner-Ober, Renate; Klein, Pascal A., „Global Engineering and CSCW (Computer Supported Cooperative Working). Ein Beispiel zur Fusion von Wissensbereichen," Erschienen in: Unikate 45/2014, S. 122-133.

Hunger, Axel; Klein, Pascal A., „Equalizing Latency Peaks Using a Redundant Multipath-TCP Scheme," 2016 International Conference on Information Networking (ICOIN), Kota Kinabalu, Malaysia, 2016, S. 184-189.

Hunger, Axel; Klein, Pascal A., Verbunt, Martin H., „Evaluation of the Redundancy-Bandwidth Trade-off and Jitter Compensation in rMPTCP," 2016 8th IFIP International Conference on New Technology, Mobile & Security (NTMS), Larnaca, Cyprus, 2016.

Betreute studentische Abschlussarbeiten

Cai, J., 2015, Masterarbeit, „Workflow Modeling of Cooperative Tele-medical Applications," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Khanshaghaghi, M., 2016, Masterarbeit, „Elaboration and Evaluation of a Scheduler Mechanism for Quality of Service Gateways," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Krishnamurthy, H., 2015, Masterarbeit, „Quality of Service and Interaction workflow model for cooperative Telemedical systems," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Nicodemus, M., 2017, Masterarbeit, „Sending and Receiving Efficiency of a redundant Multipath Transport Protocol in Case of Delay sensitive Applications," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Nweke, F.O., 2016, Bachelorarbeit, „Classifications of telepointers and their usage in telemedicine," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Püttmann, M., 2014, Bachelorarbeit, „Komponenten der Telekooperation für den Einsatz in der Telemedizin – Technische Effizienz unter rechtlichen Vorgaben," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Shoja, M., 2015, Bachelorarbeit, „Study of open and global research networks and infrastructure to establish a reliable link between Germany and Malaysia," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Siahaan, A., 2015, Masterarbeit, „Implementation and Evaluation of a Multipath TCP Simulation Environment for Tele-medical Applications," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Verbunt, M., 2017, Masterarbeit, „Implementation and Enhancement of an Adaptive Redundant Multipath Transport Protocol," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Wu, Y., 2014, Bachelorarbeit, „Quality of Service of Interconnected Entities in Telemedical Systems," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Yilmaz, Ö., 2015, Bachelorarbeit, „Steuerungsprotokolle von telemedizinischen Robotern," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

Zhang, J., 2016, Masterarbeit, „Development of an MPTCP proxy bridging environment as a gateway for Real Time applications without MPTCP capabilities," Duisburg, technische Informatik, Fakultät Ingenieurswissenschaften, Universität Duisburg-Essen.

10 Anhang

10.1 Anhang A – Forschungspartnerschaft UDE – UKM

Die Partnerschaft zwischen den ingenieurwissenschaftlichen Fakultäten der UDE und UKM existiert bereits seit dem Jahr 1997 [Hunger et al., 2013]. Erste auslandsorientierte Studiengänge wurden seit dieser Zeit durch den DAAD gefördert, woraus sich weitere Partnerschaften der ingenieurwissenschaftlichen Fakultät der UDE in Südostasien entwickelten. Sie wurden seit 2002 durch den Aufbau von Verbindungsbüros institutionalisiert (ebd.). Seitdem wurden mehrere gemeinsame Studiengänge und Promotionsprogramme entwickelt. Im Jahr 2012 wurde das Mercator Science and Education Sdn. Bhd. als eine eigenständige Firma in Malaysia gegründet, um gemeinsame Bildungs- und Forschungsprogramme zu fördern und zu finanzieren (ebd.).

Neben dem Austausch von Studierenden haben zahlreiche Gastprofessuren und Austausche von Forschern stattgefunden. Dabei erstrecken sich Forschungs- und Entwicklungsprojekte auf ingenieurswissenschaftliche Themen, die durch die Mitglieder der kooperierenden Abteilungen eingebracht werden, aber auch auf verwandte Themenbereiche, die auf die internationale Kooperation bezogen sind [Hunger et al., 2013]. In den Ingenieurswissenschaften wurden mehrere Partnerschaften mit der Industrie eingegangen. In mehreren Disziplinen wurden Labore am Standort in Malaysia eingerichtet, die heute fester Bestandteil der gemeinsamen Forschung sind. Zusammenarbeit besteht unter anderem in den Bereichen Materialforschung, Elektrotechnik, Medizintechnik, Umwelttechnik und Ingenieurswissenschaftliche Ausbildung (ebd.).

Ein wesentlicher Bestandteil der Zusammenarbeit in der Elektrotechnik und der technischen Informatik ist die Medizintechnik. In diesem Bereich werden zurzeit mehrere Forschungsprojekte durchgeführt. Die Projekte fokussieren in besonderem Maße die Verbesserung der Leitungsgüte, die für viele Anwendungen benötigt wird. Mit dem Schwellenlandstatus Malaysias verbunden sind Probleme der Breitbandnetzwerke und Verbindungen, die in Hinsicht auf die angedachten Applikationen erforscht und verbessert werden müssen. Für medizinische Anwendungen besonders problematisch sind Verbindungsabbrüche sowie Schwankungen hinsichtlich der Datenübertragungsrate und der Leitungsverzögerung.[49]

Aktuelle Forschungsschwerpunkte liegen in der Notfalltelemedizin, der stationären Telemedizin und der gemeinsamen Ausbildung im akademischen Umfeld.

[49] Vgl. Kapitel 2.4.6

10.2 Anhang B – Weathermap des X-Win Neztwerks

Abbildung 10.1 zeigt eine „Weathermap“ mit dem aktuellen Status des X-WiN und seinen Dienstgüte-Kennzahlen.

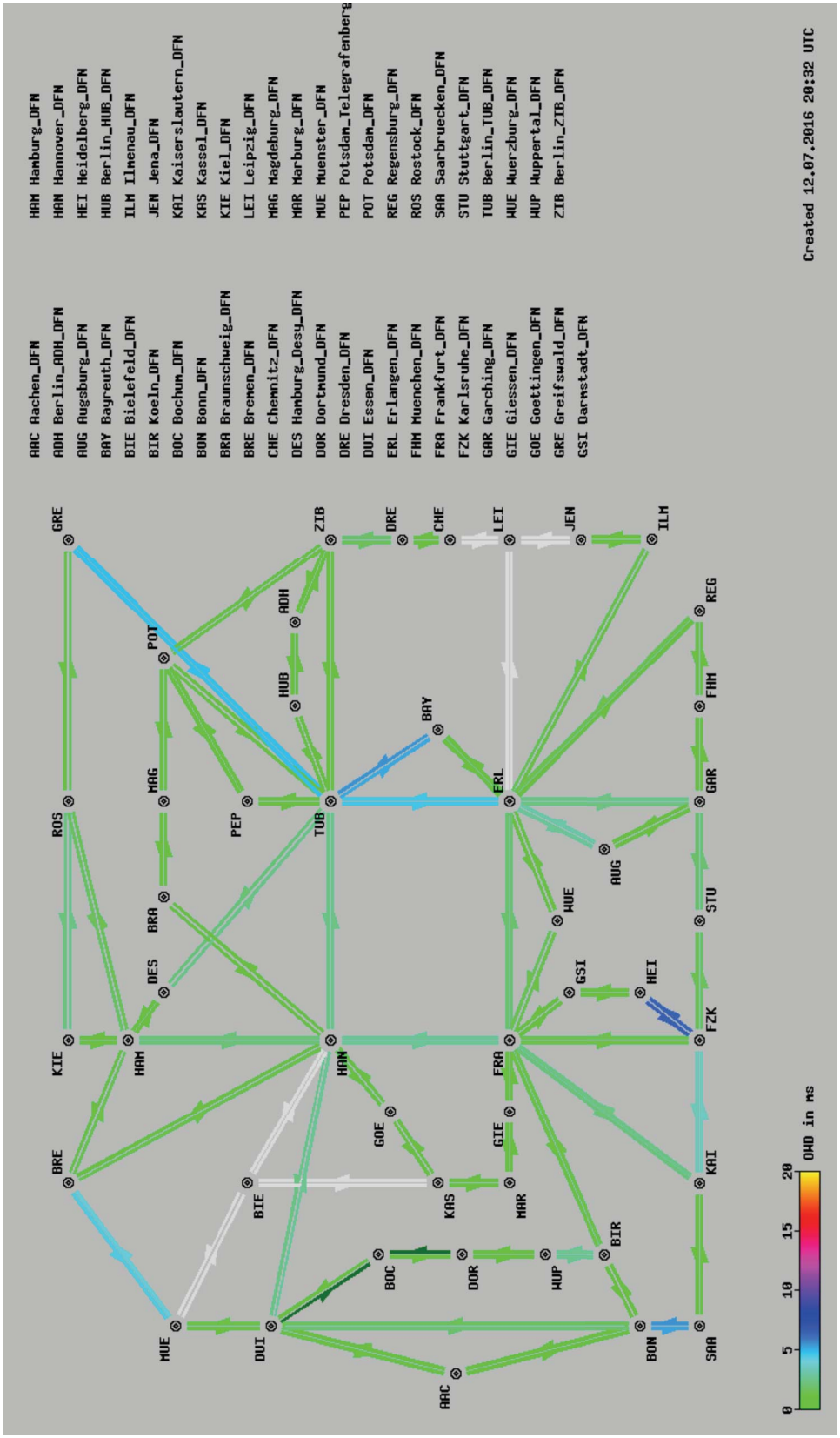

Abbildung 10.1: Weathermap des X-WiN (Zeitpunkt 12.07.2016, 20:32 Uhr, [DFN, 2016])

10.3 Anhang C – Statistiken der ICMP-Latenzzeiten

Die nachfolgenden Statistiken vermitteln einen Eindruck der gemessenen Latenzzeiten zwischen UDE und UKM über die drei Netzwerkverbindungen der UKM. Die Werte entstammen Tabelle 4.2, Tabelle 4.3 und Tabelle 4.4.

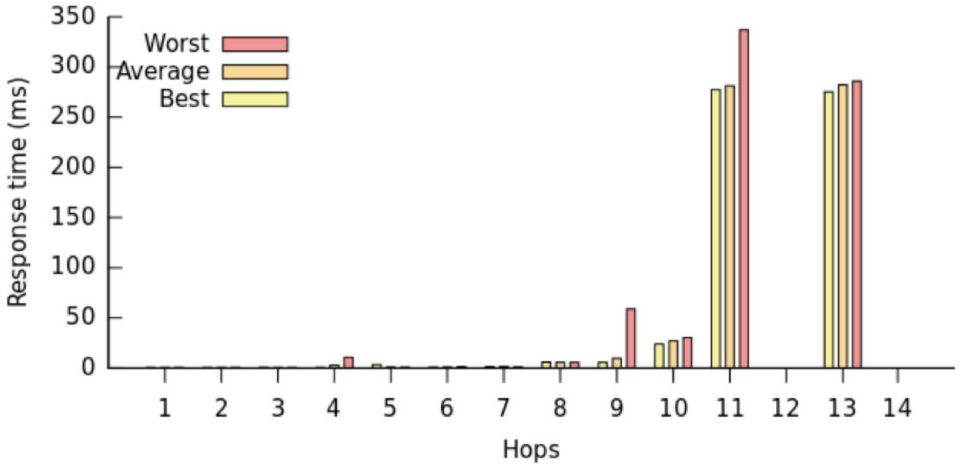

Abbildung 10.2: Traceroute Statistik der UKM Standard Netzwerkverbindung

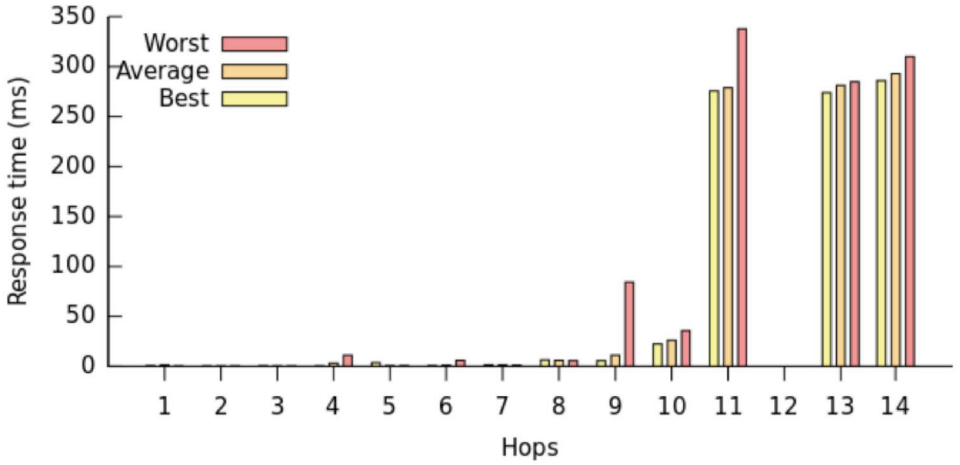

Abbildung 10.3: Traceroute Statistik der UKM Polycom Netzwerkverbindung

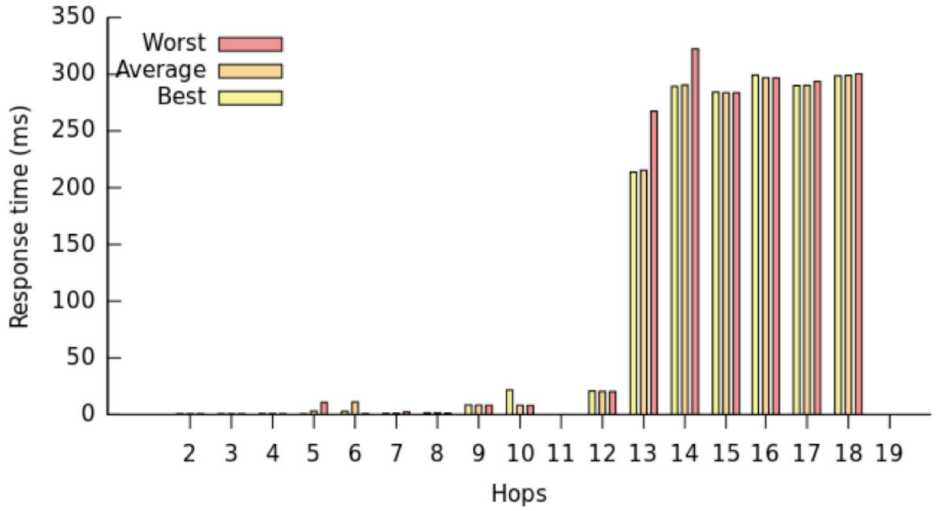

Abbildung 10.4: Traceroute Statistik der UKM 3G Netzwerkverbindung

10.4 Anhang D –Umlaufzeit der Verbindung UDE-UKM

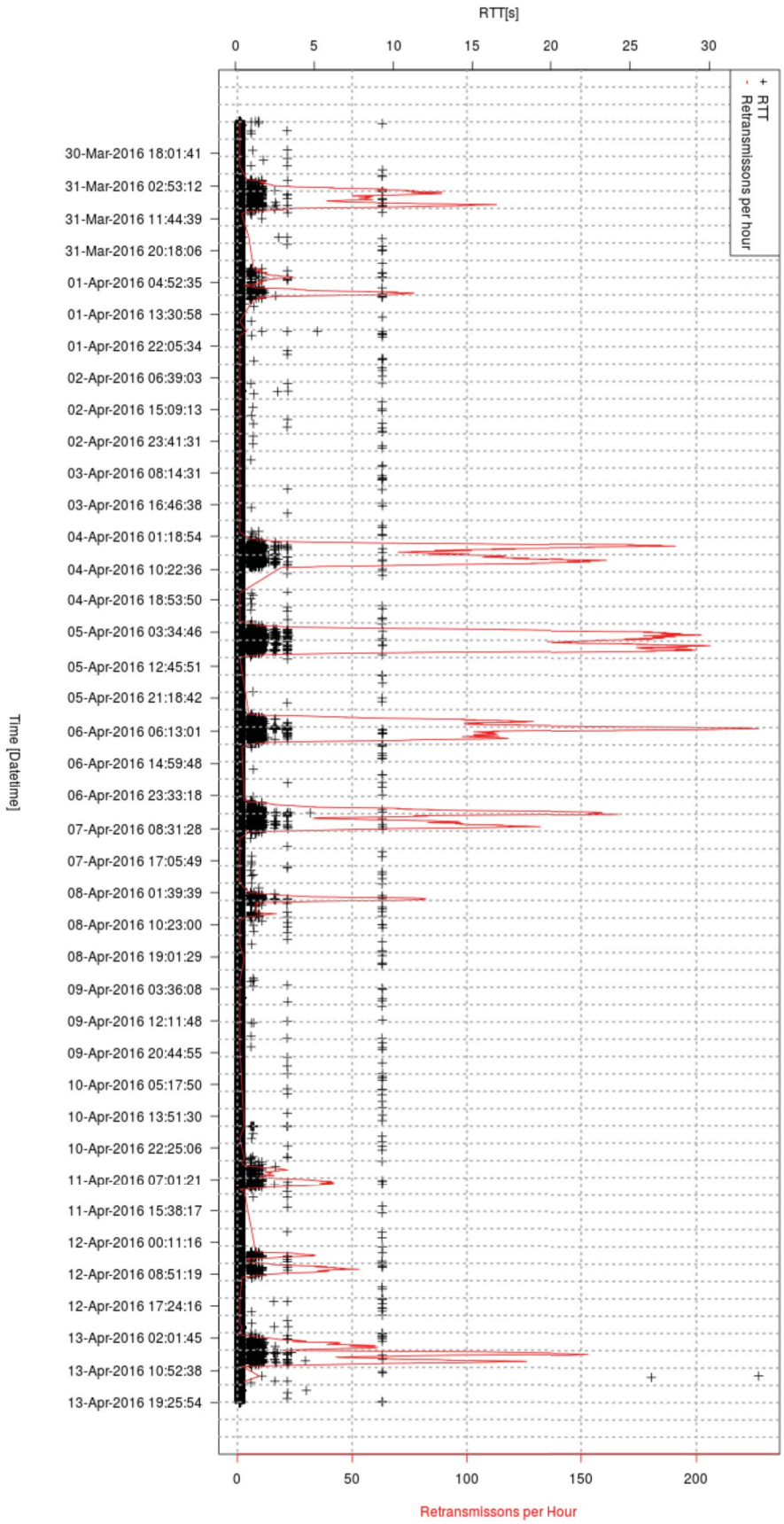

Abbildung 10.5: Umlaufzeitmessung UKM-Standardverbindung

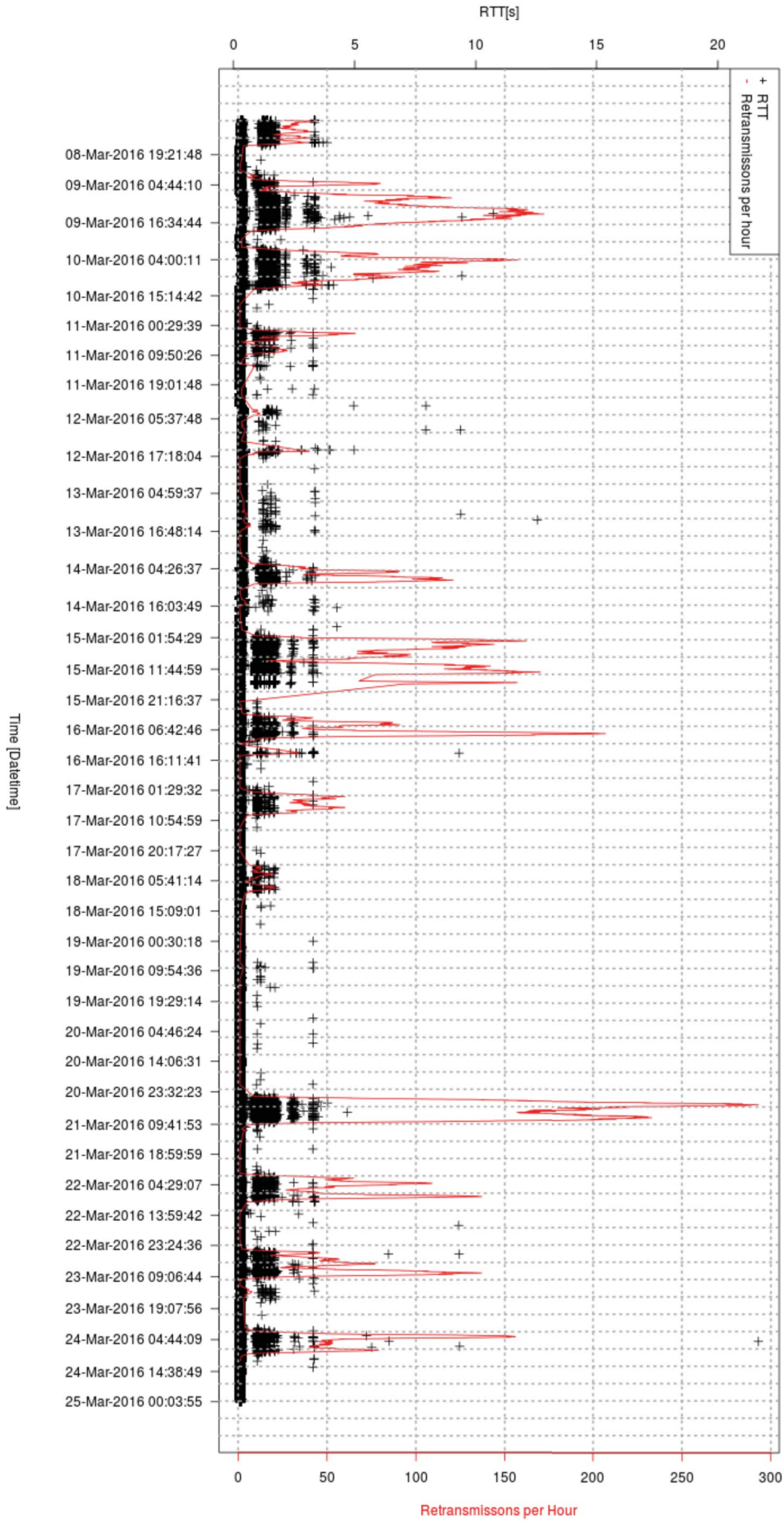

Abbildung 10.6: Umlaufzeitmessung UKM-Polycomverbindung

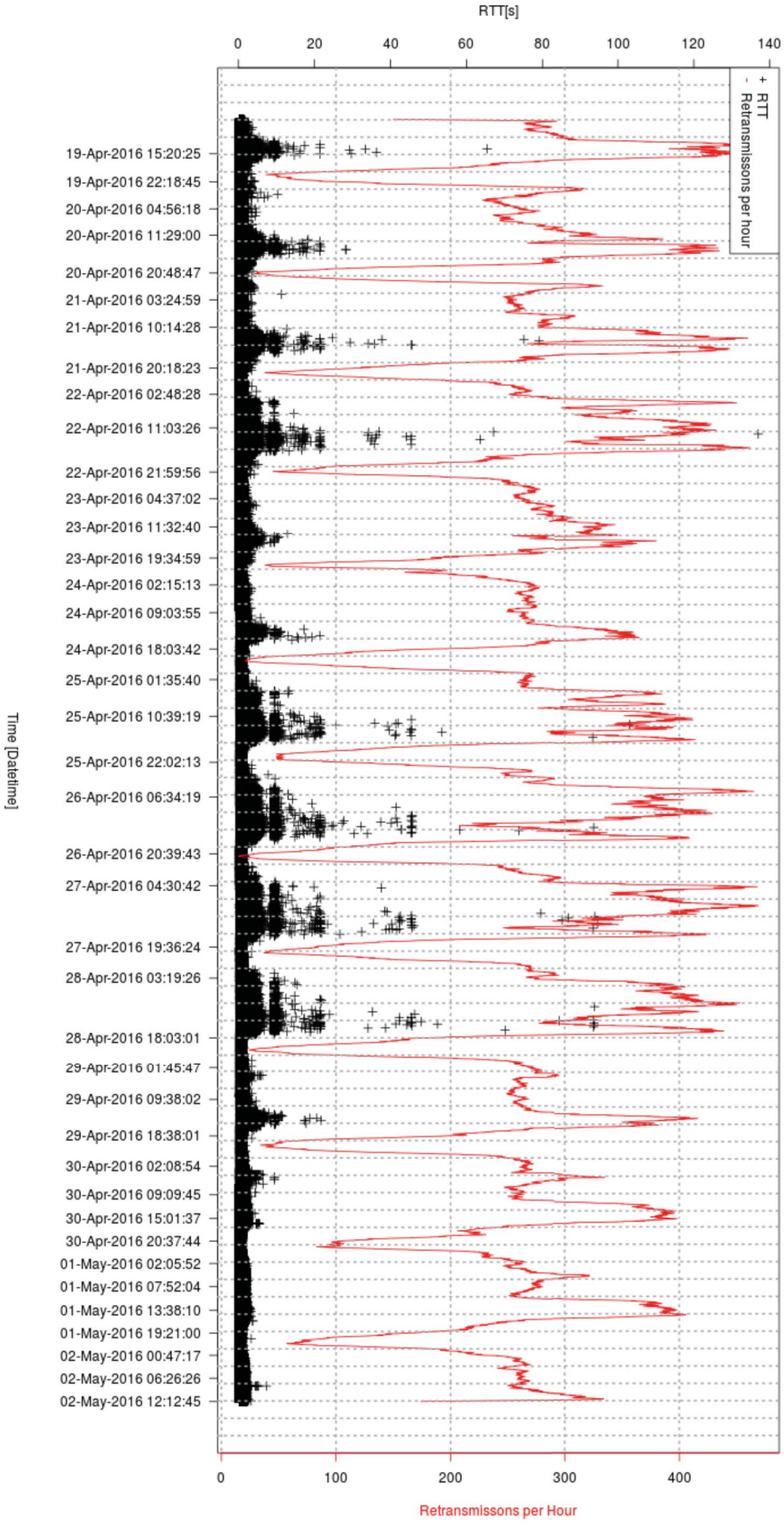

Abbildung 10.7: Umlaufzeitmessung 3G-Verbindung

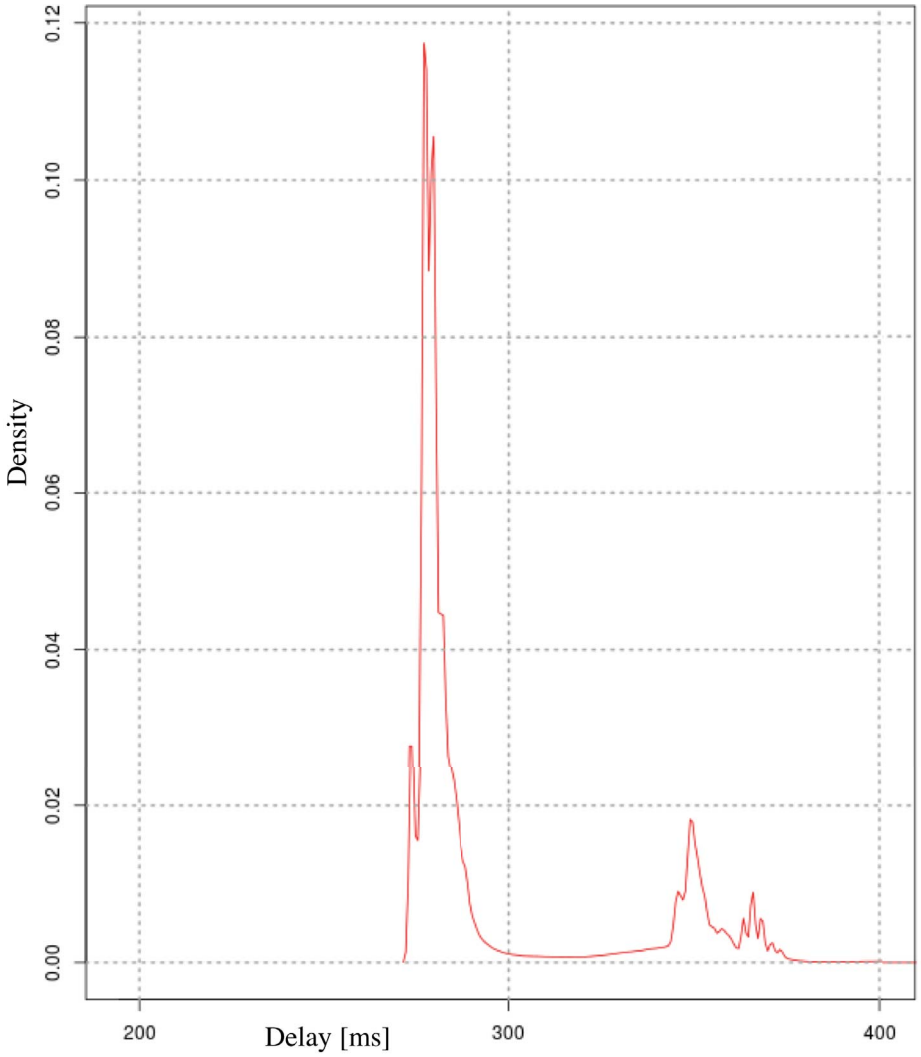

Abbildung 10.8: Dichtefunktion der Umlaufzeitmessungen für Polycom-Verbindung

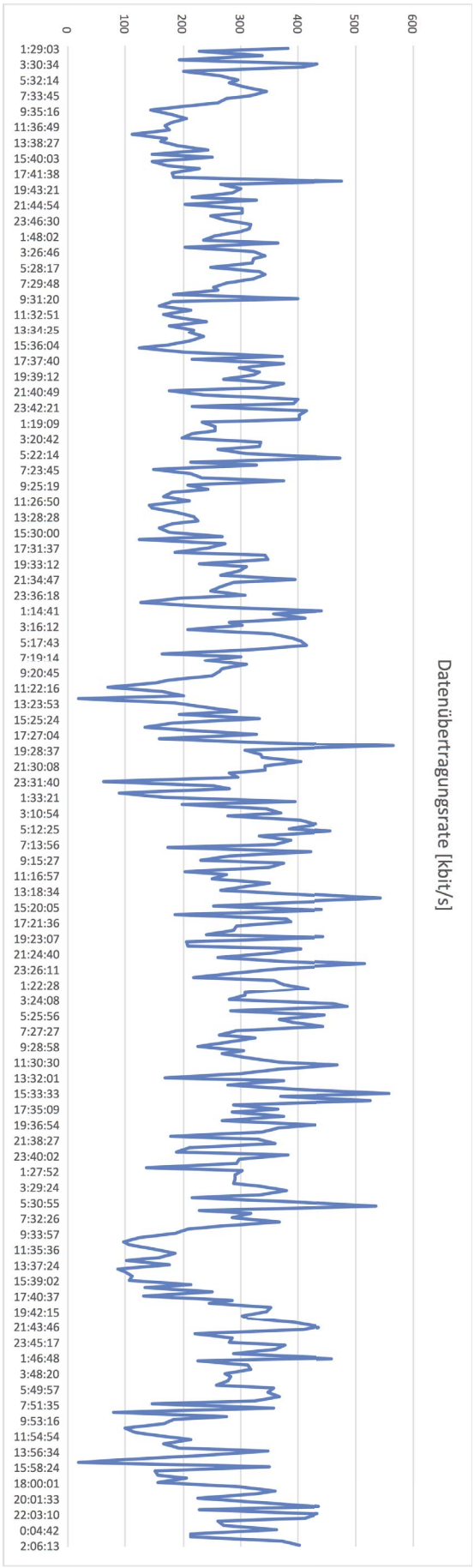

Abbildung 10.9: Liniendiagramm der Datenübertragungsrate für die UKM-Standard-Verbindung

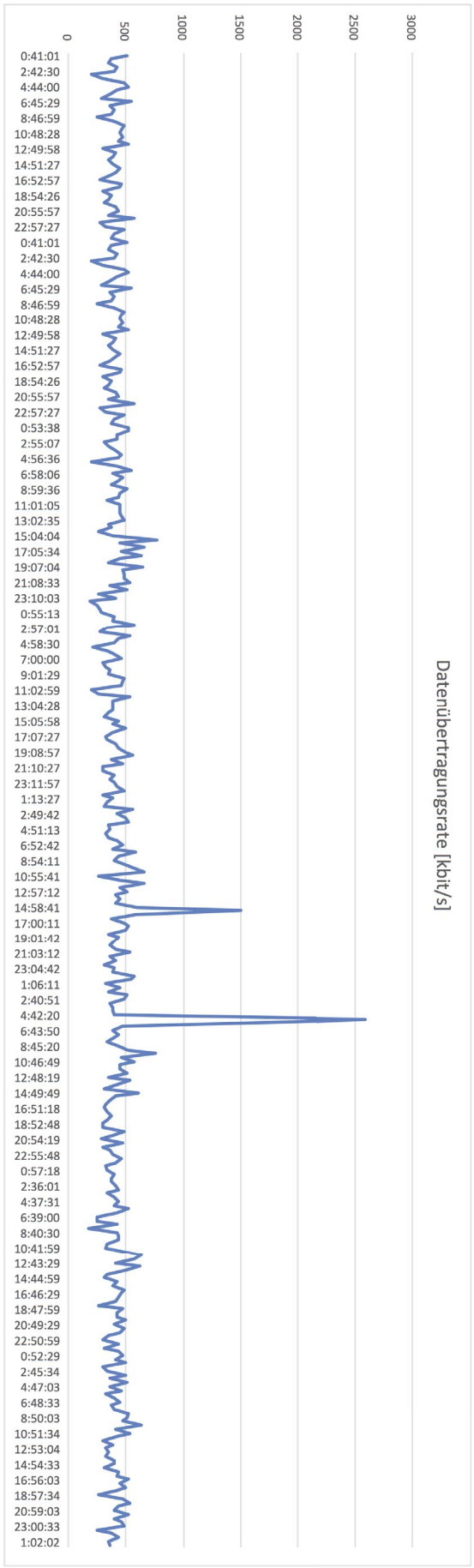

Abbildung 10.10: Liniendiagramm der Datenübertragungsrate für die Polycom-Verbindung

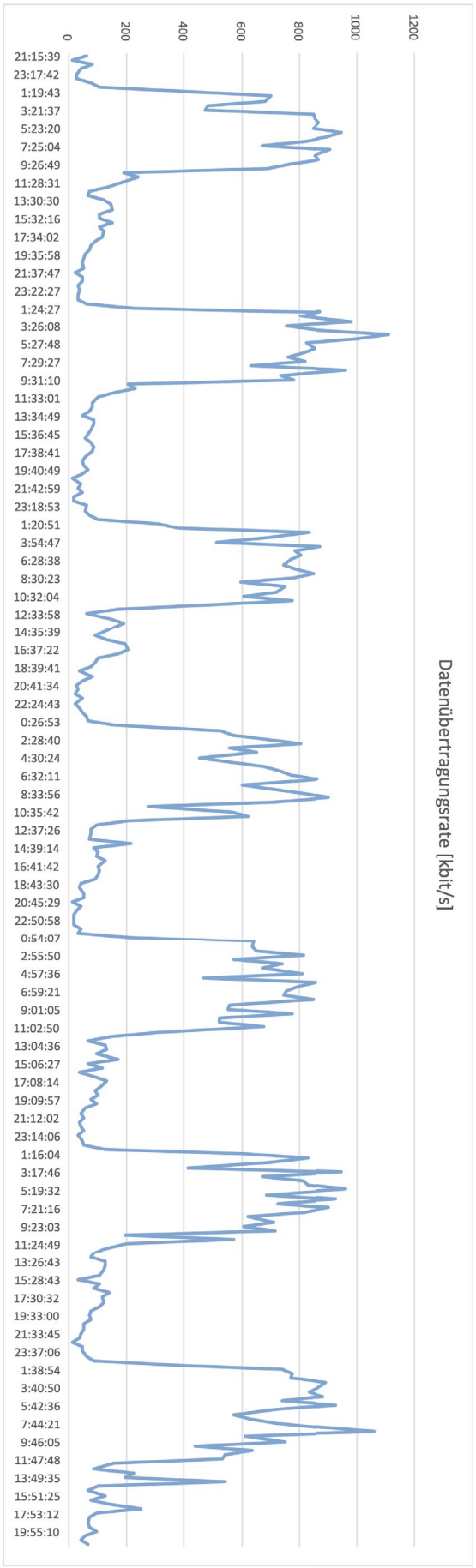

Abbildung 10.11: Liniendiagramm der Datenübertragungsrate für die 3G-Verbindung

10.5 Anhang E – Herleitung der Formel 6.8

$$f_{||}(\mathrm{n}) = f_{||}(\mathrm{n}-1) + R_{pfad,\mathrm{n}} - f_{||}(\mathrm{n}-1)\cdot R_{pfad,\mathrm{n}}$$
$$f_{||}(\mathrm{n\text{-}1}) = f_{||}(\mathrm{n}-2) + R_{pfad,\mathrm{n\text{-}1}} - f_{||}(k-2)\cdot R_{pfad,\mathrm{n\text{-}1}}$$
$$f_{||}(\mathrm{n\text{-}2}) = f_{||}(\mathrm{n}-3) + R_{pfad,\mathrm{n\text{-}2}} - f_{||}(k-3)\cdot R_{pfad,\mathrm{n\text{-}2}}$$

. . . . . .

. . . . . .

. . . . . .

$$f_{||}(2) = f_{||}(1) + R_{pfad,2} - f_{||}(1)\cdot R_{pfad,2}$$
$$f_{||}(1) = f_{||}(0) + R_{pfad,1} - f_{||}(0)\cdot R_{pfad,1}$$

Es folgt eine Addition aller Terme auf der linken Seite und einer Addition aller Terme auf der rechten Seite aller Funktionen:

$$f_{||}(\mathrm{n}) + f_{||}(\mathrm{n\text{-}1}) + f_{||}(\mathrm{n\text{-}2}) + \ldots + f_{||}(2) + f_{||}(1) = f_{||}(\mathrm{n}-1) + R_{pfad,\mathrm{n}} - f_{||}(\mathrm{n}-1)\cdot R_{pfad,\mathrm{n}} + f_{||}(\mathrm{n}-2) + R_{pfad,\mathrm{n\text{-}1}} - f_{||}(\mathrm{n}-2)\cdot R_{pfad,\mathrm{n\text{-}1}} + \ldots + f_{||}(0) + R_{pfad,1} - f_{||}(0)\cdot R_{pfad,1}$$

Durch Kürzen rechts und links erhält man:

$$f_{||}(\mathrm{n}) = f_{||}(0) + R_{pfad,\mathrm{n}} + R_{pfad,\mathrm{n\text{-}1}} + R_{pfad,\mathrm{n\text{-}2}} + \ldots + R_{pfad,2} + R_{pfad,1} - (f_{||}(\mathrm{n}-1)\cdot R_{pfad,\mathrm{n}} + f_{||}(\mathrm{n}-2)\cdot R_{pfad,\mathrm{n\text{-}1}} + f_{||}(k-3)\cdot R_{pfad,\mathrm{n\text{-}2}} + \ldots + f_{||}(2)\cdot R_{pfad,2} + f_{||}(1)\cdot R_{pfad,1})$$

Daraus ergibt sich:

$$f_{||}(n) = f_{||}(0) + \sum_{i=1}^{n} R_{Pfad,i} - \sum_{i=1}^{n} f_{||}(i-1)\cdot R_{Pfad,i}$$
$$f_{||}(n) = f_{||}(0) + \sum_{i=1}^{n} R_{Pfad,i}\cdot\left(1 - f_{||}(i-1)\right)$$

bzw.

$$f_{||}(n) = f_{||}(1) + \sum_{i=2}^{n} R_{Pfad,i}\cdot\left(1 - f_{||}(i-1)\right)$$

10.6 Anhang F – Steuerungsapplikation für rMPTCP

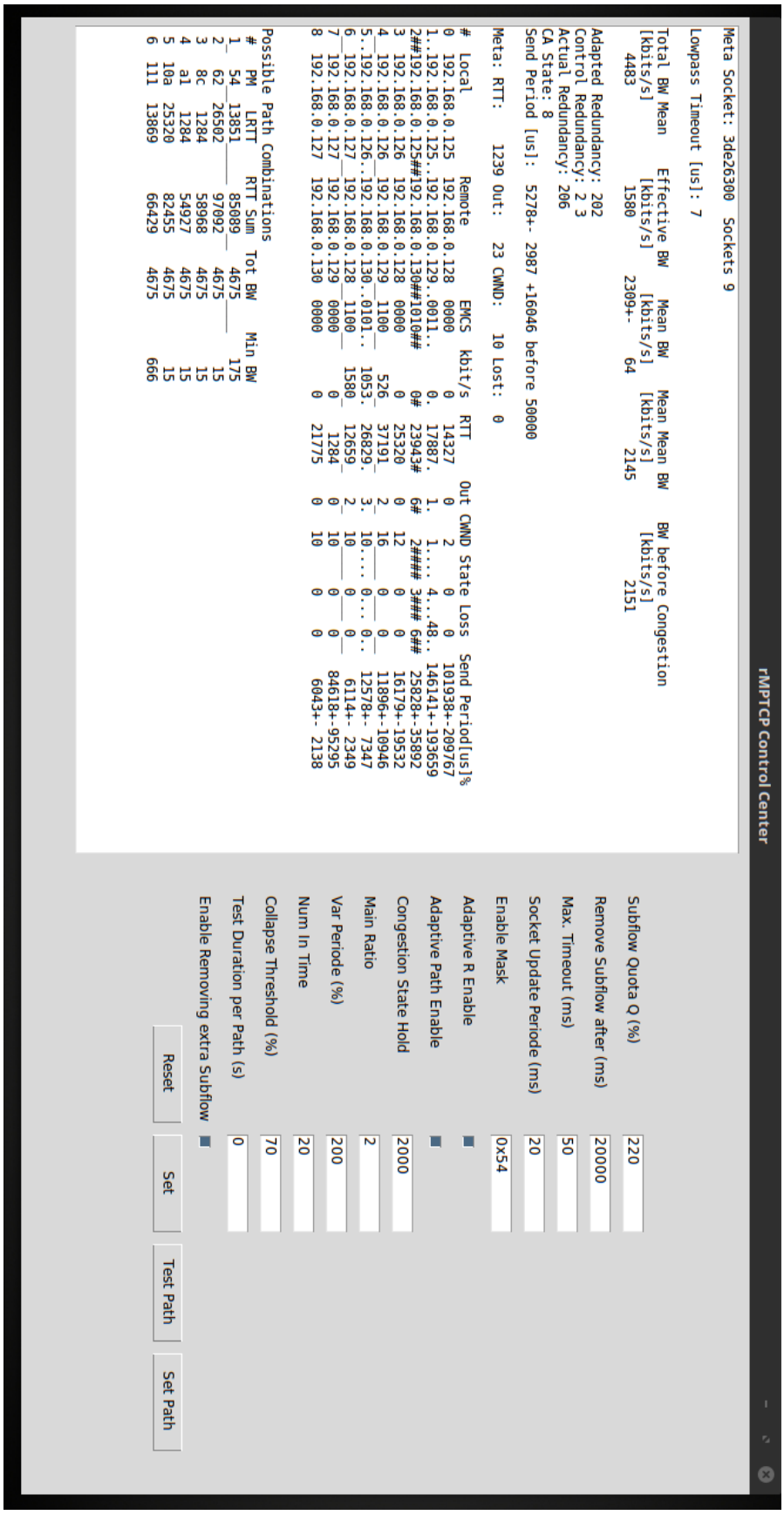

Abbildung 10.12: Screenshot der Steuerungsapplikation von rMPTCP

10.7 Anhang G – Monitoring der Leitungsqualität und Vorhersehbarkeit der Verbindung

Der Verkehr im Netz ist grundsätzlich einer zeitlichen Abhängigkeit unterworfen: Die Leitungsqualität ist in der Nacht, wenn weniger Datenverkehr herrscht, deutlich besser und Segmentverluste sowie ausschweifende Latenzsprünge kommen seltener vor. Am Tage kommt es bei der Leitung zu kritischen Ausfällen und zum Teil zu großen Latenzzeiten, die durch mehrfach wiederholte Sendungen zustande kommen.

Andererseits verhält sich der Netzwerkverkehr durchaus zufällig und es kommt sowohl zu blitzartigem Überlastaufkommen durch stoßweises Datenaufkommen als auch zu suboptimalem Lastausgleich [Xu & Li, 2014]. Eine schlagartige Änderung der Dienstgüte ist das Resultat. Eine allgemeingültige Beurteilung über die zukünftige Verbindungsqualität ist daher schwierig. Sie könnte sich aber möglicherweise für eine konkrete Verbindung wie zwischen UDE und UKM mithilfe einer Korrelation der Durchschnittswerte der Langzeitmessungen abschätzen lassen.

Folgenden Möglichkeiten sind denkbar:

1. Zählen der Störungsereignisse unter rMPTCP
2. Zählen von Ereignissen bezüglich der Dienstgüte-Parameter einer Leitung, wie zum Beispiel die Veränderung der Redundanzquote, RTT oder Datenübertragungsrate
3. Beobachten der Netzwerkteilabschnitte der Netzwerkverbindung bezüglich Lastausgleich-Ereignissen bzw. Routenänderungen.

Unter rMPTCP wurde bereits eine Störungserkennung implementiert, die über bestimmte Zustände verfügt. Eine Zunahme der Störungen bedeutet in erster Linie auch eine Abnahme der Leitungsqualität.

Ereignisse bezüglich der Dienstgüte-Parameter bei einer gleichzeitigen Verwendung von Langzeitmessungen lassen Schlüsse hinsichtlich des Zustands der Leitungsqualität zu. Die Anzahl der Ereignisse erlaubt Rückschlüsse auf die Wahrscheinlichkeit von weiterer Qualitätsverschlechterung. Dies trifft jedoch nur zu, wenn die Leitungsqualität auf der Strecke grundsätzlich denselben Mustern folgt.

Bei den Messungen der Verbindung zwischen UDE und UKM wurden gelegentlich Teilabschnittsänderungen auf der Route festgestellt. Ein Beobachten von Teilabschnitten einer Netzwerkverbindung lässt es zwar nicht zu, ein blitzartiges Datenaufkommen vorherzusehen, jedoch könnten Routenänderungen herangezogen werden, um die Situation zumindest allgemein abzuschätzen. Dies würde jedoch Langzeitmessungen der Routen und ihrer Teilabschnitte erfordern. Eine Erweiterung von rMPTCP in Hinblick auf die Pfad-Diversität, bei der die Teilabschnitte der Routen miteinander verglichen werden, könnte für eine derartige Einschätzung genutzt werden.

Bei einer Abschätzung der Verbindungsqualität könnten unter Nutzung eines oder mehrerer der oben genannten Parameter die Werte als Zustände einer Markov-Kette

erster Ordnung modelliert werden. Unter Einsatz dieser Technik könnte ein weiterer Zustandsautomat entworfen werden, mit der Zielsetzung:

- ein verbessertes Monitoring des Leitungszustands zu ermöglichen
- die in Kapitel 6 verwendeten Parameter besser an die aktuellen Gegebenheiten anzupassen.