PIK, Vol. 10, pp. 113–120, June 2010 · Copyright © by Walter de Gruyter · Berlin · New York. DOI 10.1515/piko.2010.018

Lasttransformation durch Rekonstruktion von Auftragslängen anhand von Paketdaten

Stephan Heckmüller

AG Telekommunikation und
Rechnernetze Universität Hamburg
20146 Hamburg
Deutschland
heckmueller@informatik.uni-hamburg.de

Stephan Heckmüller ist als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe „Telekommunikation und Rechnernetze“ des Departments Informatik der Universität Hamburg tätig. Seine Interessen umfassen Lastmodellierung, Leistungsbewertung von Rechnernetzen sowie Dienstgüte in drahtlosen Netzen. Er studierte Informatik an der Universität Hamburg und schloss sein Studium im Jahr 2006 ab.

Gerhard Münz

Lehrstuhl für Netzarchitekturen und
Netzdienste
Technische Universität München
85748 Garching bei München
Deutschland
muenz@net.in.tum.de

Gerhard Münz ist wissenschaftlicher Mitarbeiter am Lehrstuhl „Netzarchitekturen und Netzdienste“ der Technischen Universität München und war zuvor ebenfalls als wissenschaftlicher Mitarbeiter am Lehrstuhl „Rechnernetze und Internet“ der Universität Tübingen tätig. Seine Forschungsarbeit beschäftigt sich mit passiven Verkehrsmesstechniken sowie der Analyse der gesammelten Verkehrsdaten mit dem Ziel der Anomalieerkennung und der Verkehrsklassifizierung. Darüber hinaus ist Gerhard Münz in der Internet-Standardisierung (IETF) aktiv. Sein Studium der Elektrotechnik absolvierte er im Jahr 2003 an der Universität Stuttgart und an der Ecole Nationale Supérieure des Télécommunications in Paris.

Lothar Braun

Lehrstuhl für Netzarchitekturen und
Netzdienste
Technische Universität München
85748 Garching bei München
Deutschland
braun@net.in.tum.de

Lothar Braun ist seit 2008 wissenschaftlicher Mitarbeiter am Lehrstuhl „Netzarchitekturen und Netzdienste“ der Technischen Universität München. Er studierte Informatik an der Universität Tübingen und schloss sein Studium im Jahr 2008 ab. Im Rahmen seiner Forschungstätigkeit beschäftigt er sich mit passiven Verkehrsmessungen mit dem Ziel der Verkehrsklassifikation und der Erkennung von Malware und Botnetzen.

Aaron Kunde

AG Telekommunikation und
Rechnernetze
Universität Hamburg
20146 Hamburg
Deutschland
2kunde@informatik.uni-hamburg.de

Aaron Kunde hat bis 2009 an der Universität Hamburg Informatik mit Abschluss Diplom studiert. Der Schwerpunkt lag dabei auf technischen Aspekten von Informatiksystemen mit besonderem Fokus auf Rechnernetze und Kommunikationsprotokolle. Zurzeit ist er tätig als Angestellter bei einer IT-Support-Firma.

Bernd E. Wolfinger

AG Telekommunikation und
Rechnernetze
Universität Hamburg
20146 Hamburg
Deutschland
wolfinger@informatik.uni-hamburg.de

Prof. Dr. Bernd E. Wolfinger studierte Mathematik an der Universität Karlsruhe sowie an der Université Claude-Bernard, Lyon/F. von 1970 bis 1975 (Abschlüsse: Dipl.-Math., Maîtrise). Als wissenschaftlicher Mitarbeiter war er anschließend bis 1980 im Kernforschungszentrum Karlsruhe tätig und promovierte 1979 zum Dr. rer. nat. an der Fakultät für Informatik (Univ. Karlsruhe), wo er 1981 auch eine Hochschulassistentur übernahm. Im Oktober 1981 folgte er dann einem Ruf an die Universität Hamburg (Fachbereich Informatik), wo er seither u. a. die Gebiete Rechnernetze, Rechnerarchitektur, Betriebssysteme, (technische) Medienkommunikation und Systembewertung in Forschung und Lehre vertritt und seit 1996 als Leiter der Arbeitsgruppe Telekommunikation und Rechnernetze fungiert.

Georg Carle

Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
85748 Garching bei München
Deutschland
carle@net.in.tum.de

Prof. Dr.-Ing. Georg Carle ist Inhaber des Lehrstuhls „Netzarchitekturen und Netzdienste“ der Technischen Universität München. Er studierte Elektrotechnik an der Universität Stuttgart und der Ecole Nationale Supérieure des Télécommunications in Paris und erwarb an der Brunel University in London einen Master of Science in Digital Systems. Am Institut für Telematik der Universität Karlsruhe promovierte er 1996 als Stipendiat des Graduiertenkollegs „Beherrschbarkeit komplexer Systeme“. 1997 war er mit einem Stipendium der Europäischen Gemeinschaft am Institut Eurécom in Sophia Antipolis tätig. Beim Fraunhofer Institut für Offene Kommunikationssysteme (FOKUS) in Berlin leitete er das Kompetenzzentrum ‚Global Networking‘, von wo aus er im Dezember 2002 an die Universität Tübingen auf den neu geschaffenen Lehrstuhl “Rechnernetze und Internet“ wechselte.

Zusammenfassung

Die Analyse von Verkehrsmessdaten erfolgt heutzutage meist auf der Basis von Statistiken über das Verkehrsaufkommen oder die Verkehrszusammensetzung sowie aufgrund von Eigenschaften einzelner Verkehrsströme oder Pakete. Außer Acht gelassen wird dabei häufig, dass der gemessene Verkehr das Ergebnis einer Interaktion oder eines Datenaustausches auf Anwendungsschicht ist. Dabei ist in vielen Fällen nicht die Analyse des Verkehrs von eigentlichem Interesse, sondern die Untersuchung des Zustands oder des Verhaltens der Anwendung.

Durch Modellierung und Transformation von Lasten ist es möglich, den Zusammenhang zwischen Ankunftsprozessen auf der Anwendungsschicht und den resultierenden Ankunftsprozessen auf der Vermittlungsschicht zu beschreiben. Der vorliegende Beitrag beschäftigt sich mit der Umkehrung dieser Transformation und ihrer praktischen Umsetzung bei der Interpretation von Verkehrsmessungen. Ziel ist es, anhand von gemessenen Paketströmen auf Eigenschaften der ursprünglichen Auftragsströme auf Anwendungsschicht schließen zu können. Dazu werden insbesondere Methoden zur Rekonstruktion von Längeneigenschaften nach der Segmentierung bzw. Fragmentierung von Aufträgen vorgestellt und bewertet.

1 Einleitung

Über heutige Rechnernetze wird eine immer größere Vielzahl von unterschiedlichen Anwendungen und Diensten abgewickelt. Das dadurch verursachte Verkehrsaufkommen ist für die betroffenen Netzbetreiber als Überlagerung verschiedener Verkehrsströme zwischen kommunizierenden Endsystemen sichtbar. So wird die Messung dieser Verkehrsströme beispielsweise zu Abrechnungszwecken genutzt oder um angesichts dynamischer Veränderungen im Verkehr die richtigen Netzmanagemententscheidungen treffen zu können. In letzter Zeit wird die Verkehrsmessung zudem zunehmend zur schnellen Erkennung von Störungen, Anomalien oder auch bösartigem Verkehr eingesetzt. Einen guten Überblick über verschiedene Verkehrsmessmethoden und deren Anwendungen bietet z. B. Ziviani [1].

Heutige Analyseverfahren beruhen auf den Verkehrsmessdaten und daraus abgeleiteten Kenngrößen und Statistiken, die für einzelne oder aggregierte Verkehrsströme erhoben werden. Durch eine Betrachtung der Verkehrsströme können aber nur bedingt Aussagen über den Zustand und das Verhalten der Anwendung gemacht werden. Insbesondere kann anhand der Verkehrsmessdaten nicht direkt darauf geschlossen werden, in welchen zeitlichen Abständen Datenblöcke senderseitig von der Anwendung zum Versand an die Transportschicht gegeben wurden und wie groß diese Datenblöcke waren. Solche Kenntnisse sind aber relevant, um beispielsweise Leistungsbewertungen und Lastprognosen vornehmen zu können oder Verkehrsströme einem bestimmten Anwendungstyp zuordnen zu können.

Abschnitt 2 gibt eine kurze Einführung in die Modellierung von Lasttransformationen, mit denen ein Ankunftsstrom an einer Schnittstelle im System oder Netzwerk auf einen Ankunftsstrom an einer nachfolgenden (tieferliegenden) Schnittstelle abgebildet werden kann. Insbesondere lässt sich durch die Lasttransformation der Zusammenhang zwischen dem Auftragsstrom, den die Anwendungsschicht an die Transportschicht übergibt, und dem daraus resultierenden Paketstrom auf Vermittlungsschicht beschreiben.

Die Abbildung von solchen Auftragsströmen auf Paketströme wurde in vorangegangen Arbeiten bereits intensiv untersucht [2,3]. Darüber hinaus wird die Abbildung von Ankunfts- auf Abgangsprozesse insbesondere im Kontext der Warteschlangentheorie untersucht (siehe z. B. [4]). In diesem Beitrag betrachten wir nun die umgekehrte Richtung, um anhand von Messungen von Paketströmen Aussagen über die ursächlichen Auftragsströme machen zu können. Im Speziellen geht es darum, aus den Verkehrsmessdaten die Auftragslängen zurückzugewinnen. Dies ist deshalb notwendig, da längere Aufträge durch Segmentierung und Fragmentierung auf mehrere Pakete unterteilt werden, wodurch die Auftragslängen im Paketstrom nicht mehr direkt gemessen werden können. In Abschnitt 3 gehen wir auf dieses Problem näher ein und stellen zwei Verfahren zur Rekonstruktion der Auftragslängen vor. Das erste Verfahren lässt sich weitgehend unabhängig von den verwendeten Transport- und Vermittlungsprotokollen einsetzen, während das zweite Verfahren auf spezielle Eigenschaften von TCP zurückgreift und nur für TCP-Verkehr verwendet werden kann. Die Rekonstruktion des Ankunftsprozesses auf Anwendungsebene ist ebenfalls Thema von [5,6]. In diesen Arbeiten werden der Wechsel der Übertragungsrichtung sowie ein minimaler zeitlicher Abstand zwischen den beobachteten Paketen als Kriterien für die Auftragslängenrekonstruktion verwendet. Im vorliegenden Arti-

kel werden hierzu die Paketlängen und das TCP-PUSH-Flag betrachtet.

Die Rekonstruktionsverfahren wurden in ersten Experimenten mit Web-Verkehr untersucht. In Abschnitt 4 stellen wir die Experimente vor und diskutieren die Ergebnisse. Abschließend wird in Abschnitt 5 ein Fazit gezogen und ein Ausblick auf Verbesserungs- und praktische Anwendungsmöglichkeiten gegeben.

2 Lasttransformation und ihre Invertierung

In diesem Abschnitt erläutern wir zunächst, wie verschiedene Verarbeitungsvorgänge in Rechnernetzen als Lasttransformationen beschrieben werden können. Daraufhin erfolgt die Beschreibung der im vorliegenden Beitrag behandelten Rekonstruktion von Auftragslängen als inverse Transformation. Um (inverse) Transformationen formal beschreiben zu können, wird Last wie in Definition 1 [7] dargestellt definiert.

Definition. Die *Last* $L = L(E, S, IF, T)$ wird definiert als eine Sequenz von Aufträgen, die während des Beobachtungsintervalls T an das Bediensystem S durch seine Umgebung E übergeben werden. Die Aufträge werden über die Schnittstelle IF übergeben, welche das Bediensystem von seiner Umgebung trennt.

Die Last kann somit durch eine Sequenz von Aufträgen einer Auftragsmenge A, die während des betrachteten Zeitintervalls T eintreffen, beschrieben werden. Für wohldefinierte Lasten sei der Ankunftsprozess definiert als Tupel aus Ankunftszeitpunkten t_i und den Aufträgen a_i:

$$\{(a_i, t_i) \mid a_i \in A, t_1 \leq t_2 \leq \cdots \leq t_N, t_{1,\ldots,N} \in T\}. \quad (1)$$

Einzelne Aufträge können hierbei beispielsweise Datenübertragungs- oder Verbindungsaufbauwünsche repräsentieren und sind Ankunftszeitpunkten (bezogen auf die entsprechende Schnittstelle) zugeordnet. Um eine detaillierte Spezifikation der Aufträge zu ermöglichen, verwenden wir typisierte Aufträge sowie eine typabhängige Menge von Auftragsattributen. Aufbauend auf Definition 1 werden für die folgenden Untersuchungen zwei Klassen von Transformationen definiert, die zum einen die Aufträge und zum anderen deren zeitliche Eigenschaften betreffen. Hierbei erfolgt die Lasttransformation von Primär- zu Sekundärlast.

1. Wir definieren eine Auftragstransformation als Abbildung T_A, welche eine Sequenz von Primärlastaufträgen $A^p = (a_1^p, \ldots, a_N^p)$ auf eine Sequenz von Sekundärlastaufträgen $A^s = (a_1^s, \ldots, a_K^s)$ für eine gegebene Lasttransformation abbildet.

$$T_A : A^p \rightarrow A^s.$$

2. Die Transformation des zeitlichen Verhaltens sei als Abbildung der Ankunftszeitpunkte der Primärlast $T^p = (t_1^p, \ldots, t_N^p)$ auf die Ankunftszeitpunkte der Sekundärlast $T^s = (t_1^s, \ldots, t_K^s)$ definiert.

$$T_T : T^p \rightarrow T^s.$$

Hierauf aufbauend seien inverse Transformationen definiert als T_A^{-1} bzw. T_T^{-1}, wobei nicht davon ausgegangen werden kann, dass diese eindeutig sind.

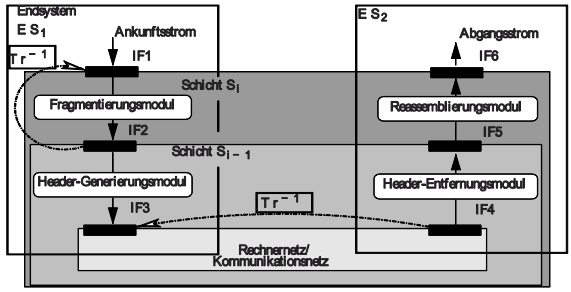

Abbildung 1 Paketübertragung als Sequenz von Transformationen.

Die beschriebenen Vorgänge sind in Abbildung 1 schematisch dargestellt: Durch Fragmentierung, Header-Generierung und Verzögerungen werden sowohl Paketeigenschaften als auch die zeitliche Abfolge verändert. Die so hervorgerufene Veränderung der Lastcharakteristiken bezeichnen wir als *Lasttransformation* von Primär- zu Sekundärlast. Um ein möglichst exaktes Bild der Charakteristiken der untransformierten Last zu erhalten, gilt es somit, die vorgenommenen Lasttransformationen zu invertieren. Dies wird in Abbildung 1 durch die mit Tr^{-1} überschriebenen Pfeile symbolisiert.

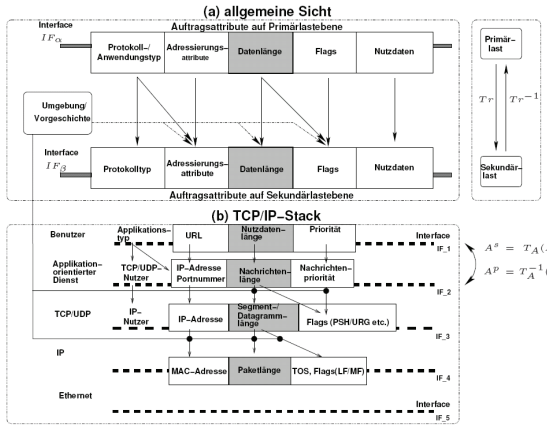

Abbildung 2 (Inverse) Transformation von Auftragsattributen im TCP/IP-Protokollstapel.

Für Lasten, die als *Batch Markovian Arrival Processes* beschrieben werden können, konnten eine Reihe von Transformationsvorgängen (z. B. die Auswirkungen von Paketverlusten, von Fragmentierungsvorgängen und von Token-Bucket-Regulierung) als Abbildungen auf solchen Prozessen erfolgreich modelliert werden [2,3]. Darüber hinaus soll nun die inverse Lasttransformation systematisch betrachtet werden. Die Invertierung von Lasttransformationsprozessen kann hierbei zur Rekonstruktion von Eigenschaften der an ein nicht direkt beobachtbares System übergebenen Aufträge genutzt werden.

Im Folgenden soll zunächst aufgezeigt werden, welche Typen von inversen Transformationen in Rechnernetzen existieren und welche Informationen zu ihrer Durchführung benötigt werden. Hierauf aufbauend werden in Abschnitt 3 inverse

Transformationen zur Rekonstuktion von Lasteigenschaften auf der Vermittlungs- und Transportschicht vorgeschlagen.

Wie bei der Lasttransformation kann zunächst zwischen einer inversen Transformation des zeitlichen Verhaltens und von Attributen unterschieden werden, wobei in vielen Fällen Zusammenhänge zwischen den beiden Transformationstypen bestehen. Während im Kontext des zeitlichen Verhaltens hauptsächlich eine Veränderung der Zwischenankunftszeiten der Aufträge von Interesse ist, ergeben sich in Bezug auf die Auftragsattribute eine Vielzahl von möglichen Transformationen. Eine Möglichkeit besteht darin, die verschiedenen Transformationen anhand der betroffenen Attribute zu unterschieden, wie es im oberen Teil von Abbildung 2 dargestellt ist. Es ist erkennbar, dass in der Regel nicht von einer 1:1-Abbildung der Attribute ausgegangen werden kann. Es sind sowohl n:1- als auch 1:n-Abbildungen der Attribute üblich, wobei sich die Abbildungen auch auf Attribute mehrerer Aufträge beziehen können. Zusätzlich zum aktuellen Auftrag beeinflusst in vielen Fällen der Zustand des Netzes und des Endsystems sowie die Historie (in Form von vorangegangenen Aufträgen) die Abbildung. Dies wird im unteren Teil von Abbildung 2 ersichtlich, in dem für einige typische Attribute der einzelnen Schichten beispielhaft die dazugehörigen Transformationen schematisch dargestellt sind. Hier wird deutlich, dass der Zustand der Umgebung häufig eine wichtige Rolle bei der Abbildung von Auftragsattributen spielt.

Im Falle des Segmentierungsprozesses im TCP-Modul erfolgt die Abbildung eines Längenattributes auf mehrere Attribute an der nachfolgenden Schnittstelle. Dabei wird die Vorgeschichte in Form von im Puffer verbliebenen Daten miteinbezogen. Abhängig davon, ob der Puffer mit dem Versenden des Paketes geleert werden kann, erfolgt neben der Abbildung des Längenattributs des Auftrags auf das Längenattribut des TCP-Segments auch die Abbildung auf das PUSH-Flag [8]. Dies impliziert, dass die Rekonstruktion von Attributen an höheren Schnittstellen ausgehend von mehreren Attributen an den unteren Schnittstellen möglich sein kann.

3 Rekonstruktion von Auftragslängen

Auf Anwendungsschicht verläuft der Datenaustausch zwischen zwei Rechnern üblicherweise nicht zeichen- oder byteweise sondern in Einheiten von größeren Datenblöcken. Für einen solchen Datenblock verwenden wir im Folgenden den Begriff *Auftrag* (hier im speziellen Sinne eines Anwendungsauftrags), wie er in Abschnitt 2 im Zusammenhang mit dem Modell der Lasttransformation eingeführt wurde. Die Paketlängen, die an tieferen Schnittstellen des Protokollstapels beobachtet werden können, entsprechen typischerweise nicht mehr den Auftragslängen auf Anwendungsschicht. Dies hängt zum einen mit der Kontrollinformation zusammen, die in Abhängigkeit von den verwendeten Protokollen hinzugefügt wird. Zum anderen ist in paketvermittelten Netzen durch die Protokolle auf tieferen Schichten im Allgemeinen eine maximale Paket- oder Rahmenlänge vorgegeben, die nicht überschritten werden darf.

Die Paketlängenbeschränkung auf der Vermittlungsschicht wird durch die *Maximum Transmission Unit* (MTU) angegeben. Die MTU gibt die maximale Länge der *Protocol Data Unit* (PDU) auf der Vermittlungsschicht an, was der Paketlänge inklusive der Kontrollinformation der Vermittlungsschicht entspricht. Nach Abzug der Byteanzahl, die für die Kontrollinformation benötigt wird, ergibt sich eine maximale Nutzdatenlänge, die je Paket übertragen werden kann, also die maximale Länge der *Service Data Unit* (SDU). Falls die Länge eines Auftrags die maximale Nutzdatenlänge überschreitet, muss der Auftrag in kleinere Blöcke unterteilt werden.

Beim Internet-Protokoll (IP) wird die Unterteilung durch die IP-Fragmentierung realisiert, wobei die Nutzdaten stets nach einem Vielfachen von 8 Byte geteilt werden [9]. Die IP-Fragmentierung auf Senderseite kommt zum Einsatz, wenn ein Auftrag mit dem Transportprotokoll UDP versendet wird und das UDP-Datagramm die maximale IP-Nutzdatenlänge überschreitet. Wenn M die maximale Nutzdatenlänge eines IP-Paketes ist, erhält man so für ein UDP-Datagramm der Länge $a > M$ nach der Unterteilung

$$\left\lfloor \frac{a}{8\lfloor \frac{M}{8} \rfloor} \right\rfloor$$

Pakete maximaler Länge und gegebenenfalls ein weiteres kürzeres Paket, das die übrigen Bytes des UDP-Datagramms enthält.

Bei Verwendung von TCP wird durch Segmentierung auf Transportschicht eine senderseitige IP-Fragmentierung vermieden. Die maximale Nutzdatenlänge eines TCP-Segments wird *Maximum Segment Size* (MSS) genannt. Die MSS wird so gewählt, dass nach Hinzunahme der IP/TCP-Kontrollinformation die MTU nicht überschritten wird [10]. Ein Auftrag, dessen Länge die MSS überschreitet, wird in mehrere Segmente der Länge MSS und gegebenenfalls ein zusätzliches kürzeres Segment unterteilt. Die Unterteilung ist hier nicht an 8-Byte-Grenzen gebunden. Durch den Algorithmus von Clark [11] wird verhindert, dass der Empfänger ein Empfangsfenster unterhalb der MSS anbietet und dadurch den Versand eines kleineren Segments provoziert. TCP-Segmentierung und IP-Fragmentierung von UDP-Datagrammen werden in Abbildung 3 veranschaulicht.

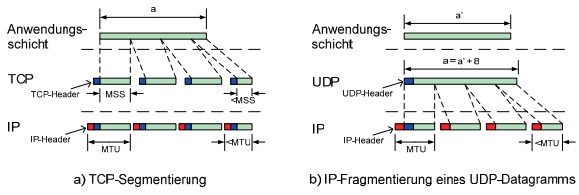

Abbildung 3 Unterteilung von Aufträgen.

Die Kenntnis der Auftragslängen ist notwendig, um die Last auf Anwendungsschicht modellieren zu können. Zudem ermöglichen die Auftragslängen Rückschlüsse auf das Benutzer- bzw. Anwendungsverhalten, die durch Betrachtung der Paketlängen alleine unter Umständen nicht möglich wären. Insbesondere gleichen sich die Paketlängenverteilungen für ganz unterschiedliche Auftragslängenverteilungen sehr stark, wenn die mittlere Auftragslänge größer als M ist. Dies wird im folgenden Unterabschnitt 3.1 illustriert.

Danach stellen wir in den Unterabschnitten 3.2 und 3.3 zwei Möglichkeiten vor, wie sich Auftragslängen anhand der im Rechnernetz beobachtbaren Pakete rekonstruieren lassen. Als Voraussetzung für beide Verfahren ist es erforderlich, dass die Pakete einem Auftragsstrom zugeordnet werden können. Im Falle von UDP und TCP liefert die Kombination aus IP-Adressen und Portnummern einen Schlüssel, mit dem eine solche Zuordnung vorgenommen werden kann. Ein Multiplexen verschiedener Auftragsströme auf höheren Schichten kann auf diese Weise nicht erkannt werden, so dass die Überlagerung der Auftragsströme in diesem Fall als ein gemeinsamer Auftragsstrom angesehen werden muss.

3.1 Fragmentlängen verschiedener Auftragslängenverteilungen

Betrachtet man die Auswirkungen der Fragmentierung bzw. Segmentierung auf die Längen der resultierenden Fragmente, so lässt sich beobachten, dass die Fragmentlängenverteilungen transformierter Auftragsströme keine großen Unterschiede aufweisen, wenn ein größerer Anteil der Auftragslängen die maximale Nutzdatenlänge M übersteigt. Dies gilt selbst dann, wenn die Auftragslängenverteilungen sehr unterschiedlich sind (vgl. [2,12]), wie im Folgenden anhand dreier Verteilungstypen illustriert wird.

Für die Verteilung der Auftragslängen werden die folgenden drei Verteilungen angenommen: eine Normalverteilung (N), eine negative Exponentialverteilung (E) und eine Pareto-Verteilung (P). Die Standardabweichungen seien in Abhängigkeit vom Erwartungswert j gegeben durch $\sigma_N = 0.25j$, $\sigma_E = j$ und $\sigma_p \approx 2.2j$. Abbildung 4 zeigt links die Verteilungsfunktionen für $j = 6000$. Unterteilt man die Aufträge in Fragmente der maximalen Länge $M = 1500$, so erhält man die in Abbildung 4 rechts gezeigten Kurven für die Verteilungsfunktionen der Fragmentlängen. Die Verteilungsfunktion der Fragmentlängen ist nahezu identisch, während sich die Verteilungsfunktionen der Auftragslängen deutlich unterschieden. Vergleichbare Resultate wurden sowohl für andere Werte von j als auch für andere Verteilungstypen erzielt.

Um verschiedene Auftragslängenverteilungen unterscheiden zu können, ist also eine Rekonstruktion erforderlich. In den beiden folgenden Unterabschnitten werden dafür zwei Verfahren vorgestellt.

3.2 Rekonstruktion der Auftragslängen durch Messung von Paketlängen

Um eine allgemeine Anwendbarkeit zu gewährleisten, verwendet das hier vorgestellte Verfahren zur Rekonstruktion der Auftragslängen nur die Längen der in den Paketen enthaltenen Nutzdaten. Das Verfahren ist damit weitgehend unabhängig vom verwendeten Protokoll. Neben der Länge der Nutzdaten ist auch die Reihenfolge maßgeblich, in der die Pakete beobachtet wurden. Die beobachtete Reihenfolge der Pakete muss dabei nicht unbedingt der Reihenfolge entsprechen, mit der die Pakete ursprünglich gesendet wurden. Welche Fehler sich durch solche Reihenfolgevertauschungen ergeben können, wird am Ende dieses Unterabschnitts erläutert.

Als weitere Voraussetzung für das Verfahren muss die senderseitige Begrenzung Paketlänge und die zugehörige maximale Nutzdatenlänge M bekannt sein, bei TCP also die MSS. Ist diese Begrenzung nicht bekannt, kann sie aus der größten beobachteten Nutzdatenlänge geschlossen werden. Wie bereits erwähnt, ergibt sich die maximale Nutzdatenlänge aus der Differenz zwischen MTU und der Länge der Kontrollinformation der Vermittlungsschicht und der Transportschicht.

Die Längen der zu einem Auftragsstrom gehörenden Pakete seien $p_1, p_2, \ldots$, die zugehörigen Nutzdatenlängen $l_1, l_2, \ldots$. Das Rekonstruktionsverfahren basiert nun darauf, dass die Grenze zwischen zwei aufeinanderfolgenden Aufträgen im Paketstrom durch ein Paket sichtbar wird, das die maximale Nutzdatenlänge nicht ausnutzt, d. h. $l_i < M$. Bei unveränderter Länge der Kontrollinformation ist dann die Paketlänge $p_i < MTU$. Die Grenze kann allerdings dann nicht erkannt werden, wenn die Auftragslänge genau einem Vielfachen der maximalen Nutzdatenlänge M entspricht und somit am Ende des Auftrags kein Paket nicht-maximaler Länge auftritt. Seien nun i_j ($j = 1, 2, \ldots$) die Positionen der Pakete nicht-maximaler Nutzdatenlänge im Paketstrom ($l_{i_j} < M$). Die Länge ($\hat{a}_j$) des j-ten Auftrags a_j lässt sich dann durch Aufsummieren der Nutzdatenlängen der Pakete $i_{j-1} + 1$ bis i_j rekonstruieren:

$$\hat{a}_j = \sum_{k=i_{j-1}+1}^{i_j} l_k. \qquad (2)$$

Diese Formel funktioniert auch für die Länge des ersten Auftrags $\hat{a}_j$, wenn man setzt $i_0 = 0$.

Bei Protokollen, die Kontrollinformation in Paketen ohne Nutzdaten austauschen (z. B. TCP), können sich durch die Rekonstruktion zusätzliche Aufträge der Länge $\hat{a}_j = 0$ ergeben. Um diesem Problem zu begegnen, kann man entweder im Nachhinein alle rekonstruierten Aufträge mit Länge 0 entfernen oder im Vorhinein den untersuchten Paketstrom auf Pakete beschränken, die Nutzdaten enthalten (d. h. $l_j > 0$).

Bei der Rekonstruktion der Auftragslängen können aus verschiedenen Gründen Fehler auftreten. Wie bereits erwähnt, wird eine Auftragsgrenze nicht erkannt, wenn die Auftragslänge a ein Vielfaches von M ist. Weitere Fehler ergeben sich dadurch, dass der beobachtete Paketstrom nicht unbedingt dem gesendeten Paketstrom entsprechen muss. Reihenfolgevertauschungen von Paketen können zu Fehlern führen, wenn sie über Auftragsgrenzen hinweg erfolgen. Pakete, die auf dem Weg vom Sender zum Messpunkt verloren gegangen sind oder aus einem anderen Grund nicht beobachtet wurden, führen zu kleineren Auftragslängen oder einer kleineren Auftragsanzahl.

Schließlich können verschiedene Transportprotokollmechanismen dazu führen, dass die rekonstruierte Auftragsfolge nicht der Auftragsfolge auf Anwendungsschicht entspricht. Offensichtlich ist dies dann der Fall, wenn verloren gegangene Pakete wiederholt übertragen werden. Bei TCP gibt es darüber hinaus noch verschiedene Mechanismen, die die Paketgröße beeinflussen. So bewirkt der Algorithmus von Nagle [13], dass neue Aufträge im TCP-Sendepuffer akkumuliert werden, wenn der Empfang des vorangegangenen Auftrags noch nicht vollständig quittiert wurde. Andererseits

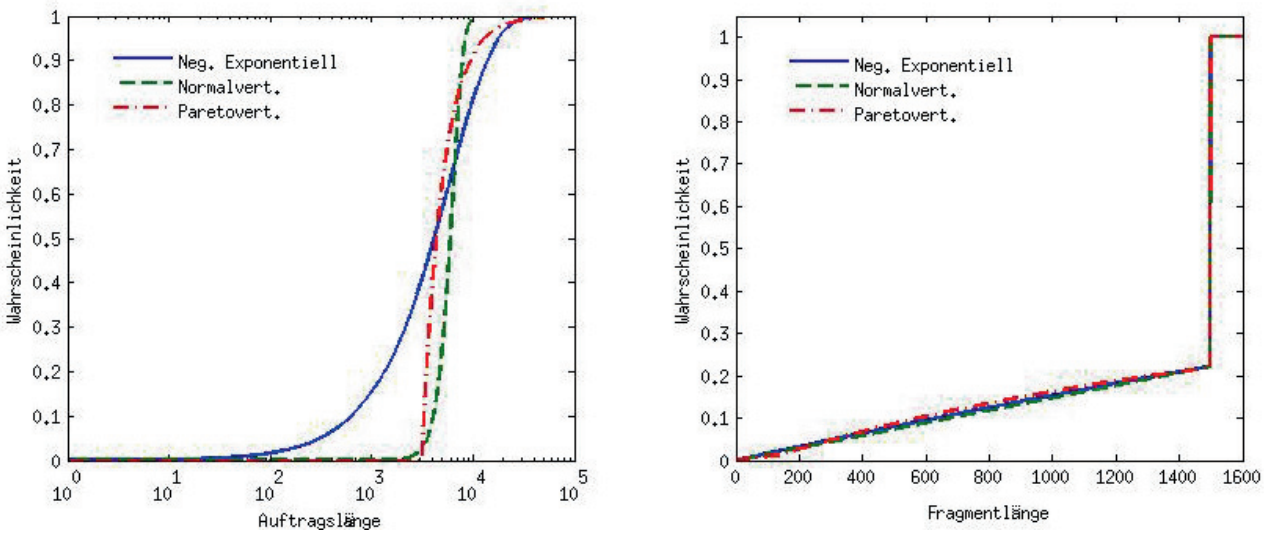

Abbildung 4 Verteilungsfunktion der betrachteten Verteilungen für $j = 6000$ (links) sowie Verteilungsfunktion der Fragmentlängen nach Fragmentierung mit $M = 1500$ (rechts).

kann die TCP-Flusskontrolle dazu führen, dass während des Versandes eines großen Auftrags Segmente kleiner als MSS auftreten, weil das Empfangsfenster ausgeschöpft ist.

Generell nicht zu erkennen sind Akkumulationen und Unterteilungen von Aufträgen, die von einem Protokoll oberhalb der Transportschicht vorgenommen werden. Beispiele hierfür sind das *Pipelining* und das *Chunking* bei HTTP 1.1 [14].

3.3 Rekonstruktion der Auftragslängen durch Verwendung des TCP-PUSH-Flags

In der ursprünglichen Version von TCP ist eine Push-Funktion vorgesehen, mit der die Anwendungsschicht TCP anweisen kann, die übergebenen Daten sofort zu verschicken, auch wenn dadurch möglicherweise ein kurzes Segment entsteht, das mit weiteren Daten aufgefüllt werden könnte. Dem Empfänger wird durch Setzen des PUSH-Flags im TCP-Header signalisiert, dass die empfangenen Daten sofort an die Anwendungsschicht zu übergeben sind und dies dem Sender durch eine Quittierung bestätigt werden soll. Die Push-Funktion findet heute in der Praxis keine Anwendung mehr und wird von vielen TCP-Socket-Implementierungen nicht angeboten. RFC 1122 [15] schreibt für diesen Fall vor, dass die senderseitige TCP-Implementierung das PUSH-Flag dann zu setzen hat, wenn mit dem Versand des TCP-Segments der Sendepuffer geleert wurde. Dies hat zur Folge, dass bei der TCP-Segmentierung eines Auftrages das letzte Segment mit einem PUSH-Flag versehen wird, womit sich die Auftragsgrenzen erkennen lassen.

Die Auftragslängen können, wie im vorherigen Abschnitt beschrieben, durch Aufsummieren der Nutzdatenlängen, die zwischen zwei Auftragsgrenzen beobachtet werden, rekonstruiert werden. Für eine genauere Berechnung kann auf die Sequenznummern im TCP-Header zurückgegriffen werden. Dieses Vorgehen wird im folgenden für den auf dem PUSH-Flag basierenden Algorithmus beschrieben. Im Falle von TCP-Verkehr kann dieses ebenso in Verbindung mit dem Längenkriterium Anwendung finden.

Die Sequenznummer gibt die Position des ersten Bytes der Nutzdaten an, die in einem Segment transportiert werden. Seien $s_1, s_2, \ldots$ die Sequenznummern der versendeten Segmente und $i_1, i_2, \ldots$ die Indizes der Segmente mit gesetztem PUSH-Flag. Setzt man zusätzlich $i_0 = 0$, lässt sich die Länge des j-ten Auftrags allgemein wie folgt rekonstruieren:

$$\hat{a}_j = s_{i_j+1} - s_{i_{j-1}+1}. \tag{3}$$

Wie man sieht, werden in obiger Formel die Sequenznummern der Segmente verwendet, die auf die Segmente mit gesetztem PUSH-Flag folgen.

Die Sequenznummern und die Längen der Nutzdaten l_i, die in den Segmenten transportiert werden, stehen in einer direkten Beziehung:

$$l_i = s_{i+1} - s_i. \tag{4}$$

Hieraus folgt:

$$\hat{a}_j = (s_{i_j} + l_{i_j}) - (s_{i_{j-1}} + l_{i_{j-1}}). \tag{5}$$

Damit diese Formel auch für die Länge des ersten Auftrags funktioniert, wird nun aber die Position des ersten Bytes des ersten Auftrags benötigt. Diese ergibt sich aus der initialen Sequenznummer s_0, die der Sender dem Empfänger während des Verbindungsaufbaus in einem Paket mit gesetztem SYN-Flag mitgeteilt hat. Auf s_0 wird die Länge $l_0 = 1$ addiert, um das SYN-Paket zu bestätigen.

Die Berechnung der Auftragslängen aus den Sequenznummern hat den Vorteil, dass man sich bei der Beobachtung des Paketstroms auf Segmente mit gesetztem SYN- oder PUSH-Flag beschränken kann. Reihenfolgevertauschungen und Übertragungswiederholungen haben keinen Einfluss auf

das Ergebnis, solange die Abfolge der SYN- und PUSH-Pakete gewahrt bleibt.

Fehler können dadurch auftreten, dass das PUSH-Flag auch bei Segmenten gesetzt wird, die nicht das Ende eines Auftrags enthalten. Die TCP-Implementierung von Linux setzt beispielsweise das PUSH-Flag bei der Übertragung einer größeren Datenmenge in regelmäßigen Abständen, um mit alten TCP-Implementierungen kompatibel zu sein, die die Daten erst nach Erhalt des PUSH-Flags an die Anwendungsschicht weitergeben. Des Weiteren kann der in Unterabschnitt 3.2 skizzierte Fall auftreten, dass der Sendepuffer voll ist. In diesem Fall setzt der Sender in dem Segment, das den Puffer vollständig füllt, das PUSH-Flag, um damit eine Quittierung vom Empfänger zu erzwingen. Auch der Nagle-Algorithmus kann wieder eine Akkumulation von Aufträgen im Sendepuffer hervorrufen, die nicht erkannt werden kann. Unterteilungen und Akkumulationen von Aufträgen oberhalb der Transportschicht bleiben ebenfalls unerkannt.

4 Bewertung der Rekonstruktionsalgorithmen

Im diesem Abschnitt untersuchen wir die Genauigkeit der vorgestellten Rekonstruktionsalgorithmen, indem wir die im vorhergehenden Abschnitt vorgestellten Methoden auf aufgezeichneten Webverkehr anwenden. Der Webverkehr wurde durch HTTP-Anfragen (Version 1.1) an die Server *de.wikipedia.org* bzw. *www.debian.org* erzeugt. Die Aufzeichnung des Verkehrs erfolgte empfängerseitig mit *tcpdump*. Als Referenzwerte wurden die Längen der einzelnen HTTP-Nachrichten vom Server aus dem Feld *content-length* der HTTP-Header extrahiert. Um eine möglichst gute Vergleichbarkeit der HTTP-Headerdaten und der Längen der im aufgezeichneten Verkehr enthaltenen HTTP-Aufträge zu gewährleisten, wurde der Browser-Cache deaktiviert.

In den Tabellen 1 und 2 ist eine Übersicht über die Ergebnisse der beiden Experimente dargestellt. In den ersten beiden Zeilen werden jeweils die tatsächliche Auftragsanzahl und die Anzahl der rekonstruierten Aufträge gegenübergestellt. Darunter werden die rekonstruierten Aufträge in solche mit korrekt und falsch rekonstruierten Längen aufgeschlüsselt. Aus den Tabellen ist ersichtlich, dass in beiden Fällen durch die längenbasierte Rekonstruktion sowohl eine höhere Anzahl an korrekten Rekonstruktionen als auch eine geringere Zahl von fehlerhaften Auftragslängen erreicht werden. Wie in Unterabschnitt 3.2 bereits angesprochen, können im Falle der längenbasierten Rekonstruktion zwei Aufträge zusammengefasst werden, falls die Länge des ersten Auftrags genau der maximalen Nutzdatenlänge entspricht. Dieser Fall tritt in den hier zugrunde liegenden Daten einmal auf (*de.wikipedia.org*). Alle anderen fehlerhaft rekonstruierten Auftragslängen kommen dadurch zustande, dass einzelne größere Aufträge als mehrere kleine Aufträge aufgefasst werden.

Diesbezüglich wurden in den Unterabschnitten 3.2 und 3.3 mögliche Fehlerquellen der Rekonstruktionsverfahren diskutiert. Über die tatsächlichen Ursachen der in unseren Experimenten aufgetretenen falschen Rekonstruktionen können wir einige Vermutungen anstellen. So beobachten wir beispielsweise bei *de.wikipedia.org* ein häufiges Auftreten von falsch rekonstruierten Aufträgen der Länge $\hat{a}_j = 4096$. Eine mögliche Ursache hierfür ist, dass der serverseitige Sendepuffer zeitweise auf eine Größe von 4 Kilobyte begrenzt wird. Dadurch entstehen Sequenzen von zwei Paketen maximaler Nutzdatenlänge $M = 1368$ Byte und einem Paket mit 1360 Byte Nutzdaten. Da beim letzten Paket jeweils auch das PUSH-Flag gesetzt ist, führt dieses Phänomen bei beiden Methoden zu fehlerhaften Rekonstruktionsergebnissen. Bei Anwendung der PUSH-Flag-Methode kann ein periodisches Setzen des PUSH-Flags zu zusätzlichen Fehlern führen. Die Häufigkeit dieses Fehlerfalls bei Paketen mit maximaler Nutzdatenlänge ist in Tabelle 2 unter "$l_i = MSS$ mit PUSH-Flag" angegeben. Alle weiteren Fehlerfälle wurden in beiden Tabellen in der Zeile "$l_i < MSS$, $\hat{a}_j \neq 4096$" zusammengefasst. Denkbare Ursachen für diese Fehler sind ein volles Empfangsfenster oder ein ausgeschöpfter Sendepuffer mit von 4096 Byte abweichender Größe.

Tabelle 1 Ergebnisse der längenbasierten Rekonstruktion mit $M = \max(l_i)$.

	de.wikipedia.org	**www.debian.org**
Empfangene Aufträge	696	181
Rekonstruierte Aufträge	1216	183
davon korrekt	425	179
Falsche Rekonstruktionen	791	4
$l_i < MSS$, $\hat{a}_j = 4096$	224	0
$l_i < MSS$, $\hat{a}_j \neq 4096$	567	4

In weiteren hier aus Platzgründen nicht dargestellten Experimenten wurden die Algorithmen auf aufgezeichneten SMTP- bzw. FTP-Verkehr angewendet. Hierbei konnten die Auftragslängen mit Hilfe des längenbasierten Algorithmus mit höherer Genauigkeit rekonstruiert werden als mit Hilfe des PUSH-Flags, wobei in beiden Fällen die Sequenznummern zur Berechnung der Längen verwendet wurden. Dies steht im Einklang mit den Ausführungen in Unterabschnitt 3.3 und den hier dargestellten Rekonstruktionsergebnissen, so dass eine längenbasierte Rekonstruktion geeigneter scheint.

5 Resümee und Ausblick

Im vorliegenden Beitrag wurde unter Anwendung des neuartigen Ansatzes der inversen Lasttransformation untersucht, inwiefern aus paketorientierten Messdaten auf der Vermittlungsschicht Eigenschaften der Ankunftsprozesse auf der Anwendungsschicht rekonstruiert werden können. Eine solche Rekonstruktion ist immer dann von Nutzen, wenn das eigentliche Interesse nicht dem unmittelbar beobachtbaren Paketverkehr gilt, sondern die den Verkehr induzierenden Anwendungen betrachtet werden sollen. Insbesondere wurden Verfahren vorgeschlagen, um ausgehend von den Eigenschaften der gemessenen Pakete die Längen der durch die Anwendung übergebenen Aufträge zu rekonstruieren. Dieses ist notwendig, weil längere Aufträge durch TCP-Segmentierung bzw.

Tabelle 2 Ergebnisse der Rekonstruktionen basierend auf PUSH-Flags.

	de.wikipedia.org	www.debian.org
Empfangene Aufträge	696	181
Rekonstruierte Aufträge	1245	236
davon korrekt	412	139
Falsche Rekonstruktionen	833	97
$l_i < MSS$, $\hat{a}_j = 4096$	224	0
$l_i < MSS$, **mit PUSH-Flag**	29	55
$l_i < MSS$, $\hat{a}_j \neq 4096$	580	42

IP-Fragmentierung in mehrere Pakete aufgeteilt werden können.

Die betrachteten Vorgänge wurden zunächst in das allgemeine Konzept der Lasttransformation eingebettet, welches aufbauend auf der Kenntnis der unmodifizierten Last (Primärlast) und des betrachteten Verarbeitungsvorgangs eine Vorhersage der modifizierten Last (Sekundärlast) erlaubt. Die in der vorliegenden Arbeit untersuchten Rekonstruktionsalgorithmen stellen hierbei eine Umkehrung der im Protokollstapel vorgenommenen Lasttransformationen dar.

Die vorgestellten Rekonstruktionsverfahren wurden auf aufgezeichneten Webverkehr angewendet, wobei sich zeigte, dass insbesondere große Auftragslängen häufig nicht korrekt rekonstruiert und als mehrere kleine Aufträge aufgefasst werden. Solche Abweichungen treten in unterschiedlichem Maße sowohl bei der Betrachtung der Paketlängen wie auch bei der Beobachtung der PUSH-Flags auf und werden durch eine Reihe von TCP-Mechanismen verursacht.

In weiterführenden Arbeiten werden wir die Ursachen der aufgetretenen fehlerhaften Rekonstruktionen genauer untersuchen und versuchen, durch Berücksichtigung zusätzlicher Kriterien die Grenzen zwischen aufeinanderfolgenden Aufträgen noch besser zu erkennen. Des Weiteren sollen die Verfahren mit Hilfe von Messdaten für verschiedene Anwendungen und aus verschiedenen Netzen quantitativ bewertet werden. Anschließend sollen die rekonstruierten Auftragslängen zur Klassifizierung von Verkehrsströmen in verschiedene Anwendungsklassen eingesetzt werden.

Danksagung

Wir danken der Deutschen Forschungsgemeinschaft (DFG) für die Förderung des LUPUS-Projekts (DFG-Geschäftszeichen: CA 595/1-1), in dessen Rahmen die vorgestellte Forschungsarbeit durchgeführt wurde.

Literatur

[1] Ziviani, A.: An Overview of Internet Measurements: Fundamentals, Techniques, and Trends. African Journal of Information and Communication Technology (AJICT), UTSePress 2(1) (March 2006), 39–49.

[2] Heckmüller, S., Wolfinger, B. E.: Using Load Transformations to Predict the Impact of Packet Fragmentation and Losses on Markovian Arrival Processes. In: Proceedings of ASMTA 2008. (June 2008), 31–46.

[3] Heckmüller, S., Wolfinger B. E.: Using Load Transformations for the Specification of Arrival Processes in Simulation and Analysis, SIMULATION 85(8), 2009, 485–496.

[4] Zhang, Q., Heindl, A., and Smirni, E.: Models of the departure process of a BMAP/MAP/1 queue, SIGMETRICS Perform. Eval. Rev. 33(2), 2005, 18–20.

[5] Weigle, M. C., Adurthi, P., Hernández-Campos, F., et al.: Tmix: A Tool for Generating Realistic Application Workloads in ns-2, *ACM SIGCOMM Computer Communication Review,* 2006, Vol 36, no. 3, 67–76.

[6] Hernández-Campos, F., Nobel, A. B., Smith, F. D., et al.: Understanding Patterns of TCP Connection Usage with Statistical Clustering. In *Proceedings of 13th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2005, 35–44.

[7] Wolfinger, B. E., Zaddach, M., Heidtmann, K. D., et al.: Analytical modeling of primary and secondary load as induced by video applications using UDP/IP. Computer Communications 25(11–12) (2002), 1094–1102.

[8] Stevens, W.: TCP/IP illustrated (vol. 1): the protocols. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1993).

[9] Postel, J.: Internet Protocol. RFC 791 (Standard) (September 1981) Updated by RFC 1349.

[10] Postel, J.: TCP maximum segment size and related topics. RFC 879 (November 1983).

[11] Clark, D. D.: Window and acknowledgement strategy in tcp (1982) RFC 813.

[12] Heckmüller, S., Münz, G., Braun, L., Kunde, A., Wolfinger, B. E., Carle, G. Lasttransformation durch Rekonstruktion von Auftragslängen anhand von Paketdaten, Tagungsbericht zu MMBnet 2009 Workshop, 2009, 93–104.

[13] Nagle, J.: Congestion control in IP/TCP internetworks. RFC 896 (January 1984).

[14] Fielding, R., Gettys, J., Mogul, J., et al.: Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard) (June 1999) Updated by RFC 2817.

[15] Braden, R.: Requirements for Internet Hosts – Communication Layers. RFC 1122 (Standard) (October 1989) Updated by RFC 1349.