Lasttransformation durch Rekonstruktion von Auftragslängen anhand von Paketdaten

Stephan Heckmüller*, Gerhard Münz†, Lothar Braun† Aaron Kunde*,
Bernd E. Wolfinger* und Georg Carle†
*AG Telekommunikation und Rechnernetze, Universität Hamburg
eMail: {heckmueller|2kunde|wolfinger}@informatik.uni-hamburg.de
†Lehrstuhl für Netzarchitekturen und Netzdienste, Technische Universität München
eMail: {muenz|braun|carle}@net.in.tum.de

***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 Verkehrsmessung. 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.

I. 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 beispielsweise 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 interessant, um beispielsweise Leistungsbewertungen und Lastprognosen vornehmen zu können oder Verkehrsströme einem bestimmten Anwendungstyp zuordnen zu können.

Abschnitt II 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]–[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 not-

wendig, 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 III 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 Rekonstruktionsverfahren wurden in ersten Experimenten mit MPEG-Videoströmen und Web-Verkehr untersucht. In Abschnitt IV stellen wir die Experimente vor und diskutieren die Ergebnisse. Abschließend wird in Abschnitt V ein Fazit gezogen und ein Ausblick auf Verbesserungs- und praktische Anwendungsmöglichkeiten gegeben.

II. Lasttransformation und ihre Invertierung

Im folgenden Abschnitt erfolgt die Beschreibung von Verarbeitungsvorgängen in Rechnernetzen als Lasttransformationen, sowie die Rekonstruktion von Lasteigenschaften als deren Invertierung. Hierbei wird zunächst der allgemeine methodische Ansatz beschrieben. 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 [5] dargestellt, definiert.

Definition 1: 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. $\Diamond$

Die Last kann somit durch eine Sequenz von Aufträgen a_i einer Auftragsmenge $\mathcal{A}$, 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:

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

Einzelne Aufträge a_i können hierbei beispielsweise Datenübertragungs- oder Verbindungsaufbauwünsche repräsentieren und sind Ankunftszeitpunkten (bezogen auf die entsprechende Schnittstelle) zugeordnet. Um eine detaillierte Spezifikation der Aufträge zu ermöglichen,

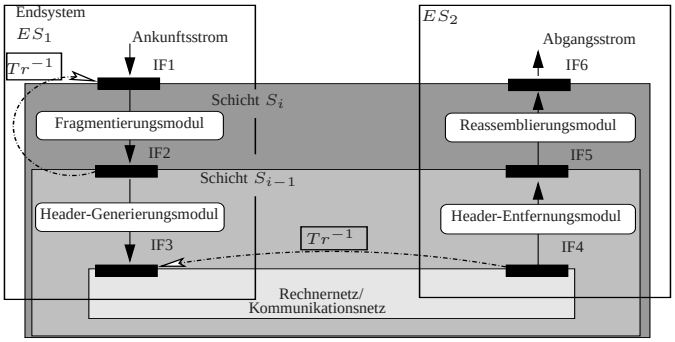


Abb. 1. Paketübertragung als Sequenz von Transformationen

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 jeweils die Aufträge bzw. 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 $A^p = (a_1^p, ..., a_N^p)$ auf eine Sequenz von Sekundärlastaufträgen $A^s = (a_1^s, ..., a_K^s)$ 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 $T^p = (t_1^p, ..., t_N^p)$ auf die Ankunftszeitpunkte der Sekundärlast $T^s = (t_1^s, ..., t_K^s)$ 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.

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.

Für Lasten, die als *Batch Markovian Arrival Processes* beschrieben werden können, konnten eine Reihe

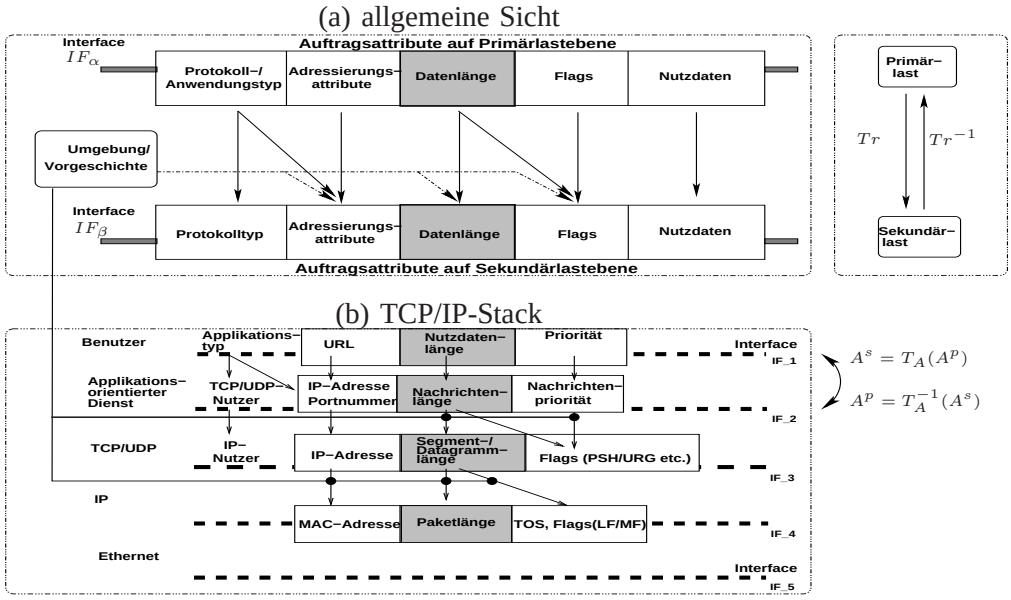


Abb. 2. (Inverse) Transformation von Auftragsattributen im TCP/IP-Protokollstapel

von Transformationsvorgängen als Abbildungen auf solchen Prozessen erfolgreich modelliert werden [2]–[4]. 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 III 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 dargestellt sind. Hier wird deutlich, dass der Zustand der Umgebung häufig eine wichtige Rolle bei der Abbildung von Auftragsattributen spielt. Zum Beispiel kann die Abbildung von IP- auf MAC-Adresse, als Folge von Routing-Entscheidungen oder bei dynamischer Adressvergabe, über der Zeit variieren. Dies erschwert wiederum die Rekonstruktion der Adressattribute der höheren Schichten zu einem späteren Zeitpunkt.

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 [6]. Dies impliziert, dass die Rekonstruktion von Attributen an höheren Schnittstellen ausgehend von mehreren Attributen an den unteren Schnittstellen möglich sein kann.

III. 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 Sinne eines Sendeauftrags), wie er in Abschnitt II 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 Anwendungssschicht. 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 [7]. 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 $n = \lfloor a / \left(8 \lfloor M/8 \rfloor\right) \rfloor$ 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 [8]. Ein Auftrag, dessen

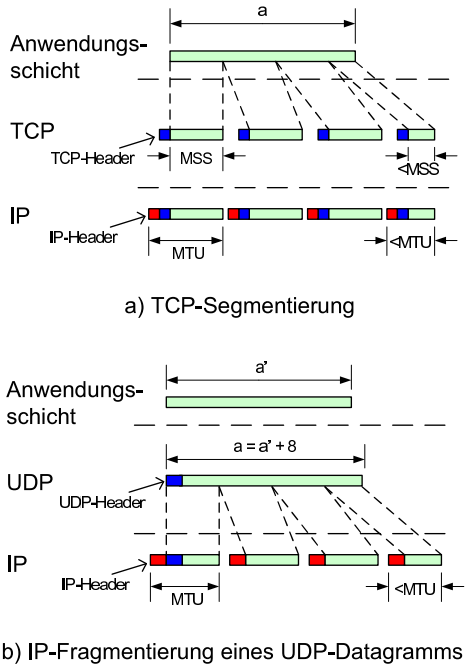


Abb. 3. Unterteilung von Aufträgen

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 [9] 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.

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 III-A an einem Beispiel illustriert.

Danach stellen wir in den Unterabschnitten III-B und III-C zwei Möglichkeiten vor, wie sich Auftragslängen anhand der im Netzwerk 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 Multiple-

xen 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.

Unterabschnitt III-D stellt schließlich eine Möglichkeit vor, mit der die Momente der rekonstruierten Auftragslängen durch eine obere und untere Schranke abgeschätzt werden können. Die Bestimmung der Momente ist dann interessant, wenn die einzelnen Auftragslängen aus Kapazitätsgründen vom Messpunkt nicht gespeichert oder exportiert werden können.

A. 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], [3]), 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 $j = 6000$ und rechts das zweite Moment (d.h. das mittlere Quadrat) der Auftragslängen in Abhängigkeit von j. Unterteilt man die Aufträge in Fragmente der maximalen Länge $M = 1500$, erhält man die in Abbildung 5 gezeigten Kurven für die Verteilungsfunktionen für $j = 6000$ (links) und das zweite Moment der Fragmentlängen (rechts). Die Verteilungsfunktion der Fragmentlängen ist für $j = 6000$ nahezu identisch, während sich die Verteilungsfunktionen der Auftragslängen deutlich unterschieden. Auch das zweite Moment der Auftragslängen weist je nach Verteilung deutlich unterschiedliche Verläufe auf, während das zweite Moment der Fragmentlängen für $j > M$ für alle Verteilungen nahezu identisch ist.

Um verschiedene Auftragslängenverteilungen unterscheiden zu können, ist also eine Rekonstruktion erforderlich. In den folgenden Abschnitten werden dafür zwei Verfahren sowie eine Methode zur Approximation der Momente der Auftragslängen vorgestellt.

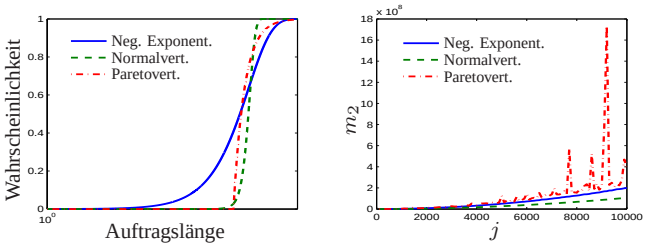


Abb. 4. Verteilungsfunktion der betrachteten Verteilungen für $j = 6000$ (links) sowie zweites Moment m_2 der Auftragslängen für verschiedene Verteilungen in Abhängigkeit von j (rechts)

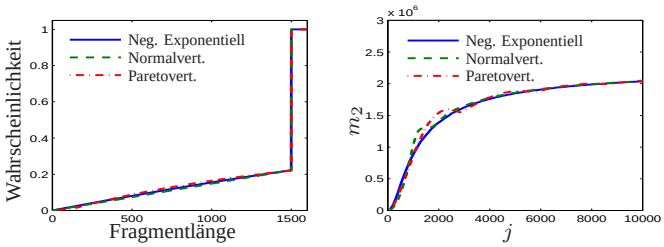


Abb. 5. Empirische Verteilung der Auftragslängen für $j = 6000$ (links) sowie zweites Moment m_2 der Fragmentlängen für verschiedene Auftragslängenverteilungen in Abhängigkeit von j nach Fragmentierung mit $M = 1500$ (rechts)

B. 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. Im Falle von IP und TCP heißt dies insbesondere, dass keine Header-Felder ausgewertet werden, die von der Fragmentierung (z.B. *More Fragments*-Flag im IP-Header) bzw. Segmentierung (z.B. TCP-Flags, Sequenznummern) beeinflusst werden. 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 Abschnitts erläutert.

Als weitere Voraussetzung für das Verfahren muss die senderseitige Begrenzung der Nutzdatenlänge M bekannt sein, bei TCP also die MSS. Ist diese Begrenzung nicht bekannt, kann sie aus der größten beobachteten Nutzdatenlänge geschlossen werden. Eine weitere Möglichkeit ist, Hintergrundwissen über die maximale

Rahmenlänge der verwendeten Sicherungsschicht zu verwenden und daraus M abzuschätzen. Beim gängigen IEEE-802.3-Standard für drahtgebundenes Ethernet ist die MTU beispielsweise 1500 Byte, bei PPPoE (*Point-to-Point Protocol over Ethernet*) [10], das bei DSL zum Einsatz kommt, 1492 Byte. Wie bereits erwähnt, ergibt sich die maximale Nutzdatenlänge aus der Differenz zwischen MTU und der Länge der Kontrollinformation der Vermittlungsschicht und ggf. auch der Transportschicht. Bei IP-Fragmentierung entspricht die Kontrollinformation dem IP-Header, der bei IP-Version 4 ohne Optionsfelder 20 Byte lang ist. Bei TCP-Segmentierung kommt in jedem Paket der TCP-Header hinzu, der ebenfalls ohne Optionsfelder eine Länge von 20 Byte besitzt. Optionsfelder werden allerdings gerade bei TCP häufig verwendet (z.B. die Zeitstempel-Option, die zur Laufzeitermittlung verwendet wird), so dass weitere Bytes für den TCP-Header reserviert werden und die MSS entsprechend kleiner ausfällt. Im Gegensatz zum TCP-Header kommt der 8 Byte lange UDP-Header nicht in jedem IP-Paket sondern nur zu Beginn eines UDP-Datagramms vor. Wird in jedem UDP-Datagramm ein vollständiger Auftrag versendet, verlängert sich somit die Auftragslänge aus Sicht von IP um 8 Byte.

Die Längen der zu einem Auftragsstrom gehörenden Pakete seien $p_1, p_2, \ldots$, die zugehörigen Nutzdatenlängen $l_1, l_2, \cdots$. 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 $\hat{a}_1$, wenn man $i_0 = 0$ setzt.

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. $l_i > 0$).

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. Als Folge wird die Auftragslänge fälschlicherweise zur Länge des darauffolgenden Auftrags hinzugezählt. Die Wahrscheinlichkeit für einen solchen Fehler beträgt $\frac{1}{M}$ unter der Annahme, dass der Rest der Division $a \div M$ gleichverteilt in $[0; M-1]$ 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 folgende Effekte:

- Der Algorithmus von Nagle [11] führt dazu, dass Segmente, die kürzer als MSS sind, nur dann verschickt werden, wenn der Empfang aller vorangegangenen Segmente bestätigt wurde. Wenn der Nagle-Algorithmus aktiv ist, werden somit neue Aufträge im TCP-Sendepuffer akkumuliert, wenn der vorangegangene Auftrag noch nicht vollständig versandt wurde.
- Umgekehrt kann die TCP-Flusskontrolle dazu führen, dass während des Versandes eines großen Auftrags Segmente unterhalb der MSS auftreten. Grund hierfür ist, dass der Sender mit diesem Segment das Empfangsfenster seit der letzten Quittierung ausgeschöpft hat und bis zur nächsten Quittierung keine zusätzlichen Daten senden darf.
- Schließlich kann auch der Sendepuffer so dimensioniert sein, dass er keinem ganzzahligen Vielfachen der MSS entspricht. Bei vollständiger Füllung entsteht also ebenfalls ein Segment unterhalb der MSS.

Abgesehen von der senderseitigen Unterteilung kann

es zudem auf dem Pfad zwischen Sender und Messpunkt zu einer zusätzlichen Fragmentierung langer Pakete kommen, wenn auf einem Pfadabschnitt eine kleinere MTU gilt als vom Sender angenommen. Nur wenn dieser zusätzliche Fragmentierungsschritt bekannt ist, können durch eine zweistufige Anwendung des vorgestellten Verfahrens die ursprünglichen Auftragslängen zurückgewonnen werden. Da die kleinste MTU auf dem Pfad aber meist im Vorhinein bestimmt wird [12], treten Fragmentierungen im Netz sehr selten auf.

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 [13]. Pipelining erlaubt, mehrere Anfragen oder Antworten hintereinander weg zu verschicken, wird allerdings von vielen Browsern nicht standardmäßig verwendet. Beim Chunking wird eine Antwort in mehrere Abschnitte unterteilt, die nacheinander gesendet werden.

C. 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 [14] schreibt für diesen Fall vor, dass der Sender 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. 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:

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

Setzt man Gleichung (4) in (3) ein, lassen sich die Auftragslängen ausschließlich mit Informationen aus den Segmenten bestimmen, bei denen das PUSH-Flag gesetzt ist:

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

Damit diese Formel auch für die Länge des ersten Auftrags $\hat{a}_1$ 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 drei wichtige Vorteile gegenüber dem Aufsummieren von Nutzlasten:

- Die Beobachtung des Paketstroms kann sich auf Segmente mit gesetztem SYN- oder PUSH-Flag beschränken.
- Nur dann, wenn ein Segment mit gesetztem PUSH-Flag verloren geht bzw. unbeobachtet bleibt, wird eine Auftragsgrenze nicht erkannt.
- Übertragungswiederholungen von Segmenten ohne gesetztes PUSH-Flag haben keinen Einfluss auf die rekonstruierten Auftragslängen. Übertragungswiederholungen von Segmenten mit gesetztem PUSH-Flag werden dadurch erkannt, dass sich eine Auftragslänge $\hat{a}_j \leq 0$ ergibt.

Fehler treten dadurch auf, dass das PUSH-Flag häufig 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 Abschnitt III-B 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.

Durch den Nagle-Algorithmus kann es wieder zu einer Akkumulation von Aufträgen im Sendepuffer kommen, die nicht erkannt werden kann. Unterteilungen und Akkumulationen von Aufträgen oberhalb der Transportschicht können ebenfalls nicht erkannt werden.

D. Momente der rekonstruierten Auftragslängen

Flow-basierte Messprozesse sind darauf ausgelegt, eine feste Anzahl von Messwerten und Statistiken pro Flow zu erheben. Die Speicherung einer beliebigen Anzahl an rekonstruierten Auftragslängen ist in diesem Fall nicht möglich. Als Alternative können aber Statistiken über die Folge der rekonstruierten Auftragslängen $\hat{a}_1, \ldots, \hat{a}_N$ erhoben werden, wie zum Beispiel die mittlere Auftragslänge bzw. ganz allgemein die Momente m_k:

$$m_k = \frac{1}{N} \sum_{j=1}^{N} \hat{a}_j^k \tag{6}$$

Da N im Voraus nicht bekannt ist, müssen über die Dauer eines Flows hinweg die Momentensummen $\sum_j \hat{a}_j^k$ mitgeführt werden, die schnell extrem große Werte außerhalb des darstellbaren Zahlenbereichs annehmen können. Diesem Problem kann dadurch begegnet werden, dass die Auftragslänge nicht in Byte angegeben sondern als ganzzahliges Vielfaches einer größeren Dateneinheit approximiert wird.

Im Folgenden wird der Fall betrachtet, dass die Auftragslänge durch das größte ganzzahlige Vielfache d_j der maximalen Nutzdatenlänge M approximiert wird, das kleiner als die rekonstruierte Länge $\hat{a}_j$ ist:

$$\max_{d_j \in \mathbb{N}} \{M d_j\} < \hat{a}_j \implies d_j = \left\lfloor \frac{\hat{a}_j}{M} \right\rfloor \tag{7}$$

Falls die Auftragslänge nach dem Verfahren aus Unterabschnitt III-B bestimmt wird, entspricht d_j der Anzahl der Pakete mit maximaler Nutzdatenlänge zwischen zwei Paketen nicht-maximaler Länge. Für das k-te Moment der Auftragslängen lassen sich mit den Potenzsummen von d_j obere und untere Schranken angeben:

$$\frac{M^k}{N} \sum_{j=1}^{N} d_j^k \leq m_k \leq \frac{M^k}{N} \sum_{j=1}^{N} (d_j + 1)^k \tag{8}$$

Die obere Schranke für das k-te Moment lässt sich durch Anwendung des Binomischen Lehrsatzes mit der ersten bis k-ten Potenzsumme von d_j ausdrücken:

$$\frac{M^k}{N} \sum_{j=1}^{N} (d_j + 1)^k = \frac{M^k}{N} \sum_{i=0}^{k} \left\{ \binom{k}{i} \sum_{j=1}^{N} d_j^i \right\} \tag{9}$$

Für die Berechnung aller unteren und oberen Schranken für die Momente 1 bis k der Auftragslängen müssen also nur die erste bis k-te Potenzsumme von d_j und die Anzahl der Aufträge N über die Dauer eines Flows hinweg berechnet und gespeichert werden.

Für kleine mittlere Auftragslängen, insbesondere für $m_1 \leq M$, stellen die Momente der Paketlängen eine gute Näherung für die Momente der Auftragslängen dar. Da die Momente der Paketlängen stets kleiner als die entsprechenden Momente der Auftragslängen sind, ergibt sich hierdurch eine weitere untere Schranke für m_k.

IV. Bewertung der Rekonstruktionsalgorithmen

Im folgenden Abschnitt soll die Genauigkeit der vorgestellten Rekonstruktionsalgorithmen betrachtet werden. Bevor die vorgeschlagenen Verfahren auf aufgezeichneten HTTP-Verkehr angewendet werden, soll zunächst die längenbasierte Rekonstruktion unter weitestgehend kontrollierbaren Rahmenbedingungen untersucht werden. Hierzu wurde die Übertragung eines MPEG-4-Videos in einem 100Mbit-Ethernet-LAN über TCP nachgestellt. Es wurden Traces mit Zwischenankunftszeiten und Auftragslängen genutzt, welche unter [15] erhältlich sind. Während der Messung wurden Pakete mit der jeweils durch den Trace vorgebenen Länge und Zwischenankunftszeit versendet. Die genutzten Videotraces wiesen eine konstante Zwischenankunftszeit von 40ms auf und die Auftragslänge war variabel. Die MSS betrug 1448 Byte; die Messung erfolgte empfängerseitig mithilfe des Werkzeugs *tcpdump*.

Die Ergebnisse der Messung sind für den Film *Mr. Bean* in Abbildung 6 dargestellt. Im linken Teil der Abbildung sind die empirischen Verteilungen der Paketlängen dargestellt. Es ist zu erkennen, dass die Segmente nicht maximaler Länge mit guter Näherung gleichverteilt sind. Weitere Experimente mit anderen Paketlängenverteilungen führten zu vergleichbaren Resultaten, wobei dies auf Fälle beschränkt werden muss, in denen große Pakete (relativ zur MSS) hinreichend wahrscheinlich sind. Dies steht im Einklang mit den in Abschnitt III-A gezeigten Resultaten bezüglich der Fragmentierung von Aufträgen, deren Längen vorgebenen, theoretischen Verteilungen entsprachen. Im rechten Teil der Abbildung sind die empirischen Verteilungen

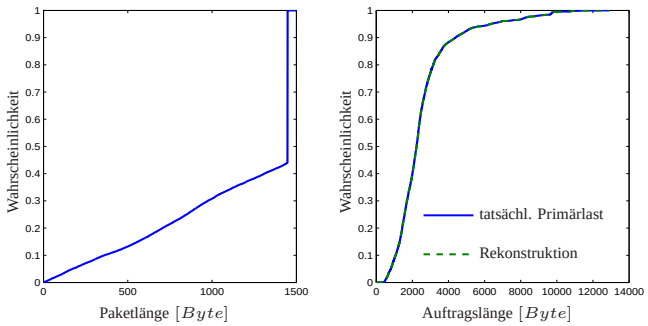


Abb. 6. Empirische Verteilung der Paketlängen bei Übertragung eines *MPEG-4*-Videos über TCP (links) und dazugehörige Rekonstruktion der Auftragslängenverteilung (rechts)

Tabelle I
ERGEBNISSE DER LÄNGENBASIERTEN REKONSTRUKTIONEN 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

Tabelle II
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

der ursprünglichen Auftragslängen sowie die empirische Verteilung der gemäß Paketlängen rekonstruierten Auftragslängen dargestellt. Es ist zu erkennen, dass diese fast ohne Fehler rekonstruiert werden kann – die beiden Kurven sind kaum unterscheidbar.

In einem weiteren Experiment wurden die vorgeschlagenen Algorithmen auf aufgezeichneten Webverkehr angewendet, welcher aus HTTP-Anfragen (Version 1.1) an die Server *de.wikipedia.org* bzw. *www.debian.org* resultierte. Die Aufzeichnung des Verkehrs erfolgte wieder 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.

Der Vergleich der rekonstruierten Auftragslängen mit den tatsächlichen Auftragslängen ist in Abbildung 7 dargestellt. Zur Rekonstruktion wurden die beiden in den Abschnitten III-B und III-C vorgeschlagenen Methoden verwendet. Im oberen Teil der Abbildung wurde für die längenbasierte Rekonstruktion die maximale Nutzdatenlänge, welche innerhalb der jeweiligen TCP-Verbindung auftrat, als Parameter M angenommen. In den unteren Diagrammen wurde mit $M = 1360$ ein fester Wert für die maximale Nutzdatenlänge verwendet.

Im Falle von *de.wikipedia.org* ist zu beobachten, dass die längenbasierte Rekonstruktion und die Rekonstruktion mithilfe des PUSH-Flags nahezu gleiche Ergebnisse liefern. Der Vergleich mit den Längen, welche aus den HTTP-Headern entnommen wurden, zeigt allerdings, dass insbesondere die Wahrscheinlichkeit für hohe Auftragslängen unterschätzt wird. Im Falle von *www.debian.org* ist die Abweichung von den Referenzwerten geringer als bei *de.wikipedia.org*. Dabei zeigen sich Unterschiede zwischen den beiden Methoden: die längenbasierte Rekonstruktion liefert bessere Ergebnisse als die Rekonstruktion anhand der PUSH-Flags.

In den Tabellen I und II 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 korrekt und falsch rekonstruierte 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 Abschnitt III-B 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.

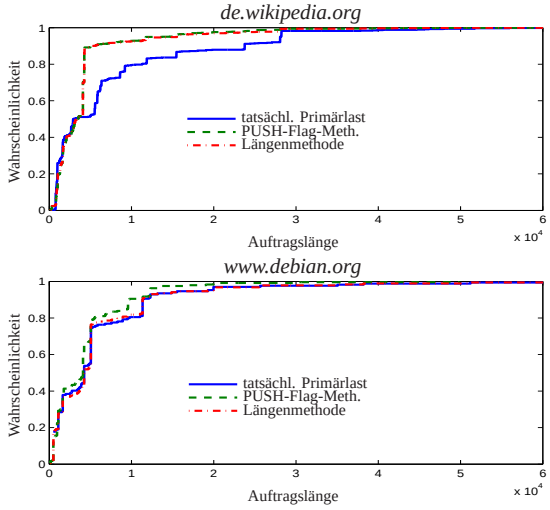
Diesbezüglich wurden in den Abschnitten III-B und III-C mögliche Fehlerquellen der Rekonstruktionsverfahren diskutiert. Über die tatsächlichen Ursachen der

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 II unter "$l_i = MSS$ mit PUSH-Flag" angegeben.

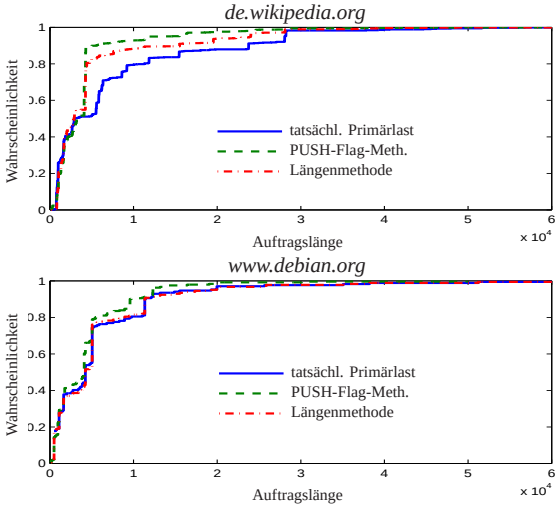
Alle weiteren Fehlerfälle wurden in beiden Tabellen in der Zeile "$l_i < MSS$, $\hat{a}_j \neq 4096$" zusammengefasst. Denkbare Ursachen für diese Fehler sind ein volles Empfangsfenster oder ein ausgeschöpfter Sendepuffer mit von 4096 Byte abweichender Größe. Wir planen diesbezüglich weitere Untersuchungen anzustellen. Die Vermutung, dass das PUSH-Flag im letzten Paket eines Auftrags stets gesetzt ist, konnte bestätigt werden.

Wie bereits erwähnt tritt insbesondere bei *de.wikipedia.org* ein relativ hoher Anteil (ca. 3%) von Paketen mit einer Nutzdatenlänge von 1360 Byte auf, die knapp unterhalb der maximal beobachteten Nutzdatenlänge $M = 1368$ Byte liegt. Diese Pakete haben ihrer Ursache vermutlich in der Beschränkung des Sendepuffers auf 4096 Byte, was sich auch daran erkennen lässt, dass sie meist zwischen Paketen mit maximaler Nutzdatenlänge vorkommen, die zum selben Auftrag gehören. Die längenbasierte Rekonstruktion wurde daher zusätzlich mit $M = 1360$ Byte durchgeführt. Wie im unteren Teil von Abbildung 7 zu sehen ist, führt dies zu einer höheren Genauigkeit der längenbasierten Rekonstruktion, wenngleich die Festsetzung der maximalen Paketlänge M eher heuristisch ist.

In einem zusätzlichen Experiment wurde eine Kombination der beiden Rekonstruktionsverfahren untersucht, bei der solche Pakete als Auftragsgrenze angesehen werden, bei denen zugleich das PUSH-Flag gesetzt und die Nutzdatenlänge kleiner als M ist. Hierdurch werden fehlerhafte Rekonstruktionen durch periodisch gesetzte PUSH-Flags vermieden, sofern diese in Paketen maximaler Nutzdatenlänge auftreten. Die Auftragslängen werden, wie in Abschnitt III-C erklärt, anhand der Sequenznummern berechnet.



(a) Längenbasierte Rekonstruktion mit $M = \max(l_i)$



(b) Längenbasierte Rekonstruktion mit $M = 1360$

Abb. 7. Rekonstruktion der Auftragslängenverteilungen zweier Webserver (PUSH-Flag-Methode und Längenmethode)

Die Ergebnisse dieses kombinierten Ansatzes sind in Tabelle III dargestellt. Es ist ersichtlich, dass im Falle von *www.debian.org* keine Fehler mehr auftreten, während für *de.wikipedia.org* keine Verbesserungen gegenüber der rein längenbasierten Rekonstruktion erreicht werden.

V. Fazit

Im vorliegenden Beitrag wurde untersucht, inwiefern aus paketorientierten Messdaten auf der Vermittlungsschicht Eigenschaften der Ankunftsprozesse auf

Tabelle III
ERGEBNISSE DER REKONSTRUKTIONEN BASIEREND AUF PUSH-FLAG UND NUTZDATENLÄNGE

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

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. 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.

Bezüglich der Rekonstruktion wurden zwei Verfahren vorgestellt: Das erste basiert ausschließlich auf den beobachteten Paketlängen und kann unabhängig vom genutzten Transportprotokoll eingesetzt werden. Das zweite nutzt Headerfelder des TCP-Protokolls und ist dementsprechend auf TCP-Verkehr beschränkt. Es wurden weiterhin die Vor- und Nachteile beider Methoden beleuchtet und eine momentenbasierte Beschränkung der Auftragslängen vorgeschlagen, die insbesondere dann von Interesse ist, wenn nicht alle rekonstruierten Längen verarbeitet werden können (z.B. aufgrund von begrenzten Speicherkapazitäten oder zu geringer Verarbeitungsgeschwindigkeit).

Die vorgeschlagenen Algorithmen wurden in ersten Experimenten anhand von MPEG-Videoströmen und Webverkehr untersucht. Im Falle der MPEG-Videoströme waren beide Verfahren in der Lage, die Auftragslängen mit hoher Genauigkeit zu rekonstruieren. Beim untersuchten Webverkehr zeigte sich, dass insbesondere große Auftragslängen häufig nicht korrekt rekonstruiert und als mehrere kleine Aufträge aufgefasst werden. Diese Abweichung wird hauptsächlich durch eine Reihe von TCP-Mechanismen ausgelöst, die in unterschiedlichem Umfang die Genauigkeit der beiden Algorithmen beeinflussen. Es zeigte sich, dass durch eine Kombination beider Verfahren die Genauigkeit in einigen Fällen verbessert werden kann.

In weiterführenden Arbeiten werden wir die Ursachen der 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. Schlussendlich möchten wir die rekonstruierten Auftragslängen zur Verkehrsklassifizierung in verschiedene Anwendungsklassen einsetzen.

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.: Load Transformations for Markovian Arrival Processes. In: Proceedings of ASMTA 2007. (June 2007) 35–43

[3] Heckmüller, S., Wolfinger, B.: 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

[4] Heckmüller, S., Wolfinger, B.: Analytical Modeling of Token Bucket Based Load Transformations. In: Proceedings of SPECTS 2008. (June 2008) 15–23

[5] 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

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

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

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

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

[10] Mamakos, L., Lidl, K., Evarts, J., Carrel, D., Simone, D., Wheeler, R.: A Method for Transmitting PPP Over Ethernet (PPPoE). RFC 2516 (Informational) (February 1999)

[11] Nagle, J.: Congestion control in IP/TCP internetworks. RFC 896 (January 1984)
[12] Mogul, J., Deering, S.: Path MTU discovery. RFC 1191 (Draft Standard) (November 1990)
[13] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard) (June 1999) Updated by RFC 2817.
[14] Braden, R.: Requirements for Internet Hosts - Communication Layers. RFC 1122 (Standard) (October 1989) Updated by RFC 1349.
[15] Fitzek, F., Reisslein, M.: MPEG-4 and H.263 Video Traces for Network Performance Evaluation. Internet: http://www-tkn.ee.tu-berlin.de/research/trace/trace.html