

Ermittlung der unidirektionaler Paketverzögerung durch netzwerkweite Messungen

Philipp Lowack

Betreuer: Dirk Haage

Seminar Innovative Internet Technologien und Mobilkommunikation WS09/10

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: lowack@in.tum.de

ABSTRACT

In diesem Artikel wird das Paper *One-Way Delay Estimation Using Network-Wide Measurements* von Omer Gurewitz, Israel Cidon und Moshe Sidi aus dem Jahr 2006 [5] vorgestellt, analysiert und bewertet. Dabei wird auf die mit diesem Konzept verbundenen Probleme und Einschränkungen aber auch dessen Vorzüge eingegangen. Ebenso werden andere Techniken kurz vorgestellt und mit dem neuen Ansatz verglichen.

Keywords

one-way delay estimation

1. EINLEITUNG

Das Übertragen eines Netzwerkpaketes von einem Host zu einem anderen ist zwangsläufig auch immer mit einer gewissen Verzögerung verbunden. Der Sender muss zunächst die zu sendenden Daten der Reihe nach in zumeist elektrische Signale umwandeln und in das verwendeten Übertragungsmedium einspeisen. Nun breiten sich die Signale auf der Leitung aus und erreichen irgendwann den nächsten Netzwerkknoten, welcher die Signale auffängt und daraus wieder das gesendete Datenpaket bildet. Sowohl das Umwandeln der Daten in Signale als auch das Ausbreiten der Signale und das Deserialisieren der selben beim Empfänger benötigt eine gewisse Zeit, genannt Serialisierungszeit und Ausbreitungszeit. Erstere hängt von der Übertragungsrate ab, letztere entspricht der Lichtgeschwindigkeit im für die Verbindung verwendeten Material. Zu diesen beiden Verzögerungen kommt jetzt noch zusätzlich die Zeit, die von den Betriebssystemen des Senders und des Empfängers sowie auf dem Pfad zwischen ihnen liegenden Routern oder sonstiger Netzwerkinfrastruktur benötigt wird um ihrerseits die Pakete zu verarbeiten. Dies sind meist sogenannte *Queing-Delays*.

Weiß man nun, wie lange ein Netzwerkpaket von A nach B benötigt, kann man einerseits herausfinden, ob das Netz aus- beziehungsweise überlastet ist (das Queing-Delay ist in diesem Fall sehr hoch da die Warteschlangen voll sind) aber auch macht dieses Wissen bestimmte Dienste überhaupt erst möglich. Zu letzterem zählt vor allem die exakte Synchronisation von Uhren über das Netzwerk. Aber auch andere zeitkritische Anwendungen wie zum Beispiel Videostreaming haben ein Interesse an der Rechtzeitigkeit ihrer Übertragungen und müssen deswegen die Dauer der Übertragung berücksichtigen [1].

Die heute in großen Netzen verwendeten Routingmechanismen arbeiten zudem auf einer dynamischen Basis um je nach Situation für ein Paket den jeweils besten Pfad zu finden. So muss der Weg welchen ein Paket zu seinem Ziel nimmt, nicht mit dem Weg, welchen ein Paket vom Empfänger zum ursprünglichen Sender nimmt übereinstimmen. Genauer gesagt können sogar zwei aufeinanderfolgende Pakete des selben Senders zum selben Empfänger unterschiedliche Wege nehmen. Deswegen ist es schwierig durch bloßes Senden eines Paketes die Laufzeit zu bestimmen. Und auch die Paketumlaufzeit vom Sender zum Empfänger und zurück zum ursprünglichen Sender ist wenig aussagekräftig, da das Halbieren derselben keine genaue Aussage über die unidirektionale Laufzeit gibt.

Um die genaue Zeitdifferenz zwischen zwei verschiedenen Uhren ermitteln zu können ist es zwingend nötig, dass beide Uhren synchronisiert sind beziehungsweise man deren Laufzeitunterschied kennt. Ist dies gegeben lassen sich Laufzeiten sehr einfach messen, indem man den momentanen Zeitpunkt in ein Datenpaket einbettet, dieses überträgt und den Empfänger die Differenz des aktuellen Zeitpunktes und dem Zeitstempel des Datenpakets ermitteln lässt. Nun ist aber wie oben schon erwähnt das Synchronisieren zweier Uhren über ein Netzwerk mit unbekannter Latenz kein triviales Problem.

Die in [5] geschilderte Methode, welche im folgenden Kapitel erklärt werden soll, bietet die Möglichkeit, alle Latenzen in einem Netzwerk zu ermitteln. Statt einzelner Messungen mit synchronisierten Uhren wird mit Hilfe der Gesamtlatenzen unabhängiger zyklischer Pfade ein Gleichungssystem aufgestellt. Daraus können schließlich die Latenzen der einzelnen Netzwerkverbindungen ermittelt werden. Zudem wird eine Bewertung abgegeben sowie Vorzüge und Nachteile aufgezeigt.

In Kapitel 3 werden schließlich anderen Methoden welche Laufzeitmessung oder -schätzung ermöglichen knapp dargestellt und hinsichtlich Leistungsfähigkeit und Durchführbarkeit verglichen.

2. DAS KONZEPT VON O. GUREWITZ

2.1 Idee

Da man die Laufzeit eines Netzwerkpaketes in einem nicht homogen belasteten Netzwerk, welches zudem durch heutige Routingtechniken asymmetrische Hin- und Rückwege auf-

weist [11], nur sehr umständlich messen kann, geht das in [5] vorgestellte Verfahren einen anderen Weg. Aufgrund eines dort vorgestellten mathematischen Modells ist es möglich die unidirektionale Latenz einer Netzwerkverbindung durch die Messung der Umlaufzeit mehrerer unabhängiger zyklischer Pfade zu schätzen.

In der Praxis hat man jedoch normalerweise nicht die Möglichkeit jeden Netzwerkknoten zur Latenzmessung einzusetzen, sei es mangels Zugriffsberechtigung, mangels Interesse dieses Knotens an Laufzeitmessungen oder aufgrund von nicht ausreichender Leistungsfähigkeit des Geräts. Deswegen wird das tatsächliche Netz mit einer abstrahierenden Schicht überlagert. In dieser stellt $N = \{A_1, \dots, A_n\}$ die Menge der tatsächlich zur Verfügung stehenden realen Netzwerkknoten dar. Die beiden gerichteten Kanten e_{ij} vom Knoten A_i zum Knoten A_j sowie e_{ji} in die andere Richtung stellen eine bidirektionale Verbindung zwischen den beiden Knoten dar. Diese kann allerdings, da wir nur ein Overlay-Netz betrachten auch ganzen realen Netzwerken, inklusive den nicht zur Latenzmessung zur Verfügung stehenden Knoten in diesem Netz, entsprechen. Sei E die Menge dieser Kanten. Dass die Kanten gerichtet sein müssen, erklärt sich dadurch, dass eine bidirektionale Verbindung in verschiedene Richtungen auch verschiedene Latenzen aufweisen kann. Somit wird jede Richtung durch eine eigene unidirektionale Kante dargestellt. So benötigt man zwischen zwei Knoten A_i und A_j auch zwei Kanten e_{ij} und e_{ji} in jeweils entgegengesetzte Richtungen. In Abbildung 1 ist ein Beispiel für ein solches Overlay-Netz dargestellt.

Da die Uhren der einzelnen Knoten nicht miteinander synchronisiert sind bezeichnet τ_i die Zeitdifferenz zu einer universellen Zeit und τ_{ij} die Zeitdifferenz zwischen den Uhren von A_i und A_j .

$$\tau_{ij} = \tau_j - \tau_i = -\tau_{ji} \quad (1)$$

O. Gurewitz geht in [5] davon aus, dass in einem Netz viele unabhängige zyklische Pfade existieren. Die Gesamtlatenz jeder dieser Pfade besteht aus den aufsummierten Latenzen der einzelnen Kanten. Da nun aber jede Kante in mehreren Pfaden enthalten ist hat man somit für die einzelnen Laufzeiten mehrere Randbedingungen. Findet man ausreichend viele solcher Pfade, und somit ausreichend viele Bedingungen, kann man durch das Lösen eines linearen Gleichungssystems auf die einzelnen Latenzen schließen. Leider ist es aufgrund einiger Eigenschaften des Overlay-Netzes und der damit verbundenen Unterbestimmtheit des Gleichungssystems (siehe Abschnitt 2.2.1) nicht möglich eine eindeutige Lösung zu bestimmen. Durch entsprechende Optimierungsverfahren lässt sich aber eine ausreichend gute Lösung finden.

2.2 Mathematisches Modell

Betrachten wir zuerst ein paar Eigenschaften der Laufzeit $x_{ij}^{[k]}$ eines Paketes k . Die Latenz setzt sich aus zwei unabhängigen Teilen zusammen: einem konstanten, deterministischen und einem variablen, stochastischen Anteil [3]. Der konstante Anteil, wir bezeichnen ihn mit $c_{ij}^{[k]}$, lässt sich wiederum in die Serialisierungszeit und die Laufzeit des Signals auf dem Medium zerlegen. Da diese beiden Werte charakte-

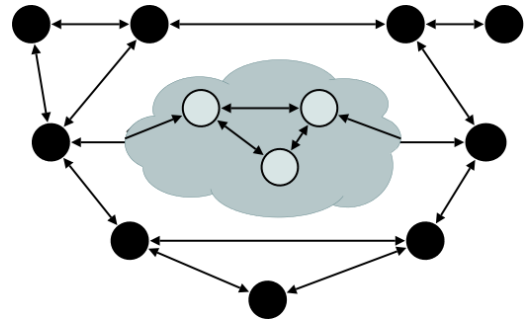


Figure 1: Beispiel für ein Overlay-Netz. Die hellen Knoten in der Wolke haben kein Interesse an einer Latenzmessung und wurden zu einer einzigen Kante zusammengefasst

ristisch für eine Verbindung sind, kann man sie über lange Zeiträume hinweg als konstant ansehen. Der variabler Anteil, bezeichnet mit $v_{ij}^{[k]}$, besteht aus dem sogenannten Queing-Delay, also der Zeit, die das Paket k in Warteschlangen verbringt. Diese Zeit kann sich von Paket zu Paket unterscheiden, da die Netzlast und somit die Anzahl der Pakete in den Warteschlangen sich ständig ändert. Es gilt also für die Gesamtlaufzeit von A_i nach A_j :

$$x_{ij}^{[k]} = c_{ij}^{[k]} + v_{ij}^{[k]} \quad (2)$$

Die tatsächlichen, nicht auf eine bestimmte Messung bezogenen Werte bezeichnen wir entsprechend mit x_{ij} , c_{ij} und v_{ij} wobei letzteres in diesem Fall kein fester Wert, sondern eine Zufallsvariable mit der Dichte $F_{x_{ij}}$ und der Verteilungsfunktion $f_{x_{ij}}$ ist (siehe Abschnitt 2.2.2).

2.2.1 Konstanter Anteil

Der konstante Anteil der Laufzeit eines Paketes lässt sich durch Optimieren eines linearen Gleichungssystems annähern.

Messung. Um die Laufzeit eines Paketes messen zu können, benötigt man zwangsläufig synchronisierte Uhren bei Sender und Empfänger. Misst man aber die Laufzeit eines Paketes entlang eines zyklischen Pfades $A_{j_0} \rightarrow A_{j_1} \rightarrow \dots \rightarrow A_{j_n} \rightarrow A_{j_0}$ so stellt dieses Kriterium kein Problem dar, da die Uhr des Senders die selbe ist wie die Uhr des Empfängers und und so kein Zeitversatz auftritt.

$$\begin{aligned} & \tau_{j_0 j_1} + \tau_{j_1 j_2} + \dots + \tau_{j_{n-1} j_n} + \tau_{j_n j_0} = \\ & = (\tau_{j_0} - \tau_{j_1}) + (\tau_{j_1} - \tau_{j_2}) + \dots + (\tau_{j_{n-1}} - \tau_{j_n}) + (\tau_{j_n} - \tau_{j_0}) = 0 \end{aligned} \quad (3)$$

In realen Netzen ist das Senden entlang eines bestimmten Pfades jedoch nur schwer umsetzen. Mit den IP-Optionen 3 und 9 [12] wäre zwar begrenztes Source-Routing möglich, jedoch werden Pakete mit diesen Optionen aus Sicherheitsgründen an vielen Firewalls verworfen [2]. Deshalb wird ein

anderes Vorgehen benötigt. Bei dem in [5] aufgezeigten Vorgehen wird nicht ein Paket entlang eines Pfades geschickt sondern jeder Teilabschnitt einzeln gemessen. Aus Gleichung 3 folgt auch direkt die Zulässigkeit dieses Vorgehens, da sich die einzelnen Laufzeitunterschiede der Uhren trotz separierten Messungen herauskürzen.

Um die Messung entlang der Kante e_{ab} also durchzuführen senden wir ein Paket k von A_a zu A_b . Dieses enthält den Zeitpunkt $T_{ab}^{[k]}$ zu dem es losgeschickt wurde bezüglich der Uhr von A_a . Kommt das Paket bei A_b an, merken wir uns den Zeitpunkt $R_{ab}^{[k]}$, welcher im Bezug auf die Uhr von A_b gemessen wird. (Das NTP-Protokoll enthält die geforderten Informationen und kann deswegen zu diesem Zweck verwendet werden) Daraus lässt sich nun leicht die Zeitdifferenz $\Delta T_{ab}^{[k]} = R_{ab}^{[k]} - T_{ab}^{[k]}$ berechnen. Es sei darauf hingewiesen, dass die beiden Zeitpunkte im Bezug auf verschiedene, nicht notwendigerweise synchronisierte Uhren gemessen wurden. Um den Empfangszeitpunkt in Abhängigkeit der Laufzeit zu erhalten, muss man also auch die Zeitdifferenz der beiden Uhren berücksichtigen.

$$R_{ab}^{[k]} = T_{ab}^{[k]} + x_{ab}^{[k]} - \tau_{ab} \quad (4)$$

Daraus folgt:

$$\Delta T_{ab}^{[k]} = x_{ab}^{[k]} - \tau_{ab} \quad (5)$$

Da τ_{ab} sehr groß sein kann sind auch negative $\Delta T_{ab}^{[k]}$ möglich. Besonders sei darauf hingewiesen, dass $\Delta T_{ab}^{[k]}$ noch den unbekanntem Laufzeitunterschied der beiden Uhren enthält und deshalb nicht dem gesuchten OWD der Kante entspricht.

Betrachtet man nun die Summe $\Delta T_{ab}^{[k]} + \Delta T_{ba}^{[k]}$, welche die Umlaufzeit eines Paketes von A_a nach A_b und wieder zurück darstellt, kann man eine wichtige Beobachtung machen:

$$\Delta T_{ab}^{[k]} + \Delta T_{ba}^{[k]} = x_{ab}^{[k_1]} - \tau_{ab} + x_{ba}^{[k_2]} - \tau_{ba} = x_{ab}^{[k_1]} + x_{ba}^{[k_2]} \quad (6)$$

Wie schon in Gleichung 3 kürzt sich auch hier die Zeitverschiebung zwischen den beiden Uhren heraus. Dieser Umstand gilt auch für einen beliebig langen Pfad, solange der Endpunkt dem Startpunkt entspricht.

Werden nun viele Messungen mit jeweils immer gleich aufgebauten Paketen gleicher Größe durchgeführt, kann man davon ausgehen, dass zumindest eines der Pakete ohne beziehungsweise nur mit minimaler Verzögerung durch Warteschlangen übertragen wird und so die Gesamtlaufzeit fast nur aus dem konstanten Anteil besteht, der variable Anteil lässt sich nicht weiter minimieren lässt und so zu dem konstanten Anteil gezählt werden kann. Die kleinste auf dem Link e_{ab} gemessene Laufzeit wird also als der konstante Anteil angenommen und als $\Delta T_{ab}^{[min]} := \min_k(\Delta T_{ab}^{[k]})$ bezeichnet. Eine solche Messung wird auf allen Kanten eines Pfades durchgeführt. Anschließend werden die Einzelmessungen zur Gesamtlatenz aufaddiert.

Aufstellen des Gleichungssystems. Für einen Pfad $p = (e_{ij_1}, e_{j_1j_2}, \dots, e_{j_ni})$ gilt also:

$$c_{ij_1} + c_{j_1j_2} + \dots + c_{j_ni} = \Delta T_p^{[min]} := \Delta T_{ij_1}^{[min]} + \Delta T_{j_1j_2}^{[min]} + \dots + \Delta T_{j_ni}^{[min]} \quad (7)$$

Ein lineares Gleichungssystem (LGS) hat im allgemeinen die Form $\mathbf{A} \cdot \vec{c} = \vec{\alpha}$. In unserem Fall ist $\vec{c} = \{c_{ij}\}$ der Vektor mit den gesuchten konstanten Laufzeiten, $\vec{\alpha} = \{\Delta T_p^{[min]}\}$ die auf den zyklischen Pfaden wie mit Formel 7 ermittelten Umlaufzeiten. $\mathbf{A} = \{a_{l,\{ij\}}\}$ ist die Koeffizientenmatrix und beschreibt, ob die Kante e_{ij} auf dem l -ten Pfad liegt indem $a_{l,\{ij\}} = 1$ und $= 0$ sonst. Dies ergibt eine Matrix mit $|E|$ Spalten und $|L|$ Zeilen wobei jede Zeile einen zyklischen Pfad darstellt.

Für eine eindeutige Lösung dieses LGS mit $|E|$ Unbekannten benötigt man nun mindestens ebenso viele unabhängige Gleichungen, also in unserem Fall unabhängige zyklische Graphen.

Finden der zyklischen Pfade. Wir haben nun ein mathematisches Modell, welches uns eine Möglichkeit bietet, durch Lösen eines linearen Gleichungssystems die konstanten Einwegverzögerungen auszurechnen. Was noch fehlt, ist eine Methode die dafür benötigten unabhängigen zyklischen Pfade in ausreichender Menge zu finden. Dazu betrachten wir den Graphen unseres Netzes. Daraus bilden wir einen (per Definition zyklischenfreien) Spannbaum S und gewinnen so eindeutige Pfade von einem zu jedem anderen Knoten. Mit einer nicht im Spannbaum enthaltenen Kante $e_{ij} \in E \setminus S$ kann man nun zwei Knoten verbinden und so einen zyklischen Pfad erzeugen. Diese Vorgehensweise garantiert, dass die einzelnen Pfade jeweils unabhängig sind, da jeder Pfad eine Kante enthält die in keinen anderen Pfad enthalten ist. Doch wie viele solcher Pfade gibt es?

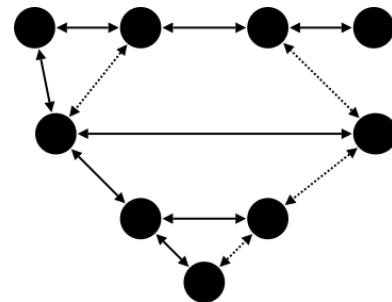


Figure 2: Beispiel eines Spannbauumes des Overlay-Netzes von Abbildung 1. Die gepunkteten Kanten sind nicht Teil des Spannbauumes und werden zur Bildung der zyklischen Pfade verwendet

Ein ungerichteter Baum mit n Knoten besteht aus $n-1$ (ungerichteten) Kanten. Unser Spannbaum besteht aber aus gerichteten Kanten wobei jeweils zwei gerichtete Kanten einer ungerichteten entsprechen (alle Verbindungen sind bidirektional). Er enthält deshalb $2 \cdot (|N| - 1)$ Kanten. (In der Tat ist unser Netz eigentlich ungerichtet, da jede Verbindung

bidirektional ist. Die gerichteten Kanten haben wir nur eingeführt um die unterschiedlichen Latenzen besser abbilden zu können. Somit lässt sich der Satz dennoch anwenden.) Es bleiben also insgesamt $|E| - 2 \cdot (|N| - 1)$ Kanten übrig um ebenso viele zyklische Pfade zu bilden. Jedoch kann davon nur die Hälfte für unsere Zwecke verwendet werden da jeder Pfad sonst in umgekehrter Reihenfolge erneut vorhanden wäre. Außerdem bildet jedes Kantenpaar $A_i \rightarrow A_j \rightarrow A_i$ ebenso einen zyklischen Pfad (e_{ij}, e_{ji}) und man erhält nochmals $\frac{|E|}{2}$ Gleichungen. Sei also P die Menge aller nach diesem Verfahren gefundenen unabhängigen zyklischen Pfade, dann gilt:

$$|P| = \frac{|E| - 2 \cdot (|N| - 1)}{2} + \frac{|E|}{2} = |E| - (|N| - 1) \quad (8)$$

Somit können wir nicht ausreichend viele Pfade finden um eine eindeutige Lösung für unser Problem zu erhalten. Aber wir können versuchen eine möglichst gut passende zu finden. Dies gelingt uns durch das Anwenden eines Optimierungsverfahrens.

Optimieren des Gleichungssystems. Gurewitz untersucht in seiner Arbeit zwei Optimierungsverfahren: Minimieren des *Least Square Errors (LSE)* und maximieren der informationstheoretischen Entropie (*ME*).

Beim LSE-Verfahren versucht man den quadrierten Fehler zu minimieren um die beste Lösung für ein Gleichungssystem zu finden. Das ME-Prinzip hingegen versucht mit Hilfe eines Wahrscheinlichkeitsraumes die Entropie, also den mittleren Informationsgehalt, zu maximieren. Die Grundlage dieses Vorgehens ist einen passenden Wahrscheinlichkeitsraum zu finden. In [5] wird dazu die Wahrscheinlichkeit, ein bestimmtes, zufälligen Wegen folgendes Paket auf einer bestimmten Verbindung zu finden, herangezogen und mit dem konstanten Anteil der Laufzeit verknüpft. Ist dieser größer, dann ist die Vorstellung, dass das Paket auch längere Zeit auf dieser Verbindung verbringt und so die Wahrscheinlichkeit, es dort anzutreffen, ebenfalls größer ist. Da man nun einen Zusammenhang zwischen den konstanten Paketlaufzeiten und dem Wahrscheinlichkeitsraum geschaffen hat, kann man nun mit Hilfe des ME-Prinzips die Wahrscheinlichkeitsverteilung und somit die Laufzeiten schätzen.

Ich möchte hier auf eine genauere Erklärung beider Verfahren verzichten und verweise nur auf passende Literatur beziehungsweise das ursprüngliche Dokument [4][5][6].

Durch Testen beider Verfahren in einer simulierten Umgebung gelangt O. Gurewitz in seiner Arbeit zu dem Schluss, dass sowohl das Maximum-Entropie-Prinzip als auch das LSE-Prinzip mindestens so gute Ergebnisse wie das vom *Network Time Protocol (NTP)* verwendete Halbieren der Paketumlaufzeit. Die geschätzten Werte, sowohl unter Verwendung des LSE-Prinzips als auch mit Hilfe der ME-Methode nehmen mit zunehmend asynchronen Laufzeiten an Genauigkeit zu. Ebenso haben die Simulationen gezeigt, dass das ME-Prinzip, gerade bei großen Netzen im Bezug auf die Genauigkeit der geschätzten Werte bessere Ergebnisse als das

LSE-Prinzip liefert.

2.2.2 Variabler Anteil

Der variable Anteil der Laufzeit eines Paketes stellt die Verzögerung durch Warteschlangen in den Implementierungen der Netzwerkstacks der Betriebssysteme dar. Da diese Dauer von Paket zu Paket unterschiedlich sein kann, wird sie durch eine Zufallsvariable repräsentiert. In [5] wird angenommen, dass es sich dabei um eine Gamma-Verteilung mit folgender Verteilungsfunktion $f_{x_{ij}}(t)$ handelt:

$$f_{x_{ij}}(t) = \frac{(t - C_{ij})^{\alpha-1} e^{-\frac{(t-C_{ij})}{\beta}}}{\beta^{\alpha} \Gamma(\alpha)} \quad (9)$$

In der Tat ist dies eigentlich die Zufallsverteilung der Messwerte $T_{ij}^{[k]}$ wobei diese um den in allen Messungen vorhandenen konstanten Anteil $C_{ij} := c_{ij}^{[k]} - \tau_{ij}$ nach unten verschoben ist. Dadurch repräsentiert die Verteilung nun nur noch den variablen Anteil. Die Parameter α , β und C_{ij} kann man zum Beispiel mit Hilfe der Maximum-Likelihood-Methode finden [13]. Die dafür benötigten Stichproben erhält man aus weiteren Messungen.

2.3 Umsetzung in der Praxis

[5] stellt nicht nur eine Technik zur Messung des OWD vor sondern legt auch Wert darauf, dass das erklärte Verfahren praktisch umsetzbar ist. Deshalb ist in dem Papier auch eine solche Umsetzung dargestellt. Dabei ist das Netz der zur Laufzeitmessung verwendeten Knoten aus Routern der Firma Cisco gebildet. Die Berechnung selbst wird von einem normalen Computer auf welchem die Management-Software *Cisco Service Assurance Agent (SAA)* installiert ist, übernommen. Dadurch brauchen die Router selbst keine aufwendigen Algorithmen berechnen, da diese Aufgabe von einem herkömmlichen Rechner mit ausreichender Leistung übernommen wird. Die für die Messung benötigten Pakete lassen sich ebenfalls mit Hilfe des SAA von den Routern aus verschicken.

Die Messung einer Verbindung läuft folgendermaßen ab: Der SAA gibt einem Router den Befehl ein entsprechendes Paket an den Nachbarn zu schicken, welcher am anderen Ende der zu messenden Leitung angeschlossen ist. Dieser empfängt das Paket mit dem darin enthaltenen Zeitstempel und liefert sowohl diesen als auch den Empfangszeitpunkt an den SAA. Eine solche Messung wird jetzt für jede Verbindung durchgeführt. Hat der SAA alle nötigen Daten kann er die Berechnungen der Laufzeiten nach dem oben geschilderten Verfahren durchführen.

2.4 Bewertung

In [5] wird ein neuer Ansatz vorgestellt, welcher das Problem der unidirektionalen Laufzeitmessung elegant und auf eine interessante Art und Weise löst. Er ist aber nur sinnvoll, wenn man die Latenz jeder Verbindung im Netzwerk wissen will und nur anwendbar, wenn viele, im Idealfall alle, Knoten zur Messung zur Verfügung stehen. Will man jedoch nur die Laufzeit zwischen wenigen, im Extremfall nur zwei Knoten wissen und hat auch nur diese wenigen Knoten zur Verfügung, ist dieses Verfahren nicht anwendbar. Bildet

man in diesem Fall das Overlay-Netz, versteckt man hinter einen Kanten zwangsläufig ganze Netze. Eines der Probleme die hierbei auftreten ist die Tatsache, dass die minimale Laufzeit nicht als über lange Zeiträume hinweg als konstant angesehen werden kann. Anders als bei einer physischen Verbindung, ändern sich die Zustände in einem Netz in unregelmäßigen Zeitabständen, weswegen wir überhaupt erst auf die in diesem Dokument vorgestellte Technik zur Messung von Laufzeiten zurückgreifen müssen. Weiterhin ist die Wahrscheinlichkeit, durch eine begrenzte Anzahl von Messungen die tatsächlich kleinste Laufzeit zu ermitteln, indirekt proportional zur Größe der versteckten Netze. Auf Grund dieser Tatsache haben wir in Abschnitt 2.2.1 die Umlaufzeit der zyklischen Pfade durch viele Einzelmessungen festgestellt. Dies ist aber in den versteckten Netzen nicht möglich da wir auf die darin enthaltenen Knoten keinen Zugriff haben. Somit sind unsere grundlegenden Annahmen in diesem Fall falsch und das Verfahren demnach nicht anwendbar.

Ein weiteres Anwendungsgebiet in dem wir das vorgestellte Verfahren nicht anwenden können ist die Zeitsynchronisation von Client-Hosts. Diese sind im Normalfall nur mit einer einzigen Verbindung mit dem restlichen Netz verbunden. Die beiden daraus in unserem Overlay-Netz resultierenden Kanten sind voneinander abhängig. Ist eine der Kante in einem Pfad enthalten, dann ist es auch die jeweils andere. Somit lässt sich für diese Verbindungen keine Lösung finden. Allgemein gesprochen funktioniert das Verfahren nur, wenn es überhaupt unabhängige zyklische Pfade gibt. Zwar ist dies in den meisten Backbone-Netzwerken der Fall, jedoch gilt diese Voraussetzungen für die meisten privaten Netze und die Netze kleiner Firmen nicht.

Weiterhin liefert uns das vorgestellte Verfahren zwar die Laufzeiten der einzelnen Kanten aber keinerlei Informationen darüber über welche Knoten ein Paket letztendlich geroutet wurde. Werden die Routingentscheidungen zudem dynamisch getroffen, so ist der von dem Pakete beschrittene Pfad unbekannt und somit kann die Gesamtlaufzeit eines Paketes auch nicht aus den OWDs der einzelnen Kanten berechnet werden. Dadurch ist das erlangte Wissen nur begrenzt nützlich.

Ob das Verfahren von O. Gurewitz auch in der Realität funktioniert, wird sich noch herausstellen müssen. Zwar wurden viele aufwendige Simulationen nach unterschiedlichen Gesichtspunkten durchgeführt und sogar ein Implementierungsansatz vorgestellt, jedoch wurden keine Tests mit realer Hardware und realen Netzwerken durchgeführt. In den Simulationen wurde eine Genauigkeit von ungefähr 4 Millisekunden erzielt. Inwiefern sich dieses Ergebnis auf eine reale Implementierung übertragen lässt muss sich jedoch noch zeigen.

3. ANDERE TECHNIKEN IM VERGLEICH

3.1 Vergleich mit NTP

Das *Network Time Protocol (NTP)* bietet an sich nicht die Möglichkeit die Paketlaufzeiten zu messen. Aber es kann zur Synchronisierung der Uhren von Sender und Empfänger verwendet werden und so kann die Latenz mittels Übertragen von Zeitstempeln gemessen werden. Die aktuelle Version 3 wurde 1992 in RFC1305 definiert [7]. Das Protokoll beruht auf der Annahme, dass ein Paket von A nach B den sel-

ben Netzwerkpfad folgt wie ein Paket von B nach A. Auf Grund dieser Annahme wird die Umlaufzeit eines Paketes ermittelt und halbiert um die Laufzeitverzögerung der einfachen Strecke zu erhalten [9]. Aus den schon in der Einleitung geschilderten Gründen ist dieses Verfahren in einem komplexen Netz mit heutigen Routing-Mechanismen möglicherweise sehr ungenau. Trotzdem hat NTP sich zu dem weitverbreitesten Protokoll für Uhrensynchronisation entwickelt.

Sind die Paketlaufzeiten in beide Richtungen symmetrisch, arbeitet NTP sehr genau. Da dies aber aus den bereits genannten Gründen selten den tatsächlichen Umständen entspricht, liegt die zum Beispiel im Internet erreichbare durchschnittliche Genauigkeit der Synchronisation je nach Asymmetrie des Netzwerkes bei ungefähr fünf bis 100 ms [8]. Dieser Fehler vererbt sich an eine der Synchronisation folgenden Laufzeitmessung entsprechend weiter.

3.2 Uhrensynchronisation mittels GPS

Das *Global Positioning System (GPS)* ermöglicht eine genaue Positionsbestimmung mit Hilfe von über 24 auf festgelegten Flugbahnen positionierten Satelliten. Diese senden in konstanten Zeitintervallen ihre Identifikationsnummer und einen mit allen Satelliten synchronisierten Zeitstempel. Letzterer wird durch mehrere Atomuhren an Bord des Satelliten ermittelt und ist somit sehr genau. Durch die unterschiedliche Laufzeit der Funksignale von den Satelliten zu einem Empfänger lässt sich die momentane Position recht genau ermitteln. Mit Hilfe des Zeitstempels lassen sich jedoch auch Uhren synchronisieren. Anders als bei einer Positionsbestimmung ist hierfür jedoch nicht das Signal mehrerer Satelliten notwendig. Bettet man den aktuellen Zeitstempel in ein Paket ein und schickt es zu einem Empfänger welcher ebenfalls mit einem GPS-Empfänger ausgestattet ist, kann man so die Laufzeit des Paketes ermitteln. Da der Zeitstempel (unter Berücksichtigung der Laufzeit des Signals) eine Genauigkeit von einigen Mikrosekunden hat kann man so auch sehr genaue Messungen durchführen [10]. Der große Nachteil an diesem Verfahren ist jedoch, dass man für den Empfang des GPS-Signals eine Sichtverbindung haben muss. Dies lässt sich aber in den meisten Serverräumen oder Rechenzentren nur sehr schlecht umsetzen, womit dieser Ansatz deutlich an Brauchbarkeit verliert. Ist man jedoch auf eine genaue Messung angewiesen und kann die genannten Voraussetzungen umsetzen, findet man in diesem Verfahren eine gute Lösung.

3.3 Synchronisation mittels Zeitzeichenempfänger

Zeitzeichenempfänger, wie die weit verbreiteten Funkuhren und -wecker empfangen ein Funksignal welches Informationen über die aktuelle Zeit sowie Datum und gegebenenfalls einigen Zusatzinformationen enthält. Dieses Signal bekommt die Zeit wiederum meist von einer Atomuhr und ist somit sehr genau. Da das Signal aber nur im Sekundentakt ausgesendet wird, ist die hiermit zu erreichende Genauigkeit ebenso in dieser Größenordnung anzusetzen. Im Gegensatz zu GPS funktioniert diese Technik auch in Räumen solange diese nicht allzu sehr abgeschirmt oder zum Beispiel im Keller eines Gebäudes sind. Diese Einschränkungen sind wohl der Grund warum sich dieses Verfahren nicht zum Messen von Verzögerungen im Netzwerk durchsetzen konnte. Eine

Genauigkeit im Sekundenbereich ist für diesen Zweck bei weitem nicht ausreichend.

4. ZUSAMMENFASSUNG

Das in [5] neu vorgestellte Verfahren bietet eine ganz neue Sichtweise auf das Problem der Laufzeitmessung. Statt durch Synchronisation der Uhren oder mit Hilfe von Zeitstempeln aus synchronisierten Quellen die Verzögerung zu messen, wird aus den auf zyklischen Pfaden gemessene Umlaufzeit ein Gleichungssystem gebildet und anschließend eine Lösung dafür gesucht. Das große Problem an diesem Vorgehen sind die nicht in ausreichender Menge vorhandenen zyklischen Pfade, was das Interpolieren einer Lösung nötig macht. Dafür liefert dieser Ansatz je nach Anwendungsgebiet und Voraussetzungen ein sehr gutes Ergebnis. Will ein Netzbetreiber die Laufzeiten der einzelnen Verbindungen in seinem Backbone-Netz feststellen oder überwachen, zum Beispiel um bessere Videostreaming-Dienste anbieten zu können, bietet dieser Ansatz ihm eine genaue und elegante Möglichkeit dies umzusetzen. Er erhält mit relativ wenig Aufwand alle Laufzeitwerte, diese zudem noch zentral auf einem Monitoring-System. Für einen Endanwender, welcher zum Beispiel die Latenz zu einem Spieleserver wissen will ist das vorgestellte Verfahren aber denkbar ungeeignet. Hier kann nicht auf die Netzwerkinfrastruktur zugegriffen werden um die nötigen Messungen durchzuführen und die zur Verfügung stehenden Knoten reichen bei weitem nicht aus um genug unabhängige zyklische Pfade bilden zu können.

Abschließend lässt sich sagen, dass das neue Konzept bei passenden Voraussetzungen ebenso gute, wenn nicht sogar bessere Ergebnisse liefert als das im weit verbreitete NTP genutzte Halbieren der Umlaufzeit [5]. Ist man jedoch auf extrem genaue Werte angewiesen oder muss mehrere Uhren sehr genau synchronisieren ist das umständlich umzusetzende Verfahren mit Hilfe von GPS aufgrund der einfach unschlagbaren Genauigkeit angebrachter. Aber gerade für die bereits angesprochen Netzbetreiber bietet der in diesem Dokument vorgestellte Ansatz eine echte Alternative zu bestehenden Techniken.

5. REFERENCES

- [1] G. Almes, S. Kalidindi, and M. Zekauskas. A One-way Delay Metric for IPPM. RFC 2679 (Proposed Standard), Sept. 1999.
- [2] R. Farrow. Source address spoofing. <http://www.ntp.org/ntpfaq/NTP-s-algo.htm#Q-ACCURATE-CLOCK>, zuletzt besucht am 17.12.2009.
- [3] S. L. for eduPERT. One-way delay, Jan 2007. <http://kb.pert.geant.net/PERTKB/OneWayDelay>, Version r12 vom 25. Jan. 2007, zuletzt besucht am 16.12.2009.
- [4] S. Guiasu and A. Shenitzer. The principle of maximum entropy. *The Mathematical Intelligencer*, 7(1), 1985.
- [5] O. Gurewitz, I. Cidon, and M. Sidi. One-way delay estimation using network-wide measurements. *IEEE/ACM Trans. Netw.*, 14(SI):2710–2724, 2006.
- [6] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. 3 edition, 1995.
- [7] D. Mills. Network Time Protocol (Version 3) Specification, Implementation and Analysis. RFC 1305 (Draft Standard), March 1992.
- [8] NTP.org. Ntp-faq: How accurate will my clock be? [http://technet.microsoft.com/de-de/library/cc723706\(en-us\).aspx](http://technet.microsoft.com/de-de/library/cc723706(en-us).aspx), zuletzt besucht am 27.01.2010.
- [9] NTP.org. Ntp-faq: How is time synchronized? <http://www.ntp.org/ntpfaq/NTP-s-algo.htm#Q-ALGO-BASIC-SYNC>, zuletzt besucht am 17.12.2009.
- [10] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot. Measurement and analysis of single-hop delay on an ip backbone network. In *IEEE Journal on Selected Areas in Communications*, page 2003, 2003.
- [11] A. Pathak, H. Pucha, Y. Zhang, Y. C. Hu, and Z. M. Mao. Characterizing internet delay asymmetry, Aug 2007.
- [12] J. Postel. RFC 791: Internet Protocol, Sept. 1981. Obsoletes RFC0760 [?]. See also STD0005 [?]. Status: STANDARD.
- [13] T. Schickinger and A. Steger. *Diskrete Strukturen (Band 2)*. Springer-Verlag, Mar 2001.