Prof. Dr. M. H. Scholl
Internet Protocols
Universität Konstanz

Werner König
Mat. 01/458695
We.Koenig@gmx.de

SEMINARARBEIT

THE PROTOCOLS THAT RUN THE INTERNET

"INTERNET NETWORKING – MEASURING DISTANCE AND BANDWIDTH BETWEEN HOSTS"

WERNER KÖNIG

PROF. DR. MARC H. SCHOLL
SEMINAR INTERNET PROTOCOLS
FB INFORMATIK UND INFORMATIONSWISSENSCHAFT
UNIVERSITÄT KONSTANZ

Juni 2003

Werner König
Mat. 01/458695
We.Koenig@gmx.de

Inhalt

Werner König
Mat. 01/458695
We.Koenig@gmx.de

1 Abstract

Für viele verschiedene Anwendungen ist die Information über die Bandbreite der verwendeten Verbindung sehr nützlich. Leider ist diese nur schwer zu ermitteln und bisherige Algorithmen haben noch erhebliche Probleme.

In dieser Seminararbeit werden diese Algorithmen zur Berechnung von Bandbreiten bestimmter Verbindungen zwischen Hosts besprochen. Es werden „Troughput“, „Pathchar“ und vor allem „Packet Pair“ beschrieben und ihre Vor- und Nachteile dargestellt.

Bei Packet Pair werden die verschiedenen Messmethoden, wie Sender Based Packet Pair, Receiver Based Packet Pair und Receiver Only Packet Pair, beschrieben und auf Fehlerquellen und entsprechende Filterungsmethoden, wie Packet Pair Filtering, näher eingegangen.

Als Erweiterungen zu Packet Pair werden Packet Window, Receiver Only Packet Pair und Potential Bandwidth Filtering vorgestellt, welche von Kevin Lai und Mary Baker [3] erstmals beschrieben und in einer Simulationsumgebung mit der bisherigen Version von Packet Pair verglichen wurden.

Ihre Tests ergaben, dass ihr Algorithmus mindestens die gleiche und bei bestimmten Situationen sogar eine um mindestens 37% bessere Genauigkeit als der bisherige Packet Pair Algorithmus aufweisen konnte.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

2 Einführung

Das Internet kann man sich als eine Menge von Knoten vorstellen, die durch gewichtete Kanten miteinander verbunden sind. Die Hosts und Clients werden dabei durch die Knoten und deren Verbindungen durch die Kanten repräsentiert. Die Knoten können mehrere Verbindungen zu einem oder mehreren Knoten besitzen.

Es gibt daher viele verschiedene Wege, um von einem Knoten zu einem anderen zu gelangen. Da nun aber die Kanten verschieden gewichtet sind, weisen gewisse Wege weniger Gesamtgewicht auf und sind deshalb zu bevorzugen. Die Gewichtung der Kanten erfolgt aufgrund verschiedener Maße und Algorithmen, die im Folgenden näher beschrieben werden.

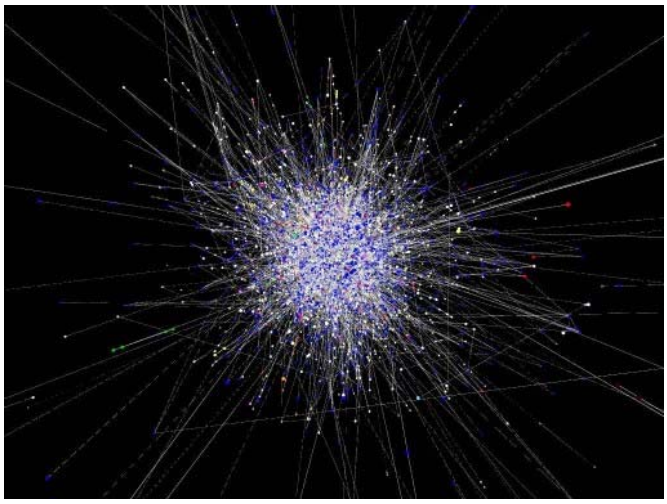

Abb. 2.1: Teil-Visualisierung des Internets mit 10334 Knoten und 8996 Kanten [1].

2.1 Maßeinheiten

Es gibt zwei Hauptgründe für Verzögerungen bei der Übermittlung von Daten im Netzwerk: die Übertragungsverzögerung und die Latenzzeit.

Die Übertragungsverzögerung, auch „transmission delay“ genannt, entsteht bei der Serialisierung der Bits auf den Link, beim Kopieren in den Buffer und anderen Verzögerungen, die sich proportional zu der Paketgröße und damit der Bandbreite der Verbindung verhalten.

Es gibt verschiedene Ebenen im Netzwerk, an denen die jeweilige Übertragungsverzögerung gemessen werden kann, wobei die übergeordneten Ebenen direkten Einfluss auf die darunter liegenden haben.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

Auf der untersten Ebene wird mit der “Application Performance” die maximale Bandbreite ermittelt, die ein Anwender mit seiner speziellen Software über das Netz erreichen kann. Diese ist aber stark abhängig von der Anwendung, da bei langsamen Programmen, wie z.B. CGI-Skripten auf dem Server, die maximale Bandbreite der Verbindung nicht ausgenutzt werden kann.

Ein unabhängigeres Maß stellt “TCP troughput” dar. Hier wird gemessen, wie viele Dateneinheiten TCP als Transport Protokoll in einer Zeiteinheit über eine Verbindung übertragen kann.

Ein protokollunspezifisches Maß ist “available bandwidth”, welches die für den momentanen Zeitpunkt höchste Bandbreite ermittelt. Auf dieses Maß hat vor allem die zum Messzeitpunkt vorherrschende Belastung der Verbindung erheblichen Einfluss. Soll die beste Verbindung über einen größeren Zeitraum ermittelt werden, ist dieses Maß zu unbeständig.

Hierfür ist “bottleneck link bandwidth” am geeignetsten, da es, unabhängig von der momentanen Belastung, die ideale Bandbreite der schwächsten Verbindung auf dem gesamten Weg zwischen zwei Hosts misst. Es wird also über einen längeren Zeitraum die Engstelle einer Übertragung ermittelt und man erhält ein Maß, welches weitgehend unabhängig von zeitlichen Belastungen, Anwendungen und Protokollen ist.

Ein nicht zur Paketgröße proportionales Maß ist die Latenzzeit. Dieses Maß kann bisher schon von Tools wie „ping“ oder „traceroute“ ermittelt werden und wird daher in dieser Arbeit nicht näher betrachtet.

2.2 Motivation

Wenn nun mittels der „bottleneck link bandwidth“ ein Engpass in einem Netzwerk lokalisiert wurde, kann diese Verbindung ersetzt, umgangen oder angepasst werden, um eine bessere Gesamtperformance des Netzwerkes zu erhalten.

So kann ein Netztwerkadministrator zum Beispiel die Verbindung oder den Router durch eine bessere Hardware ersetzten. Oder ein Client kann einen anderen Proxy verwenden, um diese Engstelle zu umgehen.

Hat ein Client mehrere Verbindungsmöglichkeiten, wie beim „Mobile Computing“ zum Beispiel Ethernet, WLAN und Modem, sollte er natürlich die bessere aktive benützen.

Auch können Informationen über die Bandbreiten dafür verwendet werden, „multicast routing trees“ wesentlich dynamischer und effizienter zu machen.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

3 Algorithmen aus der Praxis

Zur Ermittlung der Bandbreite gibt es mehrere Algorithmen, die in der Praxis unterschiedlich verbreitet und wissenschaftlich untersucht sind.

„Troughput“ misst den Datendurchsatz in Korrelation zur benötigten Zeit und ist hauptsächlich vom TCP-Protokoll bekannt.

„Pathchar“ setzt die „round trip time“ von mehreren Paketen mit ihrer jeweilig verschiedenen Größe in Zusammenhang.

„Packet Pair“ sendet immer zwei Pakete direkt hintereinander, misst dann den vom Engpass verursachten Zeitunterschied der beiden Pakete und setzt diesen in Beziehung mit der Paketgröße.

Packet Pair kann in Sender Based Packet Pair (SBPP), Receiver Based Packet Pair (RBPP) und den von Kevin Lai und Mary Baker [3] erstmals vorgestellten Receiver Only Packet Pair (ROPP) aufgeteilt werden. Beim SBPP werden „Packet Send Timing“ und „Acknowledgement (Ack) Arrival Timing“ beim Sender gemessen. Beim RBPP werden „Packet Send Timing“ beim Sender und „Packet Arrival Timing“ beim Receiver gemessen. Beim ROPP wird nur das „Packet Arrival Timing“ beim Receiver gemessen.

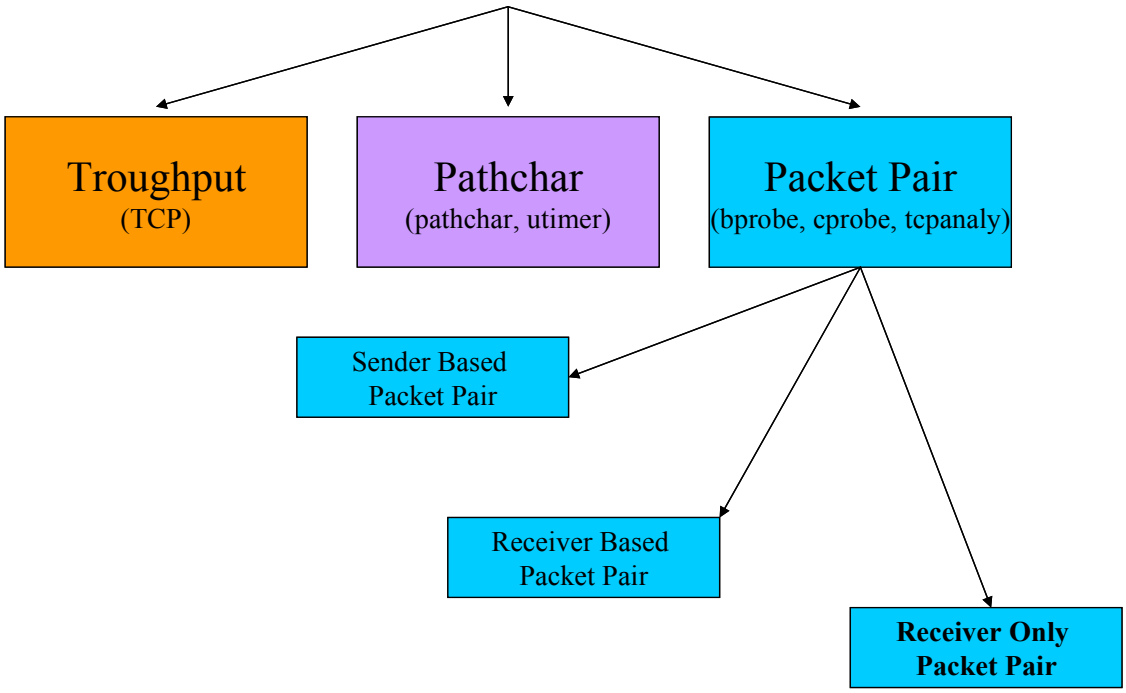

Abb. 3.1: Übersicht Algorithmen zur Bandbreitenberechnung.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

3.1 Troughput

Beim Troughput wird die Menge an Daten gemessen, die ein Transport-Protokoll, wie z.B. TCP, pro Zeiteinheit übertragen kann, gemessen.
TCP erhöht langsam die Menge der zu sendenden Pakete solange, bis die Verbindung überlastet wird und die gesendeten Pakete verworfen werden.

Troughput wird von vielen Faktoren beeinflusst, die aber für die Berechnung der Bandbreite irrelevant sein sollten. Zum Beispiel hat der Troughput einer Anwendung keinen Einfluss auf den Durchsatz einer anderen.

Durch das Herantasten an den maximalen Durchsatz verschwendet Troughput und insbesondere TCP die Netzwerk-Ressourcen und stört damit den normalen Betrieb.

Auch ist die Messung sehr langwierig, da Troughput nur langsam die „sending rate“ erhöht und somit erst spät „packet loss“ auftritt.

3.2 Pathchar

Patchar wurde von Van Jacobson von der Network Research Group, Lawrence Berkeley National Laboratory, entwickelt und setzt die „round trip time“ von Paketen verschiedener Größe in Beziehung.

Es werden aktiv standardmäßig $p = 32$ Pakete pro Hop h und je Paketgröße s gesendet, die abhängig ist von der im Netzwerk festgelegten „Maximal Transfer Unit“ (*MTU*). So werden beim Ethernet 45 verschiedene Größen von Paketen übermittelt und mittels deren Acks vom Kommunikationspartner die „round trip time“ ermittelt.

$$s = \left\lfloor \frac{MTU}{32} \right\rfloor - 1$$

Abb. 3.2.1: s verschiedene Größen.

Durch diesen aktiven Algorithmus entsteht ein nicht zu unterschätzenden Traffic, der linear zur Anzahl der gemessenen Hops steigt. Werden z.B. 10 hops gemessen, erzeugt Pathchar 10 MB zusätzlichen Traffic.

Da der Algorithmus immer auf die Acks von einem Packet warten muss (Latenzzeit l_i) um das nächste zu übermitteln, fällt die Gesamtdauer der Messung sehr hoch aus. So braucht Pathchar bei wiederum 10 Hops 144 Sekunden für einen vollständigen Durchlauf.

$$t = \sum_{i=1}^{h} p \bullet s \bullet l_i$$

Abb. 3.2.2: Laufzeit Pathchar.

Für Pathchar muss der Algorithmus nur auf einem Host implementiert sein, da die Zeitverzögerung mittels Acks beim Sender gemessen wird.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

4 Packet Pair

4.1 Idee

Packet Pair beruht auf der Tatsache, dass wenn zwei Pakete gleicher Größe direkt hintereinander an einer Engstelle aufgestaut werden, sie mit einer Zeitdifferenz von:

$\Delta t = \frac{s^1}{b_{bnl}}$ die Verbindung verlassen.

Die Größe des zweiten Paketes wird hier durch s^1 und die Bandbreite der Engstelle durch b_{bnl} symbolisiert. Die Zeitdifferenz nennt man auch „bottleneck separation“ und durch Umstellen der Formel kann leicht auf die Bandbreite der Verbindung geschlossen werden.

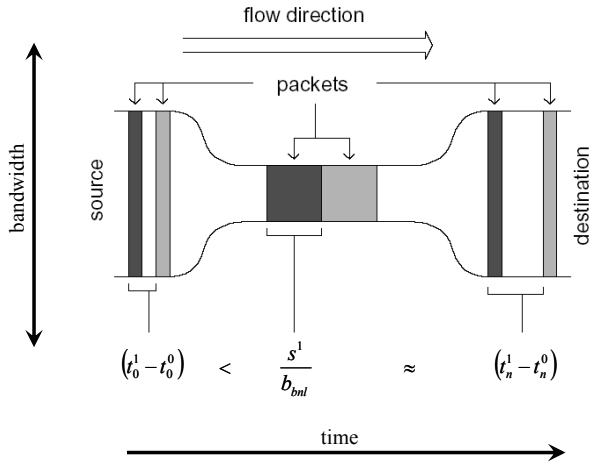

Abb. 4.1.1: Packet Pair Idee [4].

4.2 Messmethoden

Für Packet Pair gibt es drei verschiedene Messmethoden:
Receiver Based Packet Pair (RBPP), Sender Based Packet Pair (SBPP) und Receiver Only Packet Pair (ROPP).

Receiver Based Packet Pair misst die Zeitdifferenzen der Paketpaare bei der Ankunft beim Receiver, welche vom Sender aktiv abgeschickt worden sind. Außerdem übermittelt der Sender dem Receiver die ursprüngliche Zeitdifferenz beim Versand. Aus diesen Werten wird die „bottleneck bandwidth“ ermittelt. RBPP setzt aber voraus, dass auf Sender und Receiver die Packet Pair Software eingesetzt wird.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

Wenn beim Receiver die Differenzen nicht gemessen werden können, kann Sender Based Packet Pair angewandt werden, welcher die „round trip time“ misst.
Der Sender schickt dabei wieder die Paketpaare zum Receiver und wartet nun aber auf die Bestätigungspakete (Acks) des Receivers für die gesendeten Pakete.

Idealerweise wird ein Ack vom Receiver zurückgesendet, sobald ein Paket angekommen ist. So sollte die Zeitdifferenz der Acks analog zu der des Paketpaars sein. Dies kann nur gewährleistet werden, wenn die Acks so klein sind, dass sie nicht auch auf eine Engstelle treffen, aufgestaut werden und somit die Zeitdifferenz sich ändert. Vor allem asymmetrische Verbindungen können hier Fehler verursachen.

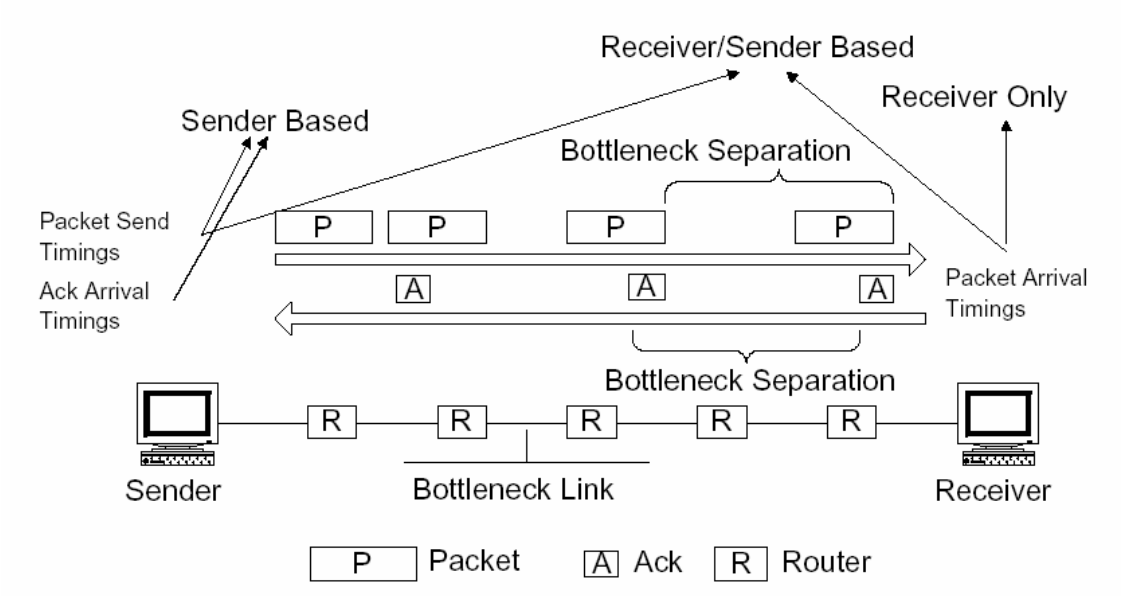

Abb. 4.2.1: Messmethoden Packet Pair [3].

Receiver Only Packet Pair misst nur auf dem Receiver, aber nicht die „round trip time“, sondern passiv die Zeitdifferenzen der ankommenden Pakete des normalen Netzwerkbetriebs und filtert daraus die interessantesten Pakete heraus.

ROPP ist also einfach einzusetzen, da nur auf dem Messhost die Software benötigt wird. Außerdem umgeht es die Fehleranfälligkeit bei der Verwendung von Acks, wie bei SBPP, und verursacht keinen zusätzlichen Traffic, wie SBPP und RBPP.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

4.3 Fehlerquellen

Packet Pair trifft die Annahme, dass die zwei kurz nacheinander gesendeten Pakete genau hintereinander an der Engstelle aufgestaut werden und nach dem Verlassen dieser Verbindung sich die Zeitdifferenz nicht mehr ändert. Dieser idealisierte Vorgang wird in der folgenden Abbildung in der obersten Reihe dargestellt.

Werden die Pakte aber in der Engstelle durch ein anderes Paket getrennt, wird die Zeitdifferenz durch die Größe des zusätzlichen Paketes verfälscht (Abb. 4.3.1: 2. Reihe). Ist diese Größe dem Algorithmus bekannt, kann sie heraus gerechnet werden. Ansonsten werden Fehlerwerte gemessen.

Ein andere Fehler kann entstehen, wenn das Paketpaar nach einer Engstelle durch ein oder mehrere andere Pakete aufgehalten wird, das zweite Paket wieder zum ersten auflaufen kann und somit die Differenz nachträglich verändert wird (Abb. 4.3.1: 3. Reihe).

Wenn die Pakete vom Sender schon mit einem zu großen Abstand abgeschickt werden, kann von der Zeitdifferenz der Pakete nicht klar auf die Bandbreite geschlossen werden (Abb. 4.3.1: 4. Reihe).

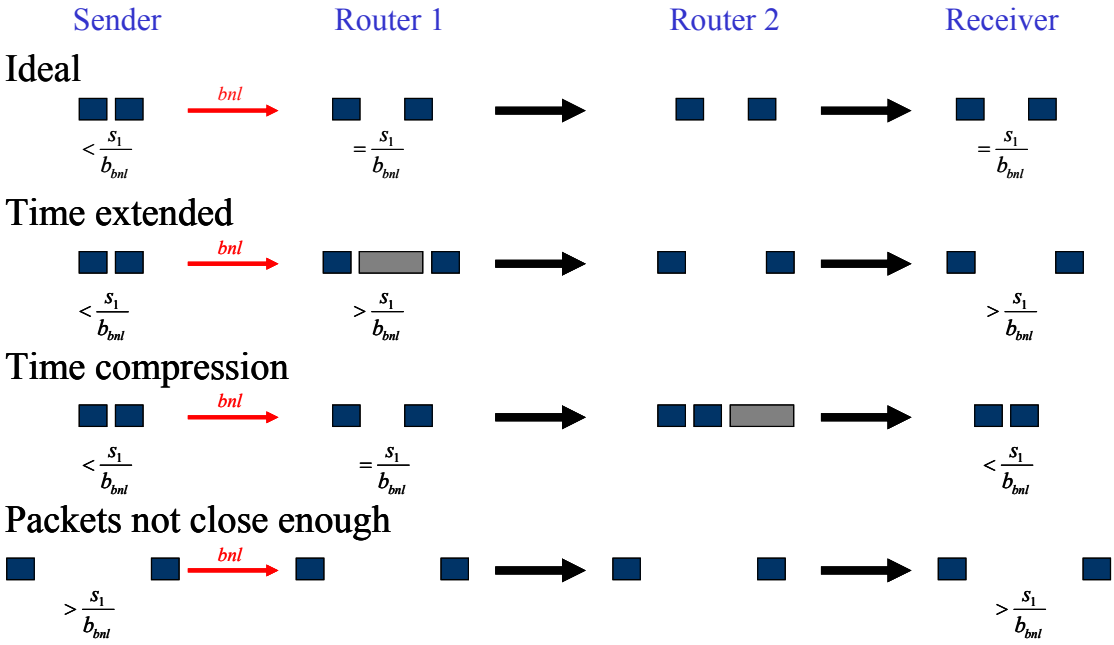

Abb. 4.3.1: Fehlerquellen bei Packet Pair.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

4.4 Voraussetzungen

Um gute Messungen der Bandbreite durch Packet Pair zu ermöglichen, müssen mehrere Bedingungen erfüllt sein.

Beide Pakete jedes Paketpaares müssen die gleiche Größe besitzen, sonst würden sich auch ihre Geschwindigkeiten unterscheiden und die Zeitdifferenzen würden sich dynamisch ändern.

Das Paketpaar muss direkt hintereinander am „bottleneck link“ eingereiht werden und zusammen den gleichen Pfad verfolgen, da sonst die durch den Engpass erzeugte Zeitdifferenz nicht fehlerfrei bestimmt werden kann.

Der Router muss die Pakete nach dem FIFO-Prinzip abarbeiten, also das erste Paket auch wieder zuerst weitersenden, sonst könnte das zweite Paket irrtümlich vor dem ersten beim Receiver bzw. Sender ankommen.

Beim Sender Based Packet Pair dürfen die Acks nicht aufgehalten werden, da sonst die Zeitdifferenz noch nachträglich verfälscht werden würde. Dieses Problem kann bei asymmetrischen Verbindungen auftreten, wenn die Acks durch die schwächere Verbindung zum Sender zurück gesendet werden.

Werden diese Bedingungen nicht gänzlich erfüllt, ergibt sich ein gewisses Rauschen bei den Messergebnissen.
Diese Fehlerwerte müssen aus den zu betrachtenden Daten entfernt werden, um eine gute Schätzung der Bandbreite gewährleisten zu können.

Kevin Lai und Mary Baker [3] schlagen für dieses Problem das von ihnen vorgestellte Packet Pair Filtering vor.

4.5 Packet Pair Filtering

Das Packet Pair Filtering [3] geht davon aus, dass gute Messwerte sich um den realen Wert konzentrieren und damit Cluster bilden. Diese Cluster werden mit dem Kernel Density Estimator ermittelt, alle übrigen Werte als Rauschen deklariert und folglich aus der Messmenge entfernt.

Dieser Algorithmus ist durch die Verwendung vom Kernel Density Estimator, der keine zusätzlichen Annahmen trifft, statistisch korrekt, einfach und schnell auszuführen.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

4.6 Vor- & Nachteile

Gegenüber anderen Algorithmen wie Troughput oder Pathchar kann Packet Pair folgende Vorteile aufweisen.

Packet Pair misst die reine Bandbreite möglichst unabhängig von anderen Faktoren, im Gegensatz zu TCP, welches sehr protokollspezifische Ergebnisse liefert.

Die Störung des normalen Traffics ist relativ gering, da wenige Pakete für eine gute Schätzung ausreichen und dieses Aufkommen unabhängig von der Anzahl der durchlaufenen Hops ist. Messungen ergaben, dass drei Pakete für ein korrektes Resultat ausreichen.
Als passiver Ansatz, wie bei Receiver Only Packet Pair, wird sogar gar kein eigener Traffic erzeugt, sondern nur ankommende Pakete gemessen.

Da zu keiner Zeit die Bandbreite durch aktives Überlasten der Verbindung gemessen wird, verursacht Packet Pair keine explizite Verwerfung von Paketen, wie zum Beispiel Troughput.

Jedoch stehen den genannten Vorteilen in der bisher betrachteten Version von Packet Pair auch gewisse Nachteile gegenüber.

Packet Pair ist noch wenig verbreitet und wird daher nur selten in der Praxis genutzt.

Auch berechnet der bisherige passive Algorithmus erst die Bandbreite, nachdem die TCP-Verbindung geschlossen wurde. Dies kann für manche Anwendungen zu spät sein, um darauf richtig reagieren zu können.

Packet Pair berücksichtigt noch zu wenig das zahlreiche Aufkommen von Paketen verschiedenster Größen. Daraus ergeben sich zum Teil Fehleinschätzungen.

Änderungen der Bandbreite werden je nach Größe der Veränderung entweder gar nicht oder langsam erkannt, da sie anfangs als Rauschen wahrgenommen werden und erst langsam in die Berechnung mit einfließen.

Der Einsatz von Packet Pair als Receiver Based Version ist sehr limitiert möglich, da Sender und Receiver die Software ausführen müssen.

Packet Pair konnte bisher noch nicht mit anderen Algorithmen in einer reproduzierbaren Umgebung verglichen und evaluiert werden.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

5 Erweiterungen zu Packet Pair

5.1 Ziel

Kevin Lai und Mary Baker [3] versuchten den bisherigen Packet Pair Algorithmus mit statistisch korrekten Netzwerk-Modellen so zu optimieren, dass er in der Praxis auch eingesetzt werden kann.

Sie schlagen zur Verbesserung des bisherigen Algorithmus die Verwendung eines „Packet Window“, die Implementierung von „Receiver Only Paket Pair“ und das „Potential Bandwidth Filtering“ als Filterung von Messrauschen vor.

5.2 Packet Window

Ein Packet Fenster bedeutet, dass nur die letzten *w* Pakete in die momentane Messung der Bandbreite einfließen und bei jedem weiteren Packet wieder eine neue Messung generiert wird. Daraus resultiert eine sehr schnelle Schätzung, da nur *w* Pakete benötigt werden, um diese zu berechnen.

Bandbreitenveränderungen werden schnell erkannt und nicht als Rauschen herausgefiltert, da nur die letzten *w* Pakete als Datengrundlage dienen und somit veränderte Werte schneller wahrgenommen werden.

Tests ergaben, dass ein kleines „Packet Window“ 144% genauer als ein unendliches Betrachtungsfenster bei der Einschätzung der Bandbreite einer Verbindung war.

Zu beachten ist, dass je kleiner das Betrachtungsfenster gewählt wird, umso geringer fällt die statistische Aussagekraft der Messung aus, da weniger Werte berechnet werden.

Werner König
Mat. 01/458695
We.Koenig@gmx.de

5.3 Receiver Only Packet Pair

Receiver Only Packet Pair misst passiv die Zeitdifferenzen des durch den normalen Betrieb verursachten Datenaufkommens. Aus den erhaltenen Paketen müssen diejenigen herausgefiltert werden, die nicht die Vorbedingungen des Packet Pair Algorithmus erfüllen.

Receiver Only Packet Pair ermöglicht eine einfachere Verbreitung des Algorithmus, da nur noch auf einem Host die Software vorhanden sein muss.

Da die „download direction“ direkt gemessen wird, tritt die Fehlerquelle durch die Rücksendung der Acks über den zweiten Kanal nicht auf. Die Messung ist aber auch auf diese eine Richtung beschränkt.

5.4 Potential Bandwidth Filtering

Ein Problem von Packet Pair Algorithmen ist die Limitierung der messbaren Bandbreite auf die Bandbreite des Senders.
Werden zum Beispiel zwei Pakete mit 1000 Bytes und 1 ms Unterschied gesendet, kann der Algorithmus maximal eine Bandbreite von 8 Mb/s messen. Dieser theoretische Höchstwert wird als „Potential Bandwidth“ bezeichnet. Umso kleiner die Pakete und umso größer die Zeitdifferenz beim Sender ist, desto geringer fällt die potentiell erreichbare maximale Bandbreite aus.

Idealerweise sollte die potentielle Bandbreite höher als die reale Bandbreite sein, was nur mit entsprechend großen Paketen und einer kleinen Zeitdifferenz erreicht werden kann. Daher geht das Potential Bandwidth Filtering davon aus, dass nur Werte richtig sind, die eine geringer gemessene als die potentiell mögliche Bandbreite haben, da bei den übrigen Werten nicht klar ist, ob der Wert durch die potentielle Bandbreite limitiert wird oder ob der Wert die reale Bandbreite widerspiegelt. Ist dies nicht der Fall, werden die gemessenen Werte aus der Datenmenge herausgefiltert.

Es werden also die gemessene und potentielle Bandbreite miteinander verglichen und die Werte durch Regressionsgeraden stabilisiert. Richtige Werte haben einen kleineren gemessenen als potentiellen Wert und befinden sich daher hinter dem Ellenbogen im Schaubild. Nur diese Werte fließen in die Berechnung mit ein.

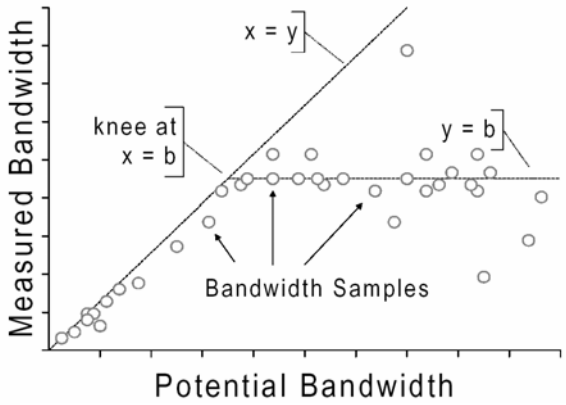

Abb. 5.4.1: Potential Bandwidth Filtering [3].

Werner König
Mat. 01/458695
We.Koenig@gmx.de

5.5 Simulationsergebnisse

Receiver Only Packet Pair und Sender Based Packet Pair unterscheiden sich fast nicht in ihrer Genauigkeit, im Gegensatz zu dem wesentlich schlechteren Sender Based Packet Pair, was hauptsächlich auf die Verwendung von Acks zurückzuführen ist.

Erstaunlich ist, dass ROPP keine Einbußen in der Genauigkeit verzeichnet, obwohl die Software nur passiv auf einem Host misst.

Potential Bandwidth Filtering erzeugt im Gegensatz zur bisherigen Filterungstechnik bei Ethernet 36% und bei asymmetrischer Umgebung sogar 435% weniger Fehler, da bei Asymmetrie, durch die geringere Bandbreite einer Verbindungsrichtung, die Größe der Pakete noch eine viel größere Rolle spielt.

- Accuracy of Receiver-Only Packet Pair

Timings Taken At	Error
Sender	1200.00%
Receiver/Sender	0.09%
Receiver	0.08%

- Accuracy of Potential Bandwidth

Timings Taken At	Filtering	Bandwidth	Error
Sender	Regular	10Mb/s	44.2%
Sender	PBF	10Mb/s	7.8%
Receiver/Sender	Regular	500Kb/s	435.0%
Receiver/Sender	PBF	500Kb/s	0.0%

3/10/99

Abb. 5.5.1: Simualtionsergebnisse von ROPP und Potential Bandwidth Filtering [3].

Werner König
Mat. 01/458695
We.Koenig@gmx.de

6 Zusammenfassung

Bisherige Algorithmen zur Messung von Bandbreite hatten noch vielerlei Probleme und wurden deshalb noch wenig in der Praxis verwendet.

Simulationen ergaben, dass die von Kevin Lai und Mary Baker [3] vorgestellten Erweiterungen zu Packet Pair, wie Packet Window, Receiver Only Packet Pair und Potential Bandwidth Filtering, schnelle und genaue Schätzungen ermöglichen, die sensibel auf Veränderungen reagieren.

Auch können sie einfacher eingesetzt werden und arbeiten mit jeglicher Netzwerkumgebung.

Es muss sich noch herausstellen, ob die Ergebnisse der Simulation auch auf die reale Welt der Vernetzung übertragen werden können.

7 Referenzen

[1]: Stephen Coast, http://www.fractalus.com/steve/stuff/ipmap.

[2]: Van Jacobson. Pathchar - a tool to infer characteristics of Internet paths. Network Research Group, Lawrence Berkeley National Laboratory, 1997.

[3]: Kevin Lai, Mary Baker. Measuring Bandwidth. Department of Computer Science, Stanford University, 1999.

[4]: Kevin Lai, Mary Baker. Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. Department of Computer Science, Stanford University, 2000.

[5]: Kevin Lai, Mary Baker. Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. Department of Computer Science, Stanford University, 2001.

[6]: Vern Paxson. Measurements and Analysis of End-to-End Internet Dynamics. PhD thesis, University of California, Berkeley, 1997.