

# Messung von Netzwerk-Bandbreite

Michael Geisinger  
(geisinge@in.tum.de)

Seminar „Internet-Measurement“,  
Technische Universität München

SS 2005 (Version vom 24. Juli 2005)

## Zusammenfassung

Die Messung der zur Verfügung stehenden Bandbreite zwischen zwei Punkten in einem Netzwerk ist interessant für viele Anwendungen. Doch mit den Protokollen, auf denen heutige Netzwerk-Anwendungen aufbauen, sind entsprechende Messungen nur bedingt möglich. Dieses Papier behandelt die Faktoren, wovon die zur Verfügung stehende Bandbreite abhängt, sowie Möglichkeiten, wie man zuverlässige Annahmen über die zur Verfügung stehende Bandbreite treffen kann, und vergleicht dabei verschiedene Ansätze hinsichtlich ihrer Genauigkeit und Effizienz. Die untersuchten Verfahren, namentlich *bprobe*, *cprobe* und *CapProbe*, lieferten dabei brauchbare Ergebnisse, zeigten aber in manchen Teilgebieten auch Schwächen, deren Ursachen ebenfalls diskutiert werden.

## 1 Einleitung

In jedem Netzwerk hängt die Zeit, die ein Datenstrom von einem Rechner zu einem anderen braucht, von der Bandbreite zwischen diesen beiden Rechnern ab. Damit hat sie einen großen Einfluss auf die Performanz von verteilten Anwendungen. Als Beispiel kann man hier die Übertragung von Dokumenten über FTP und HTTP nennen.

Die verfügbare Bandbreite wiederum wird, neben der Latenz, von zwei Faktoren bestimmt: Einerseits ist sie beschränkt durch die maximale Bandbreite, die auf dem jeweiligen Kanal zur Verfügung steht, die sogenannte Basisbandbreite. Beispielsweise kann man über ein 56kbps-Modem Daten maximal mit 7 kB/s empfangen. Andererseits nützt einem eine schnelle Anbindung nicht viel, wenn die Auslastung des jeweiligen Übertragungsweges sehr hoch ist. Nehmen wir beispielsweise an, dass sehr viele Benutzer gleichzeitig auf eine bestimmte Ressource zugreifen wollen, so wird die Bandbreite des Rechners, auf dem die Ressource liegt, auf alle Benutzer aufgeteilt.

Leider wurden beim Entwurf der Protokolle, auf denen heute die Netzwerkkommunikation basiert, nur relativ einfache Mechanismen zur Messung der zur Verfügung stehenden Bandbreite vorgesehen (vgl. Flusskontrolle bei TCP). Daher muss man auf alternative Methoden zur Schätzung der jeweiligen Werte ausweichen.

Die Anwendungsgebiete sind vielseitig. Das Wissen um die aktuell zur Verfügung stehende Bandbreite ist beispielsweise nützlich für Multimedia-Server, die damit ihre Streaming-Rate anpassen können, und Netzwerkbetreiber können damit ihr Netzwerk überwachen. Dies ist insbesondere interessant, wenn sich die Bedingungen schnell

ändern, wie beispielsweise in drahtlosen Netzwerken. Außerdem könnten überladene Netzwerkpfade auf diese Weise aktiv gemieden werden.

Die weitere Einteilung orientiert sich an folgenden Punkten: Im 2. Kapitel werden kurz einige Grundbegriffe angesprochen. Kapitel 3 erläutert die Motivation zur Messung der Bandbreite und fasst die Anforderungen und Voraussetzungen für die Messung zusammen. Kapitel 4 beschäftigt sich intensiv mit der Messung der Basisbandbreite, erläutert Verfahren zur Konvergenz der Messwerte und stellt Messergebnisse vor. Im 5. Kapitel wird auf die Messung der Pfadauslastung und deren Ergebnisse eingegangen. Kapitel 6 fasst die Ergebnisse nochmals zusammen und weist auf bestehende Probleme hin. Kapitel 7 bietet einen Ausblick auf aktuelle Anwendungen, das 8. Kapitel beinhaltet eine Zusammenfassung der Arbeit.

## 2 Grundlagen

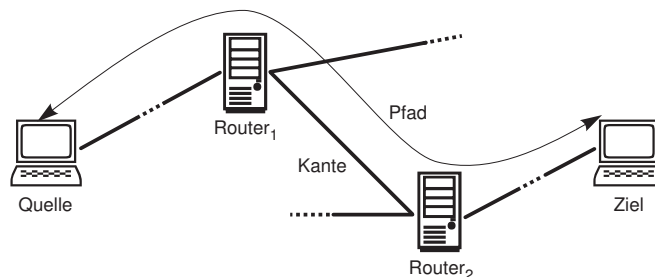


Abbildung 1: Beispielhafte Topologie eines Netzwerks mit Quelle und Ziel

Wie in **Abbildung 1** dargestellt, betrachten wir eine *Quelle* und ein *Ziel* und interessieren uns für die Bandbreite zwischen den beiden Rechnern. Auf dem Weg dazwischen, dem *Pfad*, liegt eine unbekannte Anzahl an *Routern*, die über *Kanten* (*Links*) miteinander verbunden sind. Typischerweise ist die Bandbreite der Kanten in einem Netzwerk unterschiedlich.

## 3 Messung der Bandbreite

### 3.1 Motivation

Wie bereits erwähnt sind in den Protokollen, auf denen heutige Netzwerkkommunikation basiert, kaum Möglichkeiten vorhanden, die Bandbreite direkt zu bestimmen.

Carter und Crovella [CarterCrovella96] zeigten außerdem, dass man die Länge eines Pfades in einem Netzwerk sehr schlecht durch die Anzahl an Netzwerkknoten (*Hops*) dazwischen abschätzen kann. Dazu verglichen sie die Anzahl an *Hops* zu 5825 Webservern (gemessen durch *traceroute*) mit der Latenzzeit zu dem jeweiligen Server (gemessen durch *ping*). **Abbildung 2 (a)** zeigt die Verteilung der Anzahl der *Hops*, **Abbildung 2 (b)** die der Latenzzeiten. Wenn die Anzahl der *Hops* wirklich direkten Einfluss auf die Latenzzeit hätte, würde man eine Ähnlichkeit der beiden Diagramme erwarten. Dies ist jedoch nicht der Fall.

Die Anzahl der *Hops* und die Latenz bieten außerdem nur eine relativ ungenaue Abschätzung der Bandbreite. Um zuverlässige Aussagen über die Bandbreite zu einem bestimmten Rechner treffen zu können, sind also andere Methoden nötig.

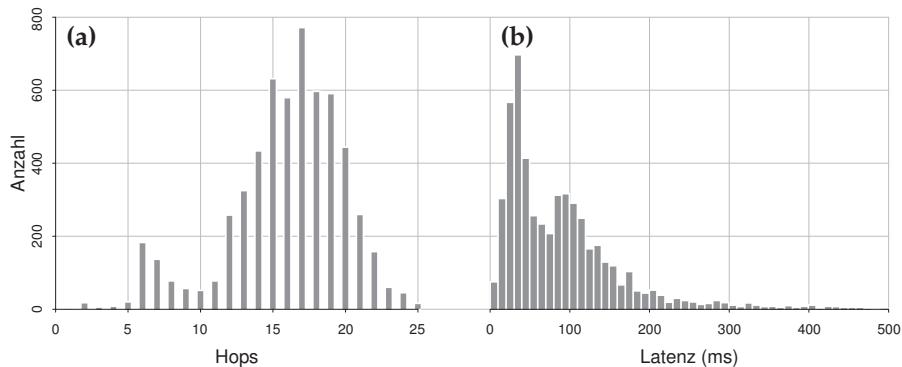


Abbildung 2: (a) Anzahl Hops und (b) Latenzzeiten zu 5825 Webservern

### 3.2 Anforderungen

Damit jedes Benutzerprogramm von den Ergebnissen der Messungen profitieren kann, sollte die Messung auf Anwendungsebene erfolgen. Dabei sollte die Messung so konzipiert sein, dass sie möglichst keinen negativen Einfluss auf das Netzwerk und den zu sondierenden Rechner hat. Der durch die Messung verursachte Overhead sollte minimal sein. Des Weiteren sollte die Messung so beschaffen sein, dass sie keine Änderungen am zugrunde liegenden Netzwerk erfordert. Im Speziellen bedeutet dies, dass keine besondere Software auf den beteiligten Netzwerkkomponenten installiert werden muss. Das Verfahren sollte außerdem robust und stabil unter sich (schnell) ändernden Bedingungen sein, wie beispielsweise in drahtlosen Netzwerken, wo die verfügbare Bandbreite stark von äußeren Faktoren abhängt. Es sollte möglichst genau sein und zeitnah die Ergebnisse zurückgeben.

Einige der hier beschriebenen Anforderungen sind idealisiert und in der Realität nur schwer oder überhaupt nicht einzuhalten. Jedoch geben diese Anforderungen an, in welche Richtung man forschen sollte, um noch bessere Methoden zu entwickeln.

### 3.3 Aktives und passives Sondieren

Es gibt verschiedene Methoden, die Bandbreite zu messen. Beim *aktiven Sondieren* (*out-of-band probing*) werden spezielle Sondierungspakete verwendet, die in den meisten Fällen auf dem *Internet Control Message Protocol* (ICMP) [RFC792] basieren. Dabei sendet man ICMP ECHO-Pakete an den zu sondierenden Rechner, der dann mit entsprechenden ICMP REPLY-Paketen gleicher Größe antwortet. Vorteilhaft an diesem Vorgehen ist, dass das ICMP theoretisch von allen IP-fähigen Netzwerkkomponenten unterstützt werden sollte. Der Nachteil ist, dass dadurch zusätzlicher Netzwerkverkehr entsteht.

Alternativ kann man auch eine *passive Sondierung* (*in-band probing*) verwenden. Hierbei werden keine zusätzlichen Pakete gesendet, sondern die vorhandenen Protokolle so modifiziert, dass die Datenpakete selbst zur Abschätzung der Bandbreite verwendet werden können. Es wird also nicht aktiv gemessen, sondern mehr beobachtet. Der Nachteil ist, dass Modifikationen des Protokolls möglicherweise an allen beteiligten Netzwerkkomponenten erforderlich sind.

Diese Arbeit konzentriert sich auf das aktive Sondieren durch ICMP-Pakete. Kapoor et al. [KapoorEtAl04] führen zwar an, dass man zur passiven Sondierung das UDP oder ein entsprechend modifiziertes TCP verwenden könnte, ihr Programm *CapProbe* basiert jedoch, wie die Programme *bprobe* und *cprobe* von Carter und Crovella, auf aktiver Sondierung.

### 3.4 Einflüsse auf die Bandbreite

Grundsätzlich gibt es (neben der Latenzzeit) zwei Faktoren, die die Bandbreite auf einem Netzwerkpfad bestimmen: die zugrunde liegende Basisbandbreite und die Auslastung der langsamsten Kante.

Unter *Basisbandbreite*  $b_{basis}$  versteht man die Bandbreite der langsamsten Kante auf einem bestimmten Pfad zwischen zwei Rechnern. Da es sich dabei um die geschwindigkeitslimitierende Kante handelt, sprechen Carter und Crovella auch vom *bottleneck link* (*Flaschenhals*), Kapoor et al. nennen sie *narrow link*. Kapitel 4 beschäftigt sich intensiv mit der Messung der Basisbandbreite.

Der zweite limitierende Faktor ist die Auslastung des Flaschenhalses durch *competing traffic* bzw. *cross traffic*. Dabei handelt es sich um Datenverkehr, der von anderen Benutzern verursacht wird, und mit dem wir uns die vorhandene Netzwerkbandbreite teilen müssen. In Kapitel 5 werden wir die Auslastung des Flaschenhalses durch die zur Verfügung stehende Bandbreite  $b_{verf}$  charakterisieren.

Weiß man die Basisbandbreite  $b_{basis}$  und die verfügbare Bandbreite am Flaschenhals  $b_{verf}$ , so kann man die Auslastung des Flaschenhalses  $u$  berechnen<sup>1</sup>:

$$u = \frac{b_{basis} - b_{verf}}{b_{basis}}$$

## 4 Messung der Basisbandbreite

### 4.1 Voraussetzungen

Um ein Modell für die Messung der Basisbandbreite angeben zu können, treffen wir einige vereinfachende Annahmen und begründen deren Sinn.

Da wir im Folgenden die Zeit zwischen der Ankunft von zwei Netzwerkpaketen messen wollen, gehen wir davon aus, dass die Pakete in der Reihenfolge ankommen, in der wir sie versendet haben. Diese Annahme mag für das Internet nicht immer zutreffen. Jedoch ließen sich falsch geordnete Pakete (durch entsprechende Nummerierung) sehr leicht erkennen und die entsprechende Messung verwerfen.

Eine weitere sinnvolle Annahme ist, dass der Pfad zwischen der Quelle und dem Ziel stabil ist, zumindest für einige Sekunden. Insbesondere bedeutet dies, dass alle Sondierungspakete über den selben Pfad laufen. Nur so können überhaupt zuverlässige Aussagen aus den Ankunftszeitpunkten der Pakete abgeleitet werden. Diese Annahme ist für das Internet sinnvoll, da die Routing-Tabellen nur recht selten aktualisiert werden.

Weiterhin nehmen wir an, dass sich die langsamste Kante sowohl für die Hin- als auch für die Rückrichtung an der selben Stelle befindet. Diese Annahme gilt in der Realität nicht immer. Beispielsweise könnte bei einer Anbindung über ADSL die langsamste Kante auf dem Hinweg durch den lokalen Upstream bestimmt sein, während auf dem Rückweg der Upstream des sondierten Rechners eine obere Schranke für die Bandbreite bildet. Es ist jedoch möglich, diese Annahme durch ein anderes Design der Messung aufzuheben.

Kapoor et al. führen an, dass ihre Methode Probleme hat, die richtige Basisbandbreite zu ermitteln, wenn sich auf dem Pfad konstanter UDP-Datenverkehr (*User Datagram Protocol*, [RFC768]) befindet, denn dann kann nicht mehr entschieden werden, ob es sich um Datenverkehr handelt, oder ob die Bandbreite wirklich so gering ist. Dieser Aspekt wird in Kapitel 4.4 genauer erläutert. Wir gehen hiermit davon aus, dass sich der Datenverkehr zufällig zusammensetzt und die Auslastung schwankt.

---

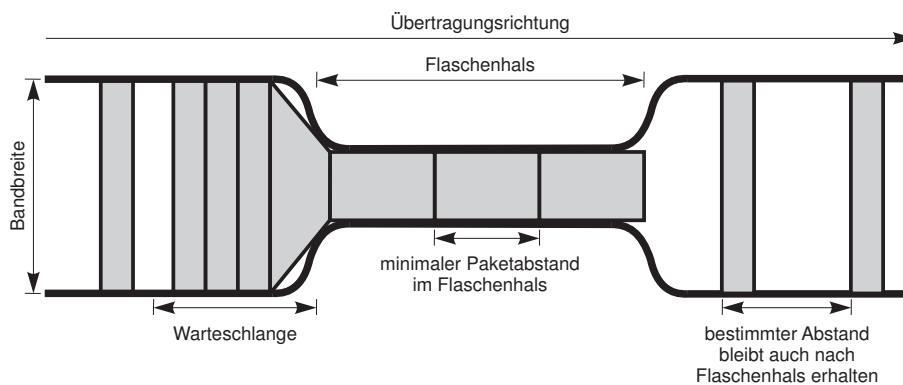
<sup>1</sup>Carter und Crovella verwenden hier die Formel  $u_{verf} = b_{verf}/b_{basis}$ , die den Anteil der verfügbaren Bandbreite und damit die *durch die Sondierung* am Flaschenhals entstandene Auslastung wiedergibt.

## 4.2 Verfahren

Die meisten Verfahren zur Messung der Basisbandbreite verwenden Paare von ICMP ECHO-Paketen. Dabei werden zwei solche Sondierungspakete  $p_1, p_2$  hintereinander gesendet. Anschließend wird der zeitliche Abstand zwischen der Ankunft der jeweiligen ICMP REPLY-Pakete  $t_{disp} = t_2 - t_1$  gemessen, die sogenannte *Dispersion*. Haben die Sondierungspakete eine Größe von  $l$ , so gilt für die Basisbandbreite  $b_{basis}$ :

$$t_{disp} = \frac{l}{b_{basis}} \quad \text{bzw.} \quad b_{basis} = \frac{l}{t_{disp}}$$

**Abbildung 3** veranschaulicht, was am Flaschenhals passiert. Vor dem Flaschenhals befindet sich ein Router, der die Pakete weiterversenden muss. Jedoch kommen die Pakete mit einer höheren Geschwindigkeit an, als sie versendet werden können (wir nehmen ja an, dass es sich hierbei um die langsamste Kante überhaupt auf dem Pfad handelt). Daher müssen die Pakete in die Warteschlange des Routers eingereiht werden.



**Abbildung 3:** Verarbeitung von Netzwerkpaketen am Flaschenhals

Nun gibt es verschiedene Szenarien:

- Die Sondierungspakete kommen direkt hintereinander am Flaschenhals an (und stehen somit direkt hintereinander in der Warteschlange). Dieser Fall ist der günstigste, da die Pakete hintereinander über die langsamste Kante gesendet werden. Aufgrund der Verarbeitungszeit verlassen die Pakete den Router mit einem größeren zeitlichen Abstand, als sie dort ankamen. Nach Passieren der langsamsten Kante entspricht die Dispersion zwischen zwei Sondierungspaketen  $p_1, p_2$  der Verarbeitungszeit des zweiten Pakets  $p_2$ .
- Zwischen den Sondierungspaketen liegen Pakete aus anderen Quellen. In diesem Fall verlassen die Pakete den Router nicht direkt hintereinander, sondern dazwischen werden andere Pakete bearbeitet. Dadurch erhöht sich die gemessene Dispersion  $t_{disp}$  um die Verarbeitungszeit der dazwischen liegenden Pakete  $\Delta t$ , die Basisbandbreite wird unterschätzt (*Dekompression*):

$$b'_{basis} = \frac{l}{t_{disp} + \Delta t} < b_{basis}$$

Dieser störende Effekt kann in jeder Warteschlange auf dem weiteren Pfad auftreten. Nehmen wir beispielsweise an, die Pakete haben nach dem Flaschenhals eine Dispersion  $t_{disp}$ . Dann können in der Zeit zwischen der Ankunft des ersten und des zweiten Pakets an einem Router *nach* dem Flaschenhals (die ja genau

$t_{disp}$  entspricht) andere Pakete bearbeitet werden, was wiederum zu einer Dekompression führt. Dieser Zeitraum wird von Kapoor et al. daher *vulnerability window* (Verwundbarkeits-Zeitraum) genannt.

- Das erste Paket muss in einer beliebigen Warteschlange nach dem Flaschenhals um  $\Delta t$  länger warten als das zweite. Es wird eine kleinere Dispersion gemessen, was zu einer Überschätzung der Basisbandbreite führt (*Kompression*):

$$b''_{basis} = \frac{l}{t_{disp} - \Delta t} > b_{basis} \quad (1)$$

Jetzt wird auch klar, dass ein Paar  $(p_1, p_2)$  von Sondierungspaketen die gleiche Größe  $l_1 = l_2 = l$  haben sollte, da es sonst durch die unterschiedlichen Verarbeitungszeiten (die proportional zur Paketgröße sind) automatisch zu Kompression ( $l_1 > l_2$ ) bzw. Dekompression ( $l_2 > l_1$ ) kommen würde, das Messergebnis wäre nutzlos.

**Bemerkung:** Unter Annahme der Stabilität des Pfades laufen die Sondierungspakete zwar zweimal über den Flaschenhals (einmal als ECHO- und einmal als REPLY-Paket), beim zweiten Mal sollte sich die Dispersion aber nicht mehr ändern – unter der Annahme, dass keine Verfälschungen (Dekompression, Kompression) aufgetreten sind.

### 4.3 Filterung und Konvergenzbeschleunigung

Führt man die Messung nur einmal durch und berechnet daraus die Basisbandbreite, so kann das Ergebnis stark vom wahren Wert abweichen. Die Ursachen dafür sind vielseitig. Beispielsweise könnten, wie in Kapitel 4.2 angesprochen, zwischen den Sondierungspaketen Pakete aus anderen Quellen bearbeitet worden sein.

Um die Messgenauigkeit zu erhöhen, werden mehrere Messungen hintereinander ausgeführt. Es werden also Folgen von Sondierungspaketen, sogenannte *packet trains*  $p_1, \dots, p_n$  gesendet, und die Dispersion zwischen je zwei dieser Pakete gleicher Größe  $t_{disp_i} = p_{i+1} - p_i, 1 \leq i < n$  gemessen.

Außerdem ist eine weitere Verarbeitung der Messwerte nötig, bevor man ein Ergebnis ausgibt. Carter und Crovella verwenden in ihrem Programm *bprobe* eine nachgeschaltete Filterung, um ungültige Messwerte auszusortieren. Es werden je 10 Sondierungspakete gesendet, und die dazwischenliegenden 9 Dispersionen gemessen. Anschließend wird nach Korrelationen zwischen den Messwerten gesucht. Das ganze wird mehrmals mit jeweils größerer Paketgröße durchgeführt. Es konnte festgestellt werden, dass die Messwerte eine um so stärkere Korrelation zeigten, je höher die Paketgröße war. Dennoch ist es wichtig, auch mit kleineren Paketen zu messen, damit die Basisbandbreite nicht unter- oder überschätzt wird. Eine Erklärung dafür folgt später in diesem Kapitel. Führt man mehrere Messungen mit verschiedener Paketgröße hintereinander aus, kann man die Ergebnisse überlagern und den Wert zurückgeben, der in der Vereinigung der Messwerte der größte ist, sofern eine genügende Anzahl an Intervallen dazu beiträgt.

Für Kapoor et al. ist es wichtig, dass die Messung „gute“ Messwerte enthält, das sind solche, bei denen keines der in einem Paar von Sondierungspaketen enthaltene Paket aufgrund von *cross traffic* in einer Warteschlange hat warten müssen. Offensichtlich gibt die Dispersion dieser Paare die exakteste Abschätzung für die Basisbandbreite wieder. Jedoch sinkt die Wahrscheinlichkeit, einen solchen Messwert zu erhalten, je höher die Auslastung der Netzwerkverbindung ist.

Um einen „guten“ Messwert zu erhalten, darf das zweite Paket in einem Paar von Sondierungspaketen nicht durch *cross traffic* in eine Warteschlange nach dem Flaschenhals eingereiht werden. Die Wahrscheinlichkeit dafür wird kleiner, wenn man die Paketgröße reduziert. Die Ursache dafür ist, dass kleinere Pakete schneller verarbeitet werden

können, was in einer kleineren Dispersion und damit einem kleineren *vulnerability window* resultiert. Die Wahrscheinlichkeit dafür, dass das erste Paket eines Paares „warten“ muss, ist jedoch unabhängig von dessen Größe. Durch kleinere Pakete wird also das Risiko einer Unterschätzung der Basisbandbreite reduziert, da in dem kleineren *vulnerability window*, d.h. der Bearbeitungszeit des zweiten Pakets desselben Paares, weniger *cross traffic* intervenieren kann.

Jedoch können kleinere Pakete auch zu einer Überschätzung der Basisbandbreite führen. Nehmen wir beispielsweise an, dass das erste Paket  $p_1$  in einem Paar von Sondierungspaketen bereits in einer Warteschlange nach dem Flaschenhals angekommen ist und dass sich das zweite Paket  $p_2$  in der Warteschlange vor dem Flaschenhals befindet. Die Wahrscheinlichkeit, dass  $p_1$  „warten“ muss, ist wie oben beschrieben unabhängig von dessen Größe. Je kleiner die Paketgröße allerdings ist, desto schneller wird  $p_2$  am Router vor dem Flaschenhals verarbeitet, da die Verarbeitungszeit proportional zur Paketgröße ist. Also wird die Wahrscheinlichkeit einer Kompression erhöht, wodurch das Paar im Durchschnitt näher „zusammenrückt“. Das kann zu einer Überschätzung der Basisbandbreite führen, da  $\Delta t$  im Verhältnis zu  $t_{disp}$  größer bei kleineren als bei größeren Paketen ist (vgl. Formel (1)).

Kapoor et al. bilden die Summe über alle Dispersionen eines *packet train* und nennen diese Summe *delay sum*. Wurde eines der Pakete in eine Warteschlange eingereiht, so ist diese Summe größer als die minimale *delay sum*. Das Vorgehen besteht also darin, die minimale *delay sum* über mehrere Messungen hinweg zu suchen. Diese gibt dann annähernd die korrekte Basisbandbreite wieder, da anzunehmen ist, dass dann nur sehr wenige Sondierungspakete in eine Warteschlange eingereiht wurden. Kapoor et al. weisen darauf hin, dass eine minimale *delay sum* nicht bedeuten muss, dass die einzelnen Dispersionen jeweils minimal waren. Um dies auszugleichen, lassen sie ihr Verfahren erst terminieren, wenn das Ergebnis zweier aufeinander folgender *packet trains* mit verschiedener Paketgröße um weniger als 5% abweicht.

Außerdem beschreiben Kapoor et al. verschiedene wahrscheinlichkeitstheoretische Modelle zur Abschätzung der Anzahl an Sondierungen, die nötig sind, um einen einzelnen „guten“ Messwert zu erhalten. Nehmen wir an, dass der Ankunftsprozess der Sondierungspakete am Flaschenhals einem Poisson-Prozess entspricht. Um einen „guten“ Messwert zu bekommen, darf weder das erste Paket in Folge von *cross traffic* in einer Warteschlange nach dem Flaschenhals warten müssen (d.h. bei der Ankunft des ersten Pakets muss die jeweilige Warteschlange leer sein), noch dürfen Pakete aus anderen Quellen zwischen dem Paar der Sondierungspakete ankommen. Nehmen wir an, dass die Anzahl der Ankünfte von *cross traffic*-Paketen Poisson-verteilt ist.  $\lambda$  sei die Ankunftsrate der Pakete des gesamten *cross traffic* am Flaschenhals,  $\mu$  die Verarbeitungsrate des Routers vor dem Flaschenhals und  $\tau$  die Dispersion des Paares. Die Wahrscheinlichkeit  $p_{leer}$ , dass das erste Paket auf eine leere Warteschlange trifft, ist gegeben durch:

$$p_{leer} = 1 - \frac{\lambda}{\mu} \quad (2)$$

Die Wahrscheinlichkeit  $p_{alleine}$ , dass zwischen dem Paar von Sondierungspaketen keine weiteren Pakete ankommen, ist gegeben durch:

$$p_{alleine} = P(\text{cross traffic-Zwischenankunftszeit} > \tau) = e^{-\lambda\tau} \quad (3)$$

Für die Wahrscheinlichkeit, einen einzelnen „guten“ Messwert zu bestimmen, gilt dann also:  $p_{gut} = p_{leer} \cdot p_{alleine}$ . Aufgrund der Annahme eines Poisson-Ankunftsprozesses ergibt sich als mittlere Anzahl  $n_{Poisson}$  der nötigen Messwerte, um einen „guten“ Messwert zu erhalten:

$$n_{Poisson} = \sum_{k=1}^{\infty} \binom{k}{1} \cdot p_{gut} \cdot (1 - p_{gut})^{k-1} = \sum_{k=1}^{\infty} k \cdot p_{gut} \cdot (1 - p_{gut})^{k-1} = \frac{1}{p_{gut}} \quad (4)$$

Kapoor et al. beobachteten, dass  $n_{Poisson}$  für *cross traffic*-Paketgrößen von mehr als 500 Bytes selbst bei hoher Auslastung (d.h. viel *cross traffic* und damit ein größeres  $\lambda$ , vgl. Formel (2) und (3)) in einem Bereich um 10 bleibt, während kleinere Paketgrößen ein sehr hohes  $n_{Poisson}$  produzierten. Dieser Effekt erklärt sich durch die bei kleineren Paketen höhere Wahrscheinlichkeit, dass zwischen zwei Sondierungspaketen ein „fremdes“ Paket am Flaschenhals eintrifft, wodurch  $p_{leer}$  und in Folge  $p_{gut}$  sinkt (vgl. Formel (4)).

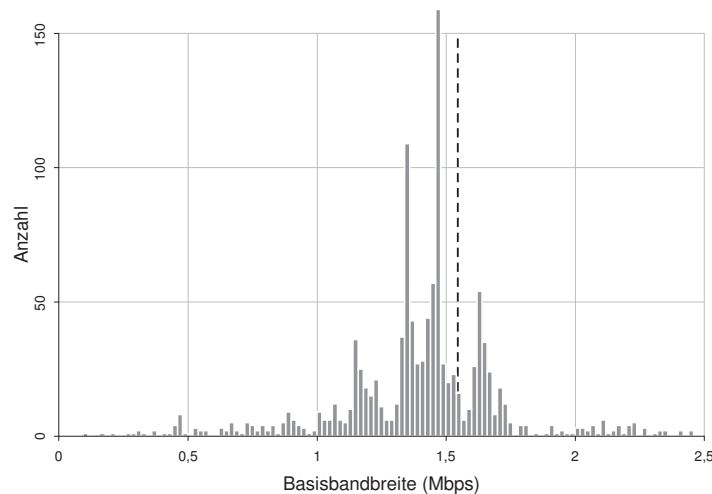
Kapoor et al. stellten noch zwei weitere Verfahren zur Abschätzung der nötigen Anzahl an Messungen vor: Annahme von deterministischem und von Pareto On/Off *cross traffic*. Die Ergebnisse blieben hierbei auch für hohe Auslastungen im Mittel unter 15 Messungen, um eine gute Messung zu erhalten.

#### 4.4 Validierung

Um die Anwendung *bprobe* zu validieren, wurden von einem lokalen Rechner aus Rechner in drei verschiedenen Typen von Netzwerken sondiert. Die Bandbreite der Flaschenhälse war jeweils bekannt. Die sondierten Netzwerktypen waren ein lokales Netzwerk, ein regionales Netzwerk und ein großräumiges regionales Netzwerk (*Wide Area Network, WAN*). Carter und Crovella führten über 4 Tage hinweg jede Minute eine Messung mit *bprobe* durch.

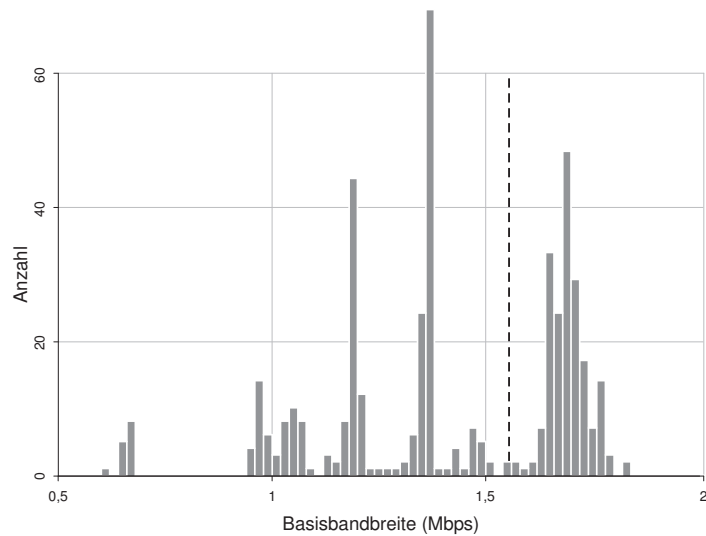
Die Tests im lokalen Netzwerk waren erwartungsgemäß relativ präzise. Ein Teil der sondierten Rechner hatte einen bekannten Flaschenhals von 56kbps, der Rest hatte einen Flaschenhals von 10Mbps. Die Messwerte zeigten eine starke Gruppierung um den korrekten Wert: 96% der Messungen wichen um höchstens 10% vom realen Wert ab.

Danach wurde im regionalen Netzwerk getestet. Hier gab es Rechner mit einem bekannten Flaschenhals von 56kbps, 1,54Mbps (T1) und 10Mbps. Die Messungen der 56kbps- und Ethernet-Hosts waren relativ exakt. Bei der Sondierung der Rechner mit einem Flaschenhals von 1,54Mbps gab es jedoch mehrere Häufungspunkte, wie **Abbildung 4** zeigt. Die gestrichelte Linie gibt die wahre Bandbreite wieder.



**Abbildung 4:** Sondierung im regionalen Netzwerk mit Flaschenhals 1,54Mbps (T1)

Schließlich folgte die Validierung im WAN. Die sondierten Hosts waren durchschnittlich 16 Hops entfernt vom sondierenden Rechner. Die bekannten Basisbandbreiten lagen bei 56kbps, 1,54Mbps (T1) und 10Mbps. Auch hier verteilten sich vor allem die Messwerte zu den Rechnern mit T1-Anbindung auf mehrere Häufungspunkte, wie **Abbildung 5** zeigt. Eine mögliche Erklärung dazu folgt später.



**Abbildung 5:** Sondierung im WAN mit Flaschenhals 1,54Mbps (T1)

Insgesamt zeigte sich in den meisten Fällen bei *bprobe* eine leichte Unterschätzung der Basisbandbreite. Carter und Crovella begründen dies damit, dass der Overhead durch die Kapselung der Pakete auf den verschiedenen Ebenen der Übertragung (vgl. ISO/OSI-Modell) nicht mit eingerechnet wurde. Eine einfache Lösung des Problems wäre eine Tabelle mit häufigen Basisbandbreiten und Rundung des Ergebnisses auf den entsprechend nächsten Eintrag in dieser Tabelle. Die Ursache für die unterschiedlichen Häufungspunkte bei den T1-Verbindungen begründen Carter und Crovella mit *cross traffic*, der aus kleinen Paketen mit nahezu gleicher Größe besteht. Wenn diese Pakete auf dem Pfad, der zur Sondierung verwendet wird, häufig genug auftreten, dann können die dadurch generierten Unterschätzungen der Basisbandbreite vom Filter nicht erkannt werden.

Kapoor et al. validierten ihr Programm *CapProbe* in einem Netzwerk mit 6 Hops zwischen den an der Sondierung beteiligten Rechnern. Unter Veränderung der Auslastung durch *cross traffic* mit einer Größe von 500 Bytes wurden über 100 Sekunden hinweg Messungen durchgeführt. Bei einer Sondierungspaketgröße von 200 Bytes und 500 Bytes und TCP-*cross traffic* lagen die gemessenen Werte selbst bei sehr hoher Auslastung sehr nahe an den wahren Werten. Bei Verwendung von UDP-*cross traffic* waren die Messwerte bei einer Auslastung des Flaschenhalses ab 50% nicht mehr so gut, was in Kapitel 4.1 bereits angesprochen wurde. Die Ursache dafür ist, dass sich UDP-Datenverkehr nicht automatisch an die zur Verfügung stehende Bandbreite anpasst.

Ein Vergleich mit zwei anderen Anwendungen, *pathchar* und *pathrate*, zeigte wie gut *CapProbe* die Basisbandbreite wirklich abschätzt. Es wurden die Basisbandbreiten zu verschiedenen Universitäten auf der ganzen Welt gemessen. Dabei lag der bekannte Flaschenhals bei 100Mbps, 11Mbps (WLAN) oder 1,5Mbps. *CapProbe* konvergierte in allen Testfällen zu 10 Universitäten weltweit innerhalb von maximal 25 Sekunden gegen den richtigen Wert. Lediglich im Fall der WLAN-Verbindung wurde eine niedrigere Basisbandbreite ermittelt, was aber mit hoher Wahrscheinlichkeit mit der Länge der entsprechenden Funkstrecke zu tun hatte. In den meisten anderen Fällen lag der Fehler bei unter 5%. *pathrate* und *pathchar* dagegen konvergierten bei Testmessungen zu 4 Universitäten weltweit in den meisten Fällen sehr viel langsamer (oft mehrere Minuten oder sogar Stunden im Fall von *pathchar*). Außerdem war die Genauigkeit selbst bei den langen Messzeiten nur recht ungenau (teilweise Abweichungen bis zu 80%).

## 5 Messung der Pfadauslastung

### 5.1 Verfahren

Carter und Crovella beschrieben ein Konzept, wie man auf einfache Weise die verfügbare Bandbreite auf einem Pfad messen kann. Dabei wird ein kurzer *packet train* von  $n$  ICMP ECHO-Paketen zum sondierenden Rechner gesendet, und die Zeit  $t_{total}$  zwischen der Ankunft des ersten und des letzten ICMP REPLY-Pakets gemessen:

$$t_{total} = (t_n - t_{n-1}) + (t_{n-1} - t_{n-2}) + \dots + (t_2 - t_1) = t_n - t_1$$

Sei die Größe der gesendeten Pakete jeweils gleich  $l$  (d.h. alle Pakete sind gleich groß). Nimmt man an, dass die Pakete mit einer höheren Geschwindigkeit gesendet werden konnten als die Bandbreite des Flaschenhalses, so kann man die verfügbare Bandbreite auf dem Flaschenhals  $b_{verf}$  mit dem Durchsatz abschätzen<sup>2</sup>:

$$b_{verf} \sim \frac{l \cdot n}{t_{total}}$$

**Bemerkung:** Auch hier kann durch Pakete aus anderen Quellen zwischen den Sondierungspaketen Dekompression auftreten, wodurch die zur Verfügung stehende Bandbreite unterschätzt wird.

### 5.2 Validierung

Carter und Crovella stießen bei der Verwendung dieses Verfahrens allerdings auf Schwierigkeiten. Die Messwerte waren teilweise zu ungenau. Besonders Verzögerungen beim Senden der Pakete und blockweise Weiterleitung an die Anwendung beim Empfang der Pakete beeinflussten das Messergebnis negativ. Diese Effekte hängen unter anderem vom Betriebssystem ab, das intern Warteschlangen zur Kommunikation mit der entsprechenden Hardware verwendet. Um die Probleme zu lösen, wurde folgendes Verfahren verwendet: Die kleinste und die größte Dispersion zwischen zwei Paketen im *packet train* geht jeweils nicht in das Ergebnis ein, d.h. die korrigierte Zeitspanne  $t_{korr}$  ergibt sich zu

$$t_{korr} = t_{total} - t_{mindisp} - t_{maxdisp}$$

wobei  $t_{mindisp}$  und  $t_{maxdisp}$  die minimale bzw. maximale Dispersion kennzeichnen. Außerdem wurden 4 Messungen mit je 8 Paketen durchgeführt, um verlorene Pakete oder ungültige Messungen, die durch in der falschen Reihenfolge empfangene Pakete verursacht wurden, auszugleichen. Es zeigte sich, dass die Messungen durch diese Anpassungen genauer wurden.

Ein weiteres Problem war die Validierung des Konzeptes, denn diese war nur im lokalen Netzwerk möglich, wo der Datenverkehr kontrollierbar ist, der die Auslastung am Flaschenhals verursacht. Hier zeigte sich, dass das entsprechende Programm mit dem Namen *cprobe* Messwerte mit einer Abweichung von 10% lieferte. In Netzen wie dem Internet ist die Validierung jedoch um ein vielfaches komplizierter, da man den Datenverkehr hier nicht so leicht kontrollieren kann.

---

<sup>2</sup>Auch hier muss wieder der Overhead durch die Kapselung der Pakete einberechnet werden, vgl. Kapitel 4.4

## 6 Auswertung

Es hat sich gezeigt, dass die Programme *bprobe*, *cprobe* und *CapProbe* nützliche Werkzeuge zur Bestimmung der Basisbandbreite bzw. verfügbaren Bandbreite sind. *CapProbe* baut dabei auf einem mathematischen Fundament auf, das Kapoor et al. sehr detailliert ausgearbeitet haben. Carter und Crovella vernachlässigen die Mathematik dahinter etwas und konzentrieren sich eher auf durch Messungen gewonnene Fakten.

Wenn wir die Ergebnisse mit den in Kapitel 3.2 beschriebenen Anforderungen vergleichen, so können wir feststellen, dass die Programme auf Anwendungsebene laufen, die Messung keine vorherige Änderung am Netzwerk erfordert und die Verfahren relativ robust und stabil sind. Die Ergebnisse werden schnell und zeitnah zurückgegeben.

Leider haben die Programme aber immer noch einen zu hohen Bandbreitenverbrauch, da sie, um Konvergenz zu erreichen, eine ausreichende Anzahl an Messungen benötigen. Außerdem besteht eine durch Eigenschaften des Betriebssystems ausgelöste Messungenauigkeit, wie etwa das Speichern von ankommenden Paketen in Warteschlangen und damit Verfälschung der gemessenen Ankunftszeiten. Leider gibt es für dieses Problem keine direkte Lösung, vielmehr ist nur eine Abschätzung der Ergebnisse durch teilweises Eliminieren von „extremen“ bzw. unrealistischen Messwerten möglich. Des Weiteren konnte festgestellt werden, dass konstanter Datenverkehr am Flaschenhals das Messergebnis verfälschen kann, da dadurch kontinuierlich zu lange Dispersionen gemessen werden und die Messwerte durch ihre Korrelation dadurch nicht als falsch erkannt werden können. Schließlich bleibt noch darauf hinzuweisen, dass nicht alle Rechner im Internet auf ICMP ECHO-Pakete reagieren. Dies kann beispielsweise sicherheitstechnische Gründe haben. Eine Messung der verfügbaren Bandbreite zu solchen Rechnern ist mit den hier vorgestellten Methoden leider nicht möglich.

Insgesamt kann man sagen, dass die vorgestellten Methoden noch einige Schwächen haben. Diese Schwächen beruhen zu einem großen Teil auf der Unberechenbarkeit der genannten Faktoren. Jedoch ist nicht abzusehen, ob sich die Genauigkeit der Messungen bei gleichzeitig angestrebter „schneller“ Konvergenz (im Rahmen von wenigen Sekunden, vgl. Kapoor et. al.) noch verbessern lässt.

## 7 Ausblick

Die Nutzung der durch die Messungen gewonnenen Daten steht im Vordergrund verschiedener aktueller Forschungen. Carter und Crovella wollten ihre Programme *bprobe* und *cprobe* zusammen mit dem Programm *ping* zur Bestimmung des aktuellen Netzwerkstatus für eine dynamische Webserverauswahl verwenden. Da Dokumente oft mehrfach im Internet vorhanden sind, könnte somit der Mirror mit der aktuell niedrigsten Last oder der höchsten Bandbreite ausgewählt und von diesem das Dokument bezogen werden, man könnte somit überladene Verbindungen aktiv meiden. Denkbar wäre in diesem Zusammenhang auch eine Browsererweiterung, die dem Benutzer durch die Farbe der Hyperlinks anzeigt, wie „gut“ die Verbindung zu dem Server ist, zu dem der Link führt. Zudem könnte man die Definition von Hyperlinks so erweitern, dass mehrere URLs (*Uniform Resource Locator*) pro Link zugelassen sind und von denen dynamisch wie oben beschrieben der „beste“ ausgewählt wird. Auch für Netzwerkbetreiber sind diese Messungen interessant, da sie Informationen über die Auslastung bestimmter Teile des Netzwerkes liefern und somit Schwachstellen in der Infrastruktur aufdecken.

Kapoor et al. fügen hinzu, dass das Wissen über die verfügbare Bandbreite zu einem Rechner interessant für Streaming-Anwendungen ist. So könnte ein Multimedia-Server erst die Bandbreite bestimmen und dann automatisch den passend aufgelösten Videostream selektieren und zum Client übertragen.

## 8 Zusammenfassung

In diesem Papier wurden verschiedene Verfahren zur Messung der Bandbreite zwischen zwei Rechnern im einem Netzwerk vorgestellt, darunter *bprobe* und *CapProbe* zur Messung der Basisbandbreite und *cprobe* zur Messung der verfügbaren Bandbreite. Dabei wurde auf Papieren von Carter und Crovella [CarterCrovella96] und Kapoor et al. [KapoorEtAl04] aufgebaut und die Methoden und Ergebnisse miteinander verglichen.

Es wurde zwischen der Messung der Basisbandbreite und der verfügbaren Bandbreite unterschieden und beschrieben, wie die Konvergenz der Verfahren beschleunigt werden kann. Außerdem wurden Modelle vorgestellt, wie man Datenverkehr anderer Benutzer im Netzwerk modellieren kann und welche Folgen sich daraus für die Konvergenz der beschriebenen Verfahren ergeben.

Beim Vergleich der Ergebnisse der genannten Autoren zeigt sich, dass die Programme unter bestimmten Annahmen eine recht gute Abschätzung für die verfügbare Bandbreite liefern. Jedoch haben sie auch noch mit Problemen zu kämpfen. Trotzdem sind die Ergebnisse genau und oftmals zeitnah genug, um die in Kapitel 7 angesprochenen Anwendungen realisieren zu können.

## Literatur

[RFC792] Network Working Group: *Request for Comments: 792 – Internet Control Message Protocol*; siehe <http://www.ietf.org/rfc/rfc0792.txt>, September 1981

[CarterCrovella96] Robert L. Carter, Mark E. Crovella: *Measuring Bottleneck Link Speed in Packet-Switched Networks*; Performance Evaluation, 27(8): 297–318, Oktober 1996; siehe <http://www.cs.bu.edu/techreports/1996-006-measuring-bottleneck-link.ps.Z>

[KapoorEtAl04] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, M. Y. Sana-didi: *CapProbe: A Simple and Accurate Capacity Estimation Technique*; Proceedings of SIGCOMM 2004 Conference on Computer Communications; siehe <http://www.acm.org/sigs/sigcomm/sigcomm2004/papers/p449-kapoor1111.pdf>

[RFC768] Network Working Group: *Request for Comments: 768 – User Datagram Protocol*; siehe <http://www.ietf.org/rfc/rfc0768.txt>, August 1980