Scalable Performance Signalling and Congestion Avoidance

Dissertation an der Technischen Universität Darmstadt
ISBN 1-4020-7570-7
Kurzzusammenfassung

Michael Welzl
Institut für Informatik
Technikerstr. 25, A-6020 Innsbruck, Österreich
michael.welzl@uibk.ac.at

Zusammenfassung Die vorliegende Dissertation beschreibt das "PTP" Protokoll, mit dem Performance-spezifische Informationen auf effiziente und skalierbare Weise von Routern abgefragt werden können. Das Ziel ist dabei, aufgrund dieser Informationen bessere Dienste zu realisieren als mit bisherigen Mitteln. Als ein besonderer solcher Dienst wird das "CADPC" Staukontrolleverfahren erläutert, das auf Basis der mit PTP abgefragten verfügbaren Bandbreite eine in vieler Hinsicht bessere Dienstgüte erzielt als herkömmliche Alternativen. Dies wird mit Simulationen unter verschiedenen Bedingungen belegt.

1 Einleitung

Die Staukontrolle von TCP stützt sich in erster Linie auf den gemessenen Paketverlust (oder binäre Stauinformation in Form eines Bits im IP-Paketheader); vereinfacht dargestellt, wird die Senderate bei jeder eintreffenden positiven Bestätigung linear erhöht und bei Erkennung eines Paketverlusts auf die Hälfte reduziert. Da auf diese Art das Netzwerk von weiteren, komplexen Arbeiten zur Unterstützung der Staukontrolle entlastet wird, ist TCP gut skalierbar. Das zugrunde liegende "AIMD" *(Additive Increase, Multiplicative Decrease)* Verfahren hat jedoch eine Reihe von Nachteilen:

- Die Stabilität eines TCP-Netzwerks ist bislang nur für vereinfachte Fälle bewiesen und wird von manchen Forschern zumindest im Fall von heterogenen Umlaufzeiten in Frage gestellt.
- Da die binäre Information *"es gab einen Stau"* unabhängig von der Kanalkapazität ist, kann die Rate immer nur auf die gleiche Weise reduziert werden — es entsteht ein "Sägezahn-Effekt", der zu einem mit der Kanalkapazität und der Umlaufzeit (dem sogenannten Bandbreiten×Delay-Produkt) ansteigenden Verlust an genützter Bandbreite führt. TCP weist also etwa bei Satellitenverbindungen ein schlechtes Verhalten auf.
- Dieser "Sägezahn-Effekt" bewirkt in jedem Fall regelmäßigen Paketverlust; auch, wenn es etwa nur einen einzigen Sender gibt. TCP überschreitet also ständig die verfügbare Bandbreite, um hinterher zu erfahren, daß es bereits an der Zeit gewesen wäre, die Senderate zu reduzieren.
- Wenn Übertragungskanäle verrauscht sind, führt dies im Normalfall zu Paketverlust aufgrund eines Prüfsummenfehlers; dies wird von TCP als ein Stauindikator fehlinterpretiert.
- Das TCP zugrunde liegende Staukontrolleverfahren ist zur Echtzeitübertragung von Multimediadaten nicht geeignet, da die Bandbreite ständigen Schwankungen unterliegt und damit schwer vorhersagbar wird. Anwendungen, die Multimediadaten an die verfügbare Bandbreite anpassen (sogenannte "adaptive Anwendungen"), müßten in diesem Fall solche Anpassungen ständig vornehmen. Für den Benutzer ist es jedoch vorteilhaft, wenn sich die Bandbreite (und damit die Qualität) auf einen einigermaßen konstanten Wert "einpendelt".

Aufgrund all dieser Probleme ging der vorliegenden Arbeit die Frage voraus: "Wäre es nicht besser, die verfügbare Bandbreite mit einfachen Mitteln abzufragen als die Grenzkapazität des Netzes zu 'erfühlen' (wie TCP es tut)?". Auf einen Punkt gebracht ist das wichtigste Ergebnis dieser Dissertation die Antwort auf diese Frage: "ja". Der Weg zu dieser Antwort besteht aus drei wesentlichen Schritten:

1. Der Entwurf und die Realisierung eines Protokolls, um die benötigten Informationen vom "Netz" abzufragen
2. Der Entwurf eines Verfahrens, das diese Informationen benutzt, um eine Verbesserung zu erzielen
3. Der Beweis, daß das neue Verfahren zumindest in einem speziellen Szenario besser ist als herkömmliche Alternativen

Diesen drei Teilbereichen sind die folgenden Abschnitte gewidmet.

2 Das PTP Framework

2.1 Das Performance Transparency Protocol

PTP ähnelt dem von ATM Netzwerken her bekannten "Explicit Rate Feedback" (dort Teil des "Available Bit Rate" Dienstes). Abgesehen davon, daß PTP für Netzwerke auf IP-Basis und nicht für ATM konzipiert wurde, ist der wichtigste Unterschied die bessere Skalierbarkeit: Router führen keine komplizierten Berechnungen durch und müssen sich auch keine einzelnen Datenflüsse merken, sondern schreiben im einfachsten Fall nur angefragte, bereits in der sogenannten "MIB" Datenbank zur Verfügung stehende Informationen in ein Paket. Wichtig ist auch, daß die Anzahl der PTP Pakete im Netzwerk (und damit auch die für PTP aufzuwendende Rechenleistung) durch einen konstanten Anteil am Gesamtverkehr beschränkbar ist; auf diese Art wird gewährleistet, daß der gesamte PTP-Verkehr im Netz einen bestimmten Anteil am Gesamtverkehr nicht überschreitet — PTP skaliert also linear mit der Nutzlast.

PTP zeichnet sich durch seine Generizität aus: während es grundsätzlich für die Abfrage der verfügbaren Bandbreite entwickelt wurde, kann es auch eingesetzt werden, um andere performance-spezifi sche Informationen abzufragen (sogenannte "Performance Parameter"); dabei kann es sich z.B. um die Paketgröße, die mittlere Queue-Länge am Bottleneck-Router oder den maximalen auftretenden Signalrauschabstand handeln. In der Dissertation werden einige dieser Performance Parameter beschrieben und deren Einsatzmöglichkeiten skizziert; die Verwendung von PTP wird jedoch nur im Fall der verfügbaren Bandbreite durch Simulationen gerechtfertigt. PTP bezeichnet sowohl das Protokoll selbst als auch das sich aus dem Protokoll und den "Performance Parametern" ergebende Framework. Abbildung 1 zeigt die zwei Modi, in denen PTP benutzt werden kann:

Forward Packet Stamping: Hier wird ein Paket aktualisiert, während es auf dem Weg von einem Sender zu einem Empfänger ist. Der Empfänger schickt die Information daraufhin zurück an den Sender.

Direct Reply: In diesem Modus beinhaltet ein Paket die Performance Anforderung der Quelle; ein Router, der dieser Anforderung nicht genügt, aktualisiert das Paket und sendet es zurück an den Absender (die Richtung des Pakets wird durch einzelne Bits im Header unterschieden).

Direct Reply kann effi zienter sein als Forward Packet Stamping, da es die Umlaufzeit für das Feedback drastisch abkürzen kann (man denke dabei nur etwa an die Möglichkeit, Feedback zu senden, bevor ein Paket eine Satellitenverbindung passiert); es ähnelt jedoch Mechanismen wie etwa "BECN", die in der IETF aus mehreren Gründen (größere Belastung für Router, Ignorieren des Empfängers, ..) äußerst kritisch betrachtet werden. Deshalb ist die routerseitige Unterstützung von Direct Reply optional.

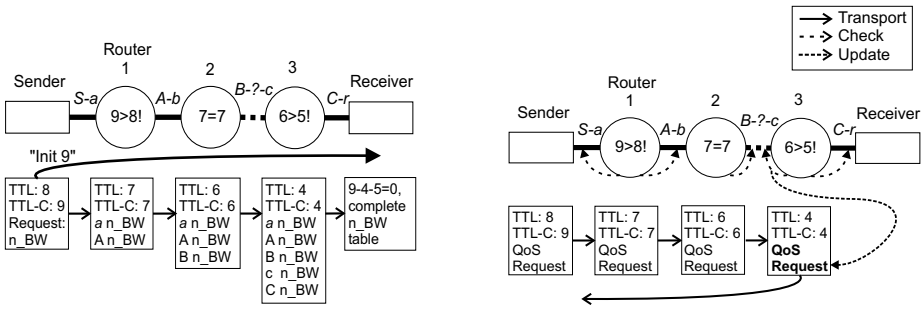

Abbildung 1. Die Arbeitsweise von PTP

In Abbildung 1 ist weiters die Funktionsweise des TTL_Check-Mechanismus von PTP dargestellt: nachdem jeder Router das TTL Feld im IP Header um mindestens 1 verringert, kann durch Kopieren und Vergleichen dieses Feldes erkannt werden, ob zwischen zwei PTP-Routern einige Router waren, die das Protokoll nicht unterstützen. In diesem Fall werden zusätzlich die Informationen der eingehenden Verbindung verwendet; jeder PTP-Router kann also fuer maximal einen "fremden" Router kompensieren.

2.2 Performance Parameter und Content Types

Um die zuvor erwähnten Performance Parameter zu erhalten, müssen sogenannte "Content Types" (Inhaltstypen) für das PTP Protokoll spezifi ziert werden; ein Performance Parameter errechnet sich aus den Informationen, die durch einen oder mehrerer dieser Content Types übermittelt werden. Der wichtigste Performance Parameter in der Dissertation ist "Available Bandwidth", die dazu gehörigen Content Types beinhalten folgende Informationen:

- Die Adresse des Router-Interfaces
- Eine Zeitmarke
- Die nominelle Linkbandbreite (die maximale Bandbreite, die für Datenflüsse von einem bestimmten eingehenden Link zu einem bestimmten ausgehenden Link zur Verfügung gestellt wird)
- Ein Bytezähler

Da es nicht möglich ist, zwei Zähler zu vergleichen, funktionieren die "Available Bandwidth" Content Types nur im Forward Packet Stamping Modus. Der Empfänger bildet dabei aus zwei nacheinander eintreffenden PTP Paketen eine Tabelle der verfügbaren Bandbreiten an allen Routern und schickt die nötigen Informationen des Bottlenecks (das ist der Router mit der geringsten verfügbaren Bandbreite) zurück an den Sender. Der vielleicht schwierigste Teil ist es nun, die Informationen dort möglichst optimal zu nützen.

3 Congestion Avoidance with Distributed Proportional Control (CADPC)

3.1 Design

Das Staukontrollverfahren auf Basis der PTP Daten wurde in einfachen Schritten entworfen:

- Zuerst wurde ein sinnvoll erscheinendes, verwandtes Verfahren gefunden und entsprechend erweitert.

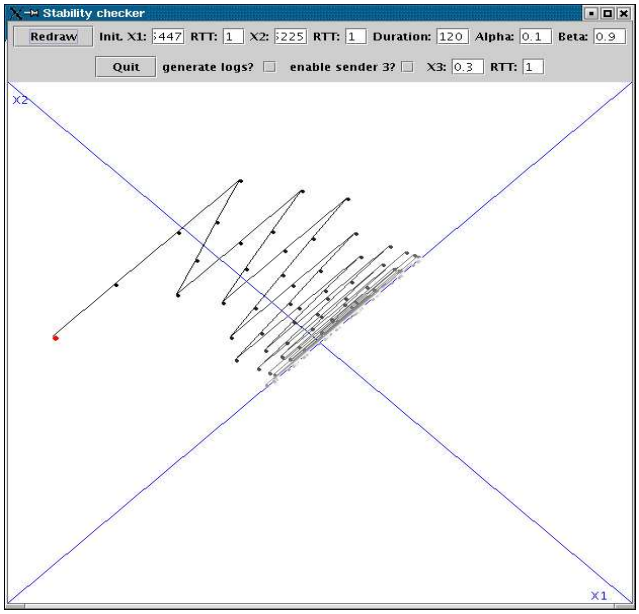

Abbildung 2. Der Vektor Diagramm Simulator als Entwurfswerkzeug

– Dann wurde unter Zuhilfenahme von Vektor Diagrammen geprüft, ob es im einfachen Fall von zwei Sendern mit synchronen Umlaufzeiten und einer Ressource zu einer fairen und effi zienten Auslastung führt.
– Im nächsten Schritt wurde mit dem in Abbildung 2 gezeigten Simulator das gleiche Szenario bei asynchronen Umlaufzeiten getestet.

Dann erst war es an der Zeit, das Verhalten im synchronen Fall mathematisch zu analysieren und im asynchronen Fall durch Simulationen zu überprüfen.

3.2 Analyse

Als verteilte Erweiterung eines ATM ABR Verfahrens namens *CAPC* bildet *CADPC* ("Congestion Avoidance with Distributed Proportional Control") eine stabile und effi ziente Möglichkeit, die Bandbreite auf Basis von PTP "Available Bandwidth" Daten anzupassen. Nach einigen Vereinfachungen ergibt sich folgende Darstellung für das Regelverhalten jedes Senders:

$$x_i(t+1) = x_i(t)\Big(2 - x_i(t) - \sum_{j=1}^{n} x_j(t)\Big) \tag{1}$$

$x_i(t)$ ist hier die mit der nominellen Bandbreite normierte Rate des Senders i zum Zeitpunkt t (die Zeit zwischen zwei PTP Meßintervallen — eine Zeiteinheit – und die Zielrate sind 1) und es sind n Sender involviert. Die Summe der Raten aller Sender bildet den von PTP zum Zeitpunkt t erhaltenen Verkehr.

Nachdem alle Sender das selbe Regelverhalten aufweisen, ergibt sich im vereinfachten zeitsynchronen Fall der gesamte Verkehr während des letzten Meßintervalls einfach als n mal der Rate eines Senders. Damit läßt sich Gleichung 1 auf die bekannte Gleichung für logistisches Wachstum zurückführen:

$$\dot{x}(t) = x(t)a\big(1 - x(t)/c\big) \tag{2}$$

Man weiß von dieser Gleichung, daß sie für $a > 0$ und $c > 0$ den instabilen Gleichgewichtspunkt $\overline{x} = 0$ und den asymptotisch stabilen Gleichgewichtspunkt $\overline{x} = c$ hat. Ersetzt

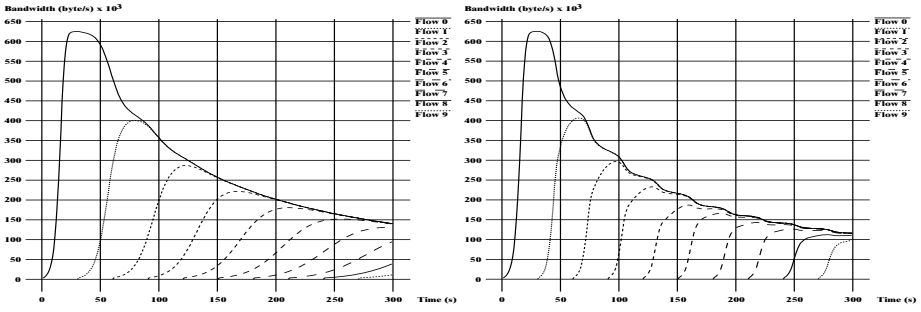

Abbildung 3. Ursprüngliches (links) und verbessertes CADPC Enschwingverhalten

man a durch 1 und c durch $1/(1+n)$, so erhält man Gleichung 1 bei stetigem Zeitverlauf. Bei diskretem Zeitverlauf muß a kleiner als 2 sein, damit der Mechanismus stabil bleibt; logistisches Wachstum hat eine typische "S-Form", die Glätte dieser Kurve kann über die Konstante a in einem gewissen Rahmen gesteuert werden.

Die Rate eines Senders konvergiert gegen c[1], der Gesamtverkehr konvergiert also gegen $nc = n/(1+n)$; diese Funktion konvergiert wiederum mit steigender Anzahl Sender schnell gegen 1. Auch wenn die Gesamtrate bei sehr wenigen Sendern niedrig ist, kann ein wesentlich besseres Verhalten als mit TCP erwartet werden: in echten Netzen sind Situationen mit etwa 2 oder 3 Sendern seltene Begebenheiten. Das genaue Verhalten eines Netzwerks mit PTP/CADPC wird anhand von Simulationen untersucht.

4 Evaluierung

Das Performance Transparency Protokoll wurde zur Durchführung einfacher Funktionalitätstests für Linux (sowohl als Router als auch als Endsystem) implementiert; zur genauen Bewertung wurde der "ns-2" Netzwerksimulator verwendet. Dabei stellte sich in Untersuchungen des dynamischen Verhaltens heraus, daß eine Abhängigkeit zwischen der Stabilität des Verfahrens und der verwendeten Paketgröße sowie der verfügbaren Bandbreite besteht. Zudem ist das anfängliche Einschwingverhalten in realistischen Fällen zu wenig aggressiv; dieses Problem wurde durch eine kleine Änderung (Skalierung des gemessenen Verkehrs mit der beobachteten Verkehrsänderung) behoben.

Die weiteren Untersuchungen wurden in i) Tests bezüglich des dynamischen Verhaltens und ii) Langzeittests zum Vergleich aus QoS-Sicht untergliedert.

4.1 Dynamisches Verhalten

Zwei Simulationsbeispiele sind in Abbildung 4 dargestellt: im linken Diagramm sieht man die auf der Empfängerseite einer Bottleneck Verbindung gemessene Gesamtbandbreite von 10 TCP und 10 CADPC Datenflüssen; dabei wurden jeweils Simulationen eines Typs getrennt durchgeführt. Alle Sender wurden zugleich gestartet, die Bandbreite der Bottleneck Verbindung war 10 Mbit/s. Offensichtlich ist die Rate von CADPC wesentlich "glätter" als die Rate von TCP.

Das rechte Diagramm zeigt die gleichermaßen gemessenen Raten von 10 CADPC Datenflüssen bei asynchronen Umlaufzeiten (jeweils Vielfache der Umlaufzeit des ersten Datenflusses). Auch in diesem Fall, der wesentlich komplexer ist als der zuvor mit dem Diagramm-basierten Simulator untersuchte, konvergiert die Rate der Datenflüsse auf den gleichen Wert: in etwa 10 Mbit/s / (1 + n) = 113636 byte/s. Weiters wurden folgende dynamischen Effekte untersucht:

[1] Ist die Anzahl der Benutzer bekannt, kann die konvergierte Rate direkt unter Verwendung eines hohen Werts für t mit der Gleichung $x(t) = \frac{c}{1+e^{-at}}$ errechnet werden.

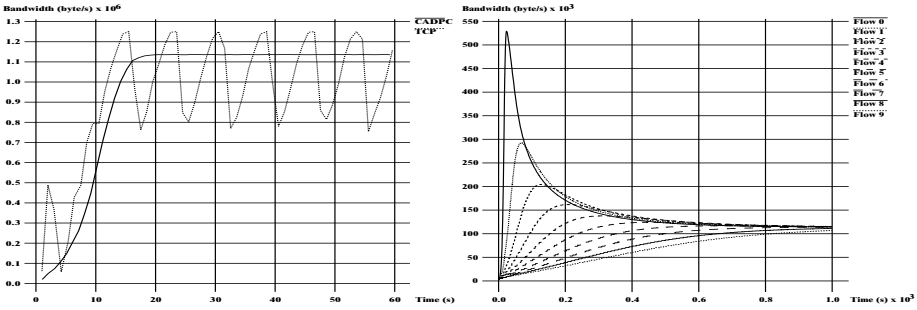

Abbildung 4. CADPC vs. TCP (links), Fairness von CADPC (rechts)

- Schwankungen der Queue bei TCP und CADPC Verkehr
- Ratenglätte von CADPC im Vergleich zu RAP, TFRC und TEAR (aus der Literatur bekannte Mechanismen, die fair gegenüber TCP sind).
- Der Einfluß von Routing auf TCP und CADPC (sowohl plötzliche Bandbreitenänderungen als auch das Mischen von bereits konvergierten Datenflüssen)
- Der Einfluß von verrauschten und stark asymmetrischen Verbindungen auf TCP und CADPC

4.2 Langfristige Leistung

Bei der langfristigen Evaluierung wurde immer das Verhalten von CADPC mit den drei TCP Varianten "Sack", "Reno" und "Vegas" (alle prinzipiell immer unter Verwendung von ECN) und den vier TCP gegenüber fairen Verfahren "RAP", "GAIMD" (eine besondere Form von AIMD), "TFRC" und "TEAR" verglichen. Die gemessenen Parameter waren in allen Fällen der Durchsatz, Paketverlust, die durchschnittliche Queue Länge am Bottleneck und die Fairness. Bis auf einzelne Ausnahmefälle wurden immer Simulationen eines Typs getrennt von anderen durchgeführt, alle Sender starteten zugleich und es gab nur eine einzige Bottleneck Verbindung.

Die Abbildungen 5 und 6 stellen ein Beispiel für die in der Dissertation beschriebenen Ergebnisse längerfristiger Simulationen dar. Es ist hier klar ersichtlich, daß CADPC in jedem Fall am wenigsten Verlust und die geringste durchschnittliche Queue Länge — also ein gutes Anpassungsverhalten — aufweist. Nur TFRC und TEAR hatten einen durchwegs höheren Durchsatz, dies allerdings auf Kosten einer höheren Paketverlustrate. Der höchste Durchsatz kann in diesem Simulationsszenario grundsätzlich erzielt werden, indem man mit einer sehr hohen Bandbreite sendet ohne sich an die verfügbare Bandbreite anzupassen — diese Metrik ist also kritisch zu betrachten. Viel Durchsatz ist nur ein gutes Zeichen, wenn er mit geringem Verlust einher geht.

Es sollte noch angemerkt werden, daß alle verwendeten Mechanismen bis auf CADPC jedes am Empfänger eintreffende Paket bestätigen, während CADPC selbst seine Berechnungen auf ein einziges PTP Paket stützt, das alle 4 Umlaufzeiten verschickt wird. CADPC generiert also wesentlich weniger zusätzlichen Signalisierungsverkehr als etwa TCP (was auch das bessere Verhalten bei stark asymmetrischen Verbindungen erklärt).

Neben den gezeigten Ergebnissen finden sich in der Dissertation noch folgende Bewertungen ("QoS" steht hier stellvertretend für Durchsatz, Verlust, mittlere Queue Länge und Fairness):

- Einhaltung des "max-min-fairness" Kriteriums bei zwei unterschiedlichen Bottleneck Verbindungen
- QoS in Abhängigkeit der Bottleneck Bandbreite
- QoS in Abhängigkeit von Delay
- QoS bei unterschiedlich starkem Web-ähnlichen Hintergrundverkehr

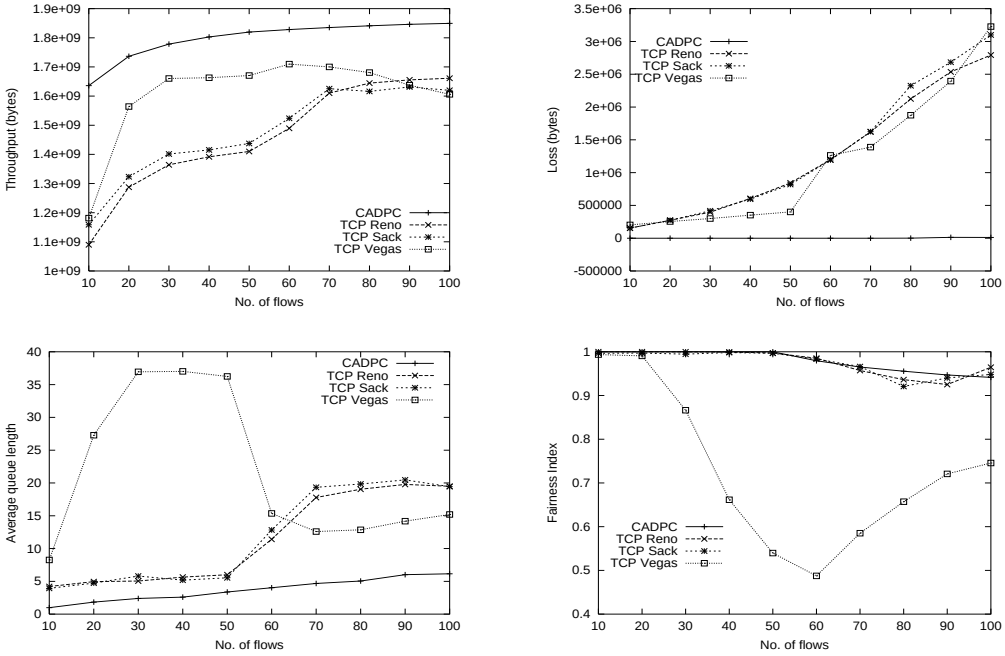

Abbildung 5. CADPC und TCP, Bottleneck 100 Mbit/s

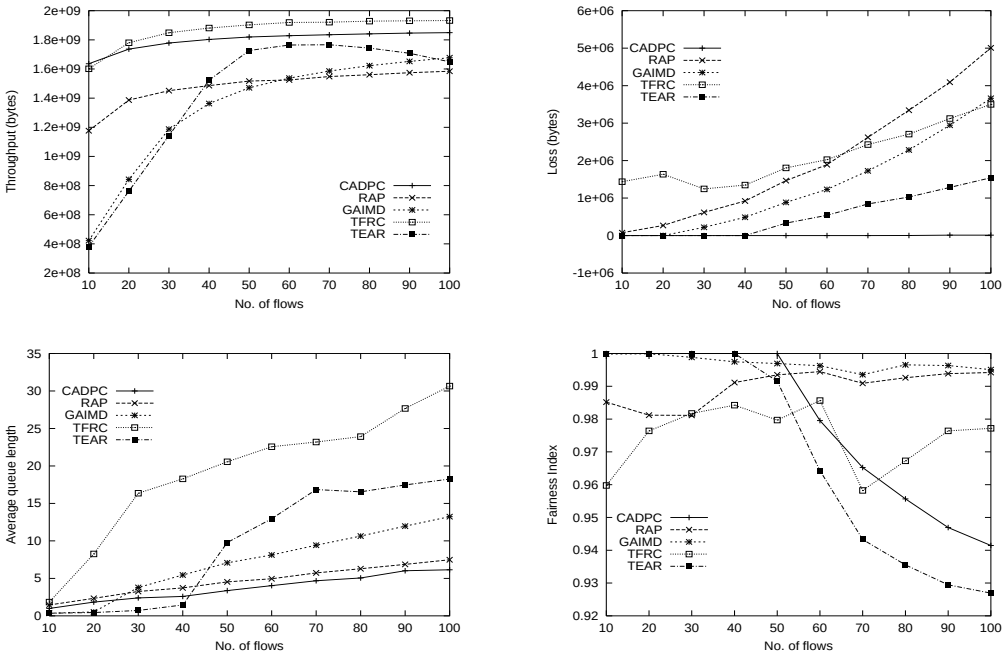

Abbildung 6. CADPC und gegenüber TCP faire Mechanismen, Bottleneck 100 Mbit/s

– QoS bei Verwendung der "Active Queue Management" Verfahren *RED*, *REM* und *AVQ*.

5 Zusammenfassung

Im Rahmen der Dissertation wurden:

– Das PTP Protokoll zur effizienten Abfrage von Performance-spezifischen Informationen von IP Routern
– Ein dazugehöriges offenes Framework für bessere Erweiterbarkeit und
– DAS CADPC Staukontrollverfahren, das die mit Hilfe des Protokolls abgefragte verfügbare Bandbreite sinnvoll einsetzt

entwickelt. PTP wurde für Linux (als Router und als Endsystem) und für den "ns-2" Netzwerksimulator implementiert. CADPC wurde im vereinfachten synchronen Fall mathematisch analysiert und in zahlreichen Simulationen im Vergleich mit mehreren TCP Varianten und gegenüber TCP fairen Mechanismen bewertet. Es zeigte sich, daß CADPC zumindest in einem speziellen Szenario ohne fremdartigen Hintergrundverkehr und bei PTP-Unterstützung durch mindestens jeden zweiten Router äußerst gute Resultate lieferte und TCP sowie einigen gegenüber TCP fairen Mechanismen in vieler Hinsicht (leichte Verfolgbarkeit der Rate, glätterer Bandbreitenverlauf, weniger Paketverlust, ...) überlegen ist. Um den Einsatz von PTP im globalen Internet zu rechtfertigen, ist als zukünftige Arbeit geplant, Möglichkeiten zur schrittweisen Einführung des Verfahrens zu untersuchen.