

Dynamisches Bandbreiten-Management

Jan Horbach, Uwe Hübner
Technische Universität Chemnitz

18. März 2009

Zusammenfassung

Im Chemnitzer Studentennetz (CSN) wird seit Oktober 2001 ein dynamisches Bandbreitenmanagement eingesetzt, um die verfügbaren Netzressourcen fairer aufzuteilen und einen kompetenten Umgang damit zu fördern. Nutzer, die bestimmte Transfermengen überschreiten, werden schrittweise in Nutzergruppen mit weniger verfügbarer Bandbreite eingeordnet. Dazu werden in einem Router (auf *Linux*-Basis) die Funktionalitäten hinsichtlich *Quality of Service* genutzt.

1 Einleitung

Die Universitäten, die am Gigabit-Wissenschaftsnetz (GWiN) des Deutschen Forschungsnetzes (DFN) angeschlossen sind, haben typischerweise Verträge über ein bestimmtes maximales Transfervolumen pro Monat. Überschreitet eine Einrichtung ihr Kontingent, muss sie mit Mahnungen und Bandbreitenbeschränkung rechnen.

Idealerweise sollten bedarfsgerechte Bandbreiten und Transfervolumina beschafft werden. Dies wird wegen beschränkter Budgets nicht immer möglich sein. Weiterhin gibt es Anwendungen (und Anwender!), die potenziell nahezu beliebig große Bandbreiten „verbrauchen“ können.

Wenn Ressourcen nicht unbegrenzt zur Verfügung stehen, muss ihre Benutzung limitiert werden. Oft sind bestimmte Bereiche oder Teilnetze für einen Großteil des Netzverkehrs verantwortlich, hier gibt es wirksame Steuerungsmöglichkeiten. Zum Beispiel kann man für jeden Nutzer die Datenmenge über einen bestimmten Zeitraum begrenzen. Nach einer Überschreitung dieser Grenze wird der Netzzugang gesperrt (ggf. erst nach einer Verwarnung bei erstmaligem Überschreiten). Diese Variante wurde im Chemnitzer Studentennetz (CSN) im Zeitraum von Dezember 2000 bis Oktober 2001 eingesetzt.

Das CSN ist eine Initiative von Studenten für Studenten mit dem Ziel, die privaten Rechner in den Studentenwohnheimen an das Campusnetz der TU Chemnitz anzuschließen (Abbildung 1). Derzeit gibt es etwa 1700 Nutzer, alle Wohnheime sind angeschlossen. Damit hat die TU Chemnitz eines der größten Studentennetze, die Anfänge reichen zurück bis ins Jahr 1994 [1].



Abbildung 1: Wohnheime im CSN (Foto: Karsten Petersen)

Ein wie oben beschriebenes „hartes“ Limit bringt allerdings auch Probleme mit sich. Größere Datenmengen für spezielle Lehr- und Forschungszwecke sind nur durch aufwendige Ausnahmeregelungen möglich. Weiterhin birgt ein Limit die Gefahr, dass kurz vor Ende des Abrechnungszeitraums von vielen Nutzern noch schnell das verbleibende Datenvolumen ausgeschöpft wird.

Der hier vorgestellte Ausweg ist ein intelligenteres Bandbreitenmanagement zur fairen Verteilung der Netzressourcen, bei der Nutzer je nach beanspruchtem Datenvolumen in verschiedene Nutzergruppen eingeordnet werden. Bei Mehrverbrauch steht ihnen weniger Bandbreite zur Verfügung. Das kann soweit gehen, dass es ihnen letztendlich nicht mehr möglich ist, ein bestimmtes Transfervolumen zu überschreiten [2].

2 Allgemeine Funktionsweise

Die Nutzer werden in Abhängigkeit ihres durchschnittlichen pro Tag verursachten Transfervolumens in Gruppen eingeteilt. Sollten sie bestimmte Grenzen überschreiten, werden sie einer niedriger bewerteten Gruppe mit weniger Bandbreite zugeteilt. Nach einer längeren Einhaltung der Grenzen gelangen die Nutzer schrittweise wieder in eine besser bewertete (d.h. mit mehr Bandbreite versorgte) Gruppe [3].

Nutzer, die nur selten eine größere Datenmenge beanspruchen, werden nur leicht eingeschränkt. Das geschieht durch eine Durchschnittsbildung bei der Zählung des Datenaufkommens, um einmalige Spitzen abzuschwächen. Auf diese Art und Weise stellen gelegentliche größere Transfers kein Problem dar, da keine Sperrung erfolgt und die „Behinderung“ durch die Beschränkung der Bandbreite in diesen Fällen gering ist.

Der Zugang zur Universität wird von der Begrenzung ausgenommen, damit Anreize für die Nutzer geschaffen werden, interne Ressourcen wie z.B. den FTP-Server des Rechenzentrums stärker zu nutzen. Die Bandbreite zu einem eventuell vorhandenen WWW-Proxy muss allerdings weiterhin begrenzt werden, da hier trotz der Zugehörigkeit zum internen Netz externes

Datenaufkommen verursacht wird. Der Caching-Gewinn wird allerdings durch einen Korrekturfaktor weitergeben.

Das System regelt sich weitgehend selbst, um bestimmte globale Zielparameter, wie z.B. eine Monatsobergrenze der Datenmenge, einzuhalten. Ein Vorteil dieser automatischen Regelung ist die optimale Ausnutzung der gegebenen Ressourcen. So stellen sich z.B. in Ferienzeiten, in denen weniger Personen das Netz nutzen, höhere Bandbreiten für die einzelnen Nutzergruppen ein. Ein Volumenkontingent kann so besser bedarfsgerecht aufgeteilt werden.

Notwendig hierbei ist eine schnelle Reaktion bei starken Anstiegen des Datenaufkommens, zum Beispiel am Semesterbeginn. Hierzu wird der Durchschnittswert über 7 Tage mit dem über 14 Tage verglichen; ein großer Unterschied stellt ein gutes Kriterium für ein erhöhtes Datenaufkommen dar (siehe Abbildung 2).

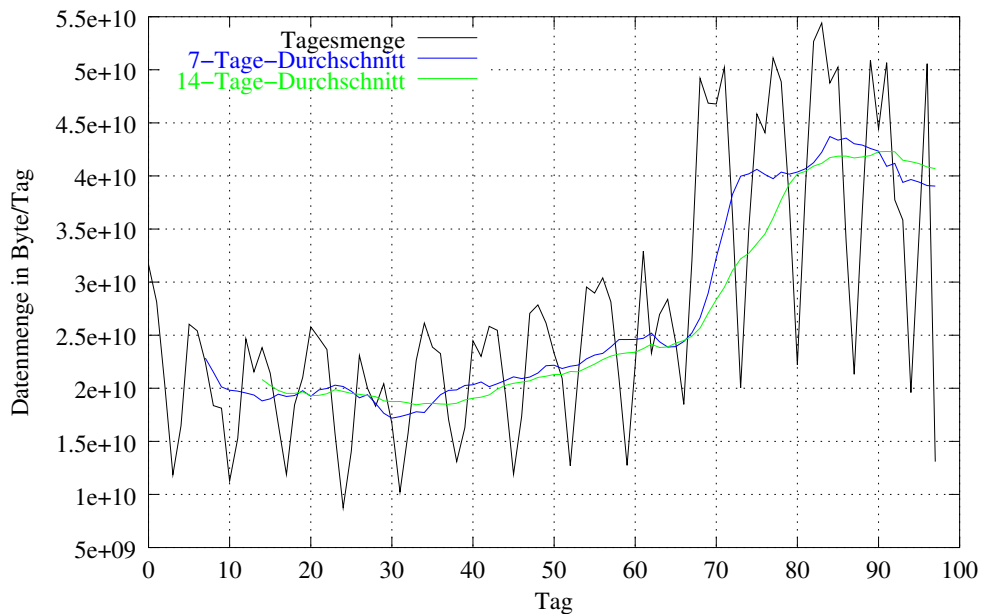


Abbildung 2: Datenaufkommen zum Semesterbeginn

Eine Durchschnittsbildung wird trotz der langsameren Reaktion verwendet, um kurzzeitige Einflüsse wie beispielsweise am Wochenende abzufangen.

Ein weiterer Vorteil des Systems ist der geringere Verwaltungsaufwand; die Nutzergruppen müssen nur einmal vor dem Einsatz eingerichtet werden, die Ausstellung von Verwarnungen und die Sperrung von Anschlüssen können entfallen.

3 Architektur des Systems

Für die internen Management-Schnittstellen findet XML-RPC Verwendung, ein RPC-Protokoll auf der Basis von XML und HTTP [4]. HTTP dient dabei als Übertragungsprotokoll, welches

die XML-kodierten Nutzdaten transportiert. Es ist ein sehr einfaches Protokoll, d.h. es ist einfach zu verstehen und einfach zu implementieren. Der große Vorteil ist seine Sprachunabhängigkeit, so dass Daten zwischen Applikationen in verschiedenen Programmiersprachen ausgetauscht werden können. Ebenso können auch komplexere Daten relativ komfortabel übergeben werden.

Die Management- und Informationsschnittstellen sind plattformunabhängig mit Hilfe von PHP realisiert, ebenso die Serverseite zum XML-RPC. Die XML-RPC-Daten werden per POST übergeben. Es ist also kein separater Server nötig, was sich günstig auf die Stabilität des Systems auswirkt, da es keine eigenen Prozesse gibt, die ständig laufen müssen und ausfallen könnten.

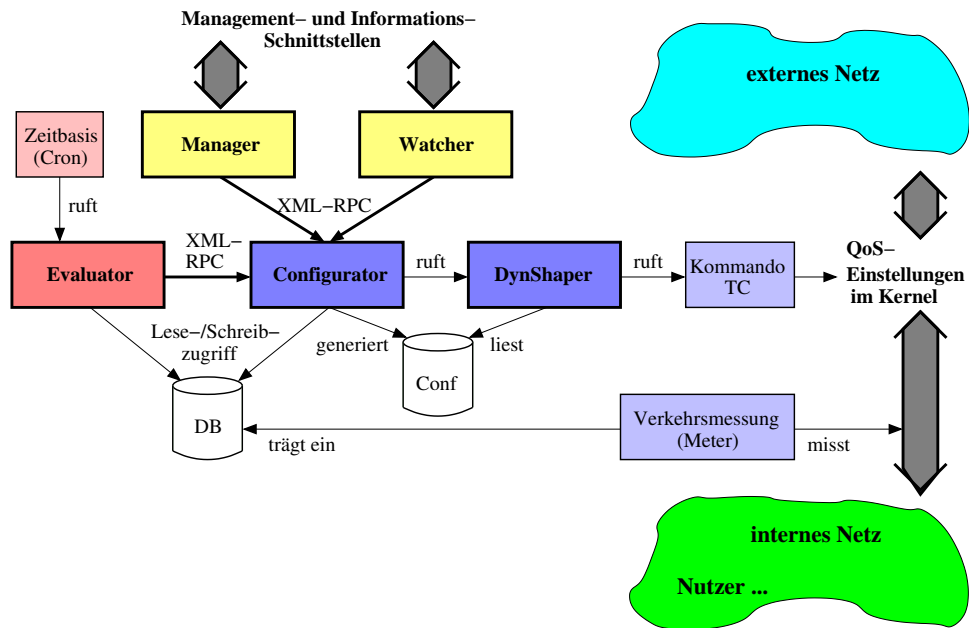


Abbildung 3: Architektur des Systems

Der *Manager* dient zur Verwaltung des *DynShapers*. Er legt die Parameter für die verwendeten Interfaces, Pfadnamen zu Programmen und eine Monatsobergrenze fest, stellt die Werte für die einzelnen Nutzergruppen ein und definiert Ausnahmeregelungen.

Mit dem *Watcher* ist es den einzelnen Nutzern möglich, ihre aktuellen Werte wie die verursachte Datenmenge, die Nutzergruppe, der sie momentan zugeordnet sind sowie deren Parameter abzufragen.

Durch den *Configurator* erfolgt die Konfiguration des *DynShapers*. Er verwaltet die Konfigurationsdaten und beantwortet XML-RPC-Anfragen, die Konfigurationsoptionen lesen oder schreiben wollen.

Der *Evaluator* wertet das Verkehrsaufkommen der Nutzer aus und ordnet die Nutzer bei Bedarf einer anderen Nutzergruppe zu. Dazu führt er einmal pro Stunde eine Abfrage der aktuellen Werte durch. Die Gruppe und das Datum der Einordnung werden in der Datenbank gespeichert. Weiterhin dient der *Evaluator* zur automatischen Regelung der Bandbreiten für die Gruppen, damit die festgelegte Monatsobergrenze nicht überschritten werden kann.

Für den *Evaluator* wird ein vom Webserver unabhängiger PHP-Interpreter benutzt, der mit XML- und Datenbankunterstützung kompiliert sein muss.

Der *DynShaper* dient zur eigentlichen Begrenzung des Verkehrs. Er wertet die vom *Configurator* erzeugte Konfiguration aus und legt *Queuing Disciplines*, Verkehrsklassen für die Nutzergruppen und Filterregeln fest.

Es handelt sich dabei um ein einfaches (Bash-)Shellskript, da hier nur wenige Konfigurationsoptionen eingelesen werden müssen, die zum Einrichten der Klassen und Filter notwendig sind.

Als Datenbanksystem wird im CSN schon seit einiger Zeit PostgreSQL eingesetzt und vom DynShaper-System mitverwendet.

4 Realisierung der Bandbreitenbeschränkungen

Zur Realisierung der dynamischen Bandbreitenbeschränkungen wird auf einem mit Linux betriebenen Router die dort vorhandene QoS-Funktionalität (*Quality of Service*) eingesetzt. Linux unterstützt eine Reihe von QoS-Verfahren: einfache Queuing-Algorithmen, die unterschiedliche Vergabe von Bandbreiten und Prioritäten, aber auch Algorithmen zur Überlastvermeidung von TCP-Flows und zur Verteilung des Datenverkehrs auf mehrere Netzinterfaces. Ein sehr leistungsfähiger Algorithmus ist das *Class-Based Queuing (CBQ)*. Es ermöglicht die Unterteilung des Verkehrs in verschiedene Klassen. [5, 6]

Bei der einfachsten Strategie FIFO werden zu versendende Pakete in der Reihenfolge ihres Eingangs in eine Warteschlange eingeordnet, über die sie dann in derselben Reihenfolge gesendet werden.

Beim *Priority Queuing* werden höher priorisierte Pakete vor niedriger priorisierten in die Warteschlange eingeordnet. Dazu muss jedes Paket analysiert und die Warteschlange gegebenenfalls umgeordnet werden, was unvorteilhaft für den Durchsatz ist.

Das *Class-Based Queuing (CBQ)* ermöglicht demgegenüber eine effizientere Verarbeitung. Hier existieren mehrere Warteschlangen, auf die die Bandbreite hierarchisch verteilt wird.

Die Klassen bilden eine Baumstruktur, in der sich die *Kinder* von den *Eltern* bei Bedarf Bandbreite borgen können, wenn dort noch Kapazität vorhanden ist. Im Gegenzug können die *Kinder* nicht benötigte Bandbreite den *Eltern* zur Verfügung stellen, die dann gegebenenfalls an andere *Kinder* verborgt wird. Den Blättern im Baum können unterschiedliche *Queuing Disciplines* zugewiesen werden, so dass verschiedene Service-Klassen durch geeignete Algorithmen behandelt werden können (Abbildung 4).

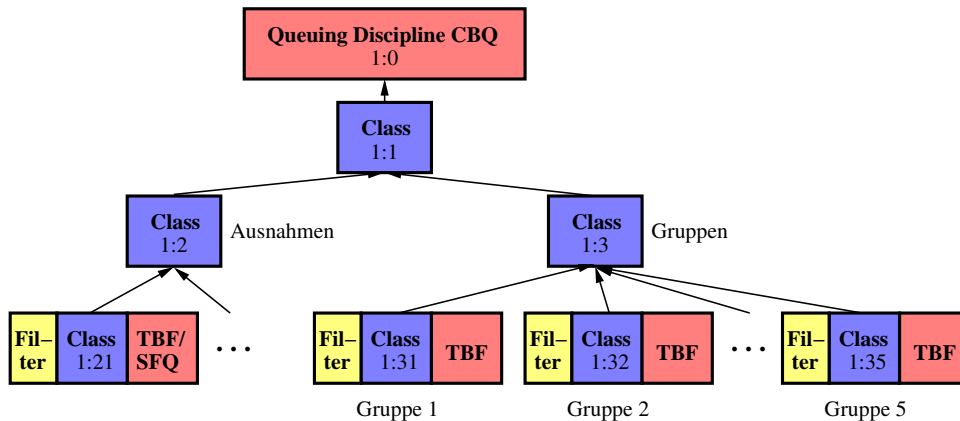


Abbildung 4: Klassenbaum im Chemnitzer Studentennetz

Die Klassifikation erfolgt über Filter, welche IP-Adressen oder andere Merkmale im Paket auswerten. Nach der Klassifizierung folgt die eigentliche Verkehrsbeeinflussung. Bei uns wird für jeden Nutzer ein Filter angelegt, der ihn seiner entsprechenden Gruppe bzw. Verkehrsklasse zuweist, so dass pro Klasse mehrere Filter existieren.

5 Fazit

Bisher wurde QoS häufig als Instrument zur Gewährleistung einer bestimmten Bandbreite (z.B. für Videodatenströme) angesehen. Wir sehen hier eine neue Anwendungsdomäne zur Sicherung der Einhaltung bestimmter Zielparameter (z.B. Datenvolumen). Dies erfolgt durch eine möglichst „intelligente“ Bandbreitenbegrenzung für einzelne Nutzergruppen.

Das System befindet sich seit Oktober 2001 im Chemnitzer Studentennetz (CSN) im praktischen Einsatz. Während der Testphase gab es sehr angeregte Diskussionen um Grundsätzliches und Details dieser Lösung. Einige Erfahrungen sind in Änderungen von Betriebsparametern eingeflossen.

Insgesamt schätzen wir ein, dass dieser Ansatz einer „intelligenten“ Ressourcensteuerung gerade in Zeiten knapper Budgets eine überlegte und sensible Netznutzung fördert.

Literatur

- [1] „[CSN] Informationen - Allgemein“, Chemnitzer Studentennetz, 2000
https://www.csn.tu-chemnitz.de/info/allg_info.html
- [2] Jan Horbach: „Diplomarbeit - Dynamische Bandbreitenbeschränkung mit QoS“, TU Chemnitz, Professur Rechnernetze und verteilte Systeme, 2001
<http://archiv.tu-chemnitz.de/pub/2001/0100/>
- [3] „[CSN] Informationen - Beschränkungen“, Chemnitzer Studentennetz, 2001
<https://www.csn.tu-chemnitz.de/info/limits.html>

- [4] UserLand: „XML-RPC Home Page“, UserLand Software, Inc., 2001
<http://www.xmlrpc.com>
- [5] Saravanan Radhakrishnan: „Linux - Advanced Networking Overview“, 1999
<http://qos.ittc.ukans.edu/howto/>
- [6] Mark Lamb: „iproute2+tc notes“, 1999
<http://snafu.freedom.org/linux2.2/iproute-notes.html>