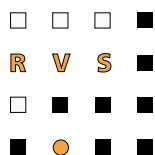




# Konzeption einer Cache-Server-Infrastruktur auf dem Wissenschaftsnetz

## Abschlußbericht

Version 0.1, August 1998, C. Grimm, J.-S. Vöckler



Lehrgebiet Rechnernetze und Verteilte Systeme (RVS)

Prof. Dr.-Ing. Helmut Pralle

Universität Hannover

### Inhalt

1.	Bezeichnung des Projektes	2
2.	Beschreibung des Projektes	2
	2.1 Motivation	2
	2.2 Projektziel	2
3.	Durchführung des Projektes	3
	3.1 Personal	3
	3.2 Investitionen	3
	3.3 Vorträge, Veröffentlichungen	4
	3.4 Weitere Aktivitäten	4
4.	Verlauf des Projektes	6
	4.1 Installation und Betrieb der 10 zentralen Server	6
	4.1.1 Hardware	7
	4.1.2 Betriebssystem	7
	4.2 Cache-Software	8
	4.3 Überwachung der Server	8
	4.3.1 Sammlung und Visualisierung der Betriebsparameter	8
	4.3.2 Automatische Überwachung der Cache-Prozesse	12
	4.3.3 Verkehrsstatistiken	12
	4.4 Nutzung der DFN-Caches	13
	4.5 Konzeption DFN-Cache-Verbund	14
	4.5.1 Ursprüngliche Konzeption	15
	4.5.2 Verbesserte Konzeption	17
	4.5.3 Weiterführende Überlegungen	18
	4.6 Weitere Arbeiten	18
	4.6.1 Teilnahme am internationalen Cache-Verbund com-Mesh	18
	4.6.2 Entwicklung von webtee	19
5.	Ergebnisse des Projektes	21
	5.1 Nutzen des DFN-Cache-Verbundes	21
	5.1.1 Bewertung	22
	5.2 Entwicklung WWW-Verkehr	23
	5.3 Fortführung des Projektes	23
A.	Anhang	25
	A.1 Verkehrsstatistiken	25
	A.2 Beispiele für die Nutzung von webtee	41

# 1 Bezeichnung des Projektes

Projekttitel	Konzeption einer Cache-Server-Infrastruktur auf dem Wissenschaftsnetz
Kurztitel	DFN-Cache-Service
Ansprechpartner	Christian Grimm Jens-Sönke Vöckler Lehrgebiet Rechnernetze und Verteilte Systeme (RVS), Universität Hannover (Leiter: Prof. Dr.-Ing. Helmut Pralle)
Ausführende Einrichtung	RVS, Universität Hannover
Projektlaufzeit	18 Monate (1.8.1996 - 31.1.1998)
Verlängerung	2 Monate (1.2.1998 - 31.3.1998)

## 2 Beschreibung des Projektes

### 2.1 Motivation

Das Projekt *Konzeption einer Cache-Server-Infrastruktur auf dem Wissenschaftsnetz* wurde am 1. August 1996 gestartet. Bereits zu dieser Zeit war das starke Anwachsen des World-Wide Web-Verkehrs im Internet zu erkennen. Ursache dafür war die zunehmende Anzahl an WWW-Servern und der deutliche Zuwachs an Nutzern, die die Informationsangebote von den WWW-Servern abrufen konnten. Zudem zeigte sich, daß verstärkt Multimedia-Daten in das World-Wide Web eingebunden wurden. Es war abzusehen, daß bei einer unkoordinierten Weiterentwicklung dieses Szenarios auch im Wissenschaftsnetz deutliche Engpässe bei der Kommunikation auftreten würden, sowohl auf den internen als auch im besonderen auf den externen Leitungen. Auch eine kontinuierliche Vergrößerung der im Wissenschaftsnetz zur Verfügung gestellten Bandbreite konnte keine dauerhafte Lösung darstellen.

Zeitgleich gab es die ersten Entwicklungen im Bereich der Cache-Verbünde. Ausgelöst durch das Harvest-Projekt der University of South Colorado, Boulder, erschien der *Harvest Object Cache*. Mit dieser Software wurde erstmals ein Protokoll für die effiziente Kommunikation von Caches untereinander realisiert, das sogenannte *Internet Cache Protocol (ICP)*. Somit wurde es möglich, auch in Weitverkehrsnetzen Verbünde von Caches zu bilden. Der erste Cache-Verbund im Wissenschaftsnetz wurde Anfang 1996 zwischen den Universitäten Bochum, Bonn und Frankfurt gebildet. Kurze Zeit später schloß sich eine Vielzahl weiterer Caches an diesen Verbund an, so daß bald ein unkoordiniertes und unstrukturiertes Netz von Caches im Wissenschaftsnetz vorhanden war.

Aus dieser Situation heraus wurde das Projekt *Konzeption einer Cache-Server-Infrastruktur im Wissenschaftsnetz* gestartet. Die Ziele des Projektes werden im folgenden erläutert.

### 2.2 Projektziel

Das Ziel des Projektes bestand zunächst in der Klärung der Grundlagen für einen übergreifenden Cache-Service im Wissenschaftsnetz. Die oben dargelegte Situation machte es jedoch erforderlich, daß bereits im Rahmen dieses Projektes eine Cache-Infrastruktur im Backbone

des Wissenschaftsnetzes installiert und in Betrieb genommen wurde. Vereinfacht wurde diese Erweiterung des ursprünglichen Projektzieles durch die Tatsache, daß zeitgleich mit dem Projektbeginn aus damaliger Sicht leistungsfähige Server an den 10 zentralen Knoten im B-WiN installiert wurden. Diese Server konnten im Rahmen des Projektes als zentrale DFN-Caches eingesetzt werden.

Neben dem eigentlichen Betrieb des DFN-Cache-Verbundes sollte auch die weitere Bildung lokaler Cache-Verbünde beobachtet und gegebenenfalls koordiniert werden. Dabei stand insbesondere die Empfehlung von Konzepten zur Interaktion lokaler Cache-Verbünde mit dem DFN-Cache-Verbund im Vordergrund. Ziel war dabei die Verbesserung des lose verteilten Verbundes lokaler Caches über das gesamte Wissenschaftsnetz.

Neben den betrieblichen Aspekten des DFN-Cache-Verbundes mußte auch die Entwicklung der Caching-Software verfolgt und beurteilt werden. Ziel war es hier, auf der Basis geprüfter Cache-Software mit den DFN-Caches eine möglichst stabile und performante Cache-Infrastruktur zu schaffen. Anhand eigener Untersuchungen sollten auch Empfehlungen für Cache-Konfigurationen ausgesprochen werden.

Ein weiteres Ziel des Projektes war die allgemeine Aufarbeitung theoretischer Hintergründe des Caching. Hierbei stand die Gewährleistung der Aktualität und Integrität von Dokumenten sowie die Datensicherheit innerhalb eines Cache-Verbundes im Vordergrund.

Der Verlauf und die Ergebnisse des Projektes werden mit diesem Abschlußbericht vorgelegt.

## **3 Durchführung des Projektes**

### **3.1 Personal**

An dem Projekt waren folgende Personen beteiligt:

- Projektleitung
  - Prof. Dr.-Ing. Helmut Pralle
- Wissenschaftliche Mitarbeiter
  - Dipl.-Ing. Christian Grimm
  - Dipl.-Ing. Jens-Sönke Vöckler
- Wissenschaftliche Hilfskräfte
  - Jörn Jachalsky
  - Marc Neitzner
  - Yibing Zhao

### **3.2 Investitionen**

Zur Durchführung des Projektes wurden folgende Investitionen getätigt:

### Hardware aus Mitteln des DFN-Vereins

- 10 Cache-Server für den Produktionseinsatz, jeweils ausgestattet mit:
  - Sun Ultra2 Enterprise
  - zwei 167 MHz Prozessoren
  - 256 MByte Hauptspeicher
  - Plattenkapazität 20 GByte

### Hardware aus Projektmitteln

- 2 Cache-Server für lokales Caching-Testbed sowie zur Aufbereitung von Cache-Statistiken, baugleich zu oben. Abweichungen:
  - Plattenkapazität 2 x 4 GByte
  - 155 Mbps ATM Interfaces

## 3.3 Vorträge, Veröffentlichungen

Teilergebnisse des Projektes wurden wie folgt präsentiert:

- Vorträge über das Projekt
  - Grimm, C.: DFN-Arbeitskreis Informationsdienste, Hannover, Oktober 1996
  - Grimm, C.: 25. DFN Betriebstagung, Berlin, Oktober 1996
  - Grimm, C.: 1st Workshop on Caching, Warschau, Oktober 1996
  - Grimm, C.: 1. Treffen Cache-Betreiber BelWue, Ulm, April 1997
  - Grimm, C.: 2nd Workshop on Caching, Boulder, Juni 1997
  - Grimm, C.: *"Die Cache-Struktur des DFN im B-WiN - Konzept, Aufbau und Betriebserfahrungen"*, Arbeitskreis Systemadministration heterogener verteilter Rechnersysteme, Universität Mainz, März 1998.
- Allgemeine Vorträge zum Thema Caching
  - Grimm, C.: *"Caching im World-Wide Web"*, Seminar Multimedia im Internet, Universität Hannover, Februar 1997.
- Veröffentlichungen
  - Grimm, C.: *"DFN-Caches im B-WiN"*, DFN Mitteilungen 43, März 1997.
- Projekteigener WWW-Server: <http://www-cache.dfn.de/>

## 3.4 Weitere Aktivitäten

- Ständige aktive Teilnahme an der TERENA Caching Task Force (TF-Cache):
  - September 1996, Amsterdam (G. Foest),
  - Januar 1997, Amsterdam (C. Grimm),
  - Mai 1997, Edinburgh (C. Grimm),
  - September 1997, Amsterdam (C. Grimm, J.-S. Vöckler),
  - Januar 1998, Amsterdam (C. Grimm, J.-S. Vöckler).
- *Peer Reviews* von Berichten anderer Projekte:
  - Bekker, H., de Jong, A., Melve, I., Verschuren, T.: *"Report on the Costs and Benefits of Operating Caching Services"*, DESIRE Project Deliverable, September

1997,

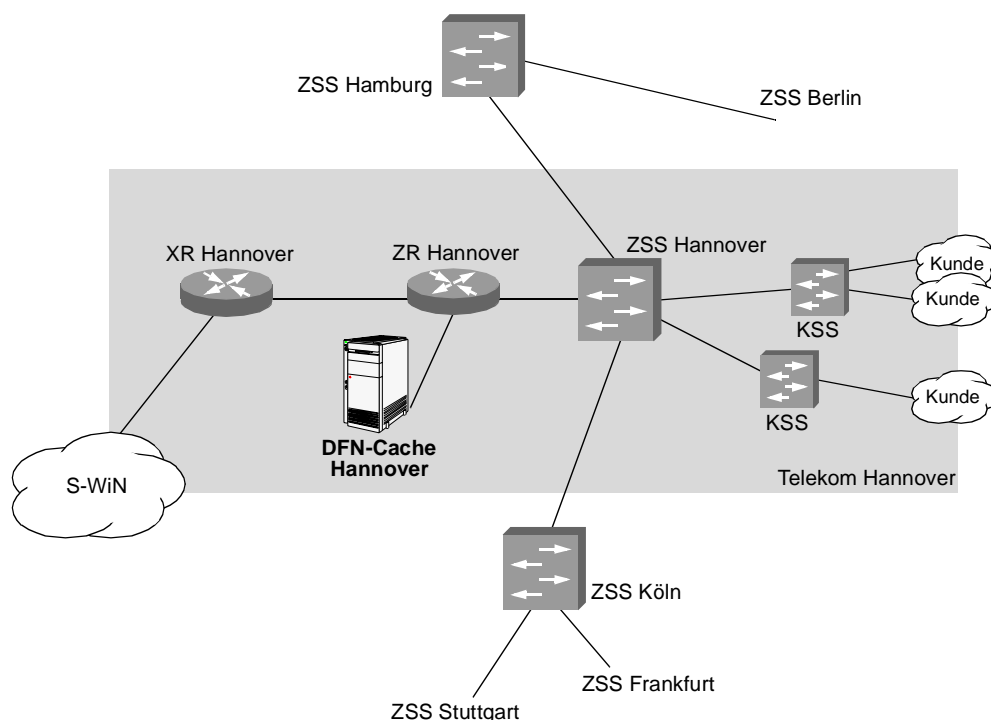
Reviewers: C. Grimm, J.-S. Vöckler

- Bekker, H, Melve, I., Slettjord, L, Verschuren, T.: *"Practical Experiences of Establishing Web Proxy Caching Meshes"*, DESIRE Project Deliverable, März 1998, Principal Reviewer: J.-S. Vöckler.

## 4 Verlauf des Projektes

### 4.1 Installation und Betrieb der 10 zentralen Server

Die Installation und Aufstellung der 10 zentralen Server zog sich von September bis Dezember 1996 hin. Abbildung 1 zeigt den prinzipiellen Aufbau eines zentralen Knotens sowie den Anschluß eines Servers an einen zentralen Router (ZR) im B-WiN. Stellvertretend für alle 10 annähernd identisch aufgebauten Knoten ist hier als Beispiel der Standort Hannover gewählt. Der Aufstellungsort in unmittelbarer Nähe zu den ZRs bedingte, daß die Server in Gebäuden der Deutschen Telekom installiert wurden.



*Bild 1: Integration DFN-Caches in B-WiN - Beispiel Standort Hannover*

Die 10 Server wurden während des Cache-Projektes im wesentlichen für drei Aufgaben genutzt:

1. Sammlung von Accounting-Informationen der zentralen Router (NOC, Stuttgart),
2. Verteilung von News<sup>1</sup> (DFN-GS, Berlin) und
3. Caching (RVS, Universität Hannover).

Die Administration der Server wurde von der DFN-Geschäftsstelle vorgenommen. Mitarbeitern des Cache-Projektes sowie des NOC wurden keine Rechte zur Administration eingeräumt (sog. root-Privilegien). Hierdurch kam es lediglich in der Anfangsphase zu Schwierigkeiten bei der Installation von speziellen Software-Tools, die für die Durchführung des Projektes notwendig waren. Im späteren Verlauf zeigte sich allerdings, daß diese Maß-

1. <http://www.dfn.de/service/newskonzept.html> (April 1997)

nahme für eine gemeinsame und dennoch störungsfreie Nutzung der Server unumgänglich war.

#### 4.1.1 Hardware

Die über den gesamten Projektverlauf unveränderte Ausstattung der Cache-Server war wie folgt:

- Sun UltraSparc,
- zwei 167 MHz Prozessoren,
- 256 MByte Hauptspeicher,
- 20 GByte Plattenkapazität sowie
- 10 Mbps Ethernet und 100 Mbps Fast Ethernet Interface.

Da die ZRs lediglich mit 10 Mbps Ethernet Interfaces ausgestattet waren, konnten die Server nicht über die 100 Mbps Fast Ethernet Interfaces betrieben werden. Neben Server und Router waren keine weiteren Maschinen an das Ethernet angeschlossen

Für den Betrieb als Cache-Server wurde auf jeder der 10 Maschinen eine Cache-Partition von 6 GByte eingerichtet. Diese Partition diente der ausschließlichen Speicherung von WWW-Objekten. Die Logfiles der Cache-Prozesse wurden auf einer anderen Partition abgelegt. Insgesamt standen somit 60 GByte Plattenplatz für den DFN-Cache-Service zur Verfügung.

#### Ausfälle

Auf allen Servern traten im Verlauf des Projektes Störungen durch defekte Festplatten auf. Konnten in vielen Fällen die Probleme durch Reparatur der Dateisysteme (sog. filesystem check) behoben werden, mußten eine Reihe von Platten ausgetauscht werden. Die Dauer der Ausfälle betrug zwischen einigen Stunden für eine Überprüfung der Dateisysteme bis zu mehreren Tagen für den Austausch einer defekten Platte. Eine deutliche Häufung der Ausfälle von Festplatten war Mitte 1997 festzustellen. Dies Ausfälle lassen sich an den Unregelmäßigkeiten in den Betriebsstatistiken erkennen (s. Anhang A.1). Abgesehen von den durch defekte Festplatten verursachten Störungen traten keine weiteren Hardware-Ausfälle auf.

#### 4.1.2 Betriebssystem

Auf den 10 Servern wurde das Betriebssystem Solaris 2.5.1 eingesetzt. In unregelmäßigen Abständen wurden zudem Betriebssystem-Patches installiert.

#### Probleme

Das Betriebssystem Solaris 2.5.1 wurde vornehmlich für den Einsatz in lokalen Netzen konzipiert. Die meisten Kernel-Parameter, insbesondere des TCP/IP-Stack, weisen Default-Einstellungen auf, die die Verwendung von Solaris auf einem Server im WAN beeinträchtigen. Da die DFN-Caches im Prinzip ausschließlich im WAN-Verkehr eingesetzt wurden, ergab sich eine erkennbare Einschränkung der Performance allein durch das Betriebssystem.

Die Default-Einstellungen können zwar durch Umkonfigurieren des Kernel verändert werden, Warnungen über Nebeneffekte und fehlende längerfristige Erfahrungen ließen von der Umsetzung auf den DFN-Caches weitgehend absehen. Zudem waren Änderung nur in Absprache mit den Betreibern der anderen Dienste möglich. So konnten lediglich einige geringfügige Änderungen der Parameter vorgenommen werden. Wie erwartet, konnten dadurch keine wesentlichen Verbesserungen der Performance bewirkt werden.

## 4.2 Cache-Software

Als Software für die Cache-Prozesse wurde ausschließlich *Squid* eingesetzt. Squid wird im Rahmen eines Caching-Projektes am *National Laboratory for Applied Network Research* (NLANR) entwickelt und ist frei verfügbar<sup>1</sup>.

Im Verlauf des Projektes wurden 24 verschiedene Versionen von Squid freigegeben, 7 davon wurden auf den DFN-Caches eingesetzt. Die Vielfalt der Versionen liegt zum einen darin begründet, daß die Quellen von Squid offenliegen und dadurch viele Entwickler oder Betreiber von Caches schnell und unkompliziert Bugfixes oder Verbesserungen beitragen können. Zum anderen erfordert die Komplexität und Dynamik des WWW bzw. des dem WWW zugrundeliegenden Protokolls HTTP ständig Anpassungen der Cache-Software. So waren in fast jeder neuen Version von Squid Features enthalten, mit denen die Performance, Stabilität oder Funktionsvielfalt erhöht werden konnte. Folglich war mit jeder Installation neuer Squid-Versionen auf den DFN-Caches eine Behebung von Fehlern oder eine Verbesserung des Service verbunden.

## 4.3 Überwachung der Server

Die Überwachung der Server wurde zu Beginn des Projektes von den Administratoren in der DFN-GS vorgenommen. Es zeigte sich jedoch bald, daß für den störungsfreien Betrieb der Cache-Prozesse eine Vielzahl spezieller Systemparameter beobachtet werden mußten. Daher wurde das Monitoring der Server weitgehend an das DFN-Cache-Projekt übergeben. Mitte 1997 wurde ein Konzept für ein umfassendes Monitoring der Cache-Server entwickelt und umgesetzt.

Die im Rahmen des Monitoring gesammelten Daten beinhalten im wesentlichen alle wichtigen mit dem Cache-Service zusammenhängenden Informationen über

- den Squid-Prozeß,
- das Betriebssystem und
- das Netzinterface.

Aus den drei genannten Parameterblöcken ergibt sich ein umfassendes Bild über das momentane Verhalten bzw. den Zustand der Caches. Durch die geeignete Auswahl von Parametern und Festlegung von Grenzwerten wurde zusätzlich eine automatische Überwachung möglich, anhand derer ein Fehlverhalten frühzeitig erkannt werden konnte. Die Visualisierung der gesammelten Daten erfolgte mit Hilfe des *Multi Router Traffic Grapher* (MRTG). MRTG ist ein Perl-basiertes Tool, welches ursprünglich für die Auswertung von Router-Statistiken entwickelt wurde. Durch geeignete Modifizierungen und Erweiterungen ist es jedoch auch in der Lage, Daten anderer Quellen darzustellen.

### 4.3.1 Sammlung und Visualisierung von Betriebsparametern

Von einem zentralen Rechner am RVS ([statistics.www-cache.dfn.de](http://statistics.www-cache.dfn.de)) erfolgt die Sammlung und Visualisierung der Betriebsparameter. In 5-minütigen Intervallen werden von diesem

---

1. <http://squid.nlanr.net/> (August 1998)

Rechner die relevanten Daten aller 10 DFN-Caches abgerufen, in einer Datenbank abgelegt und anschließend ausgewertet.

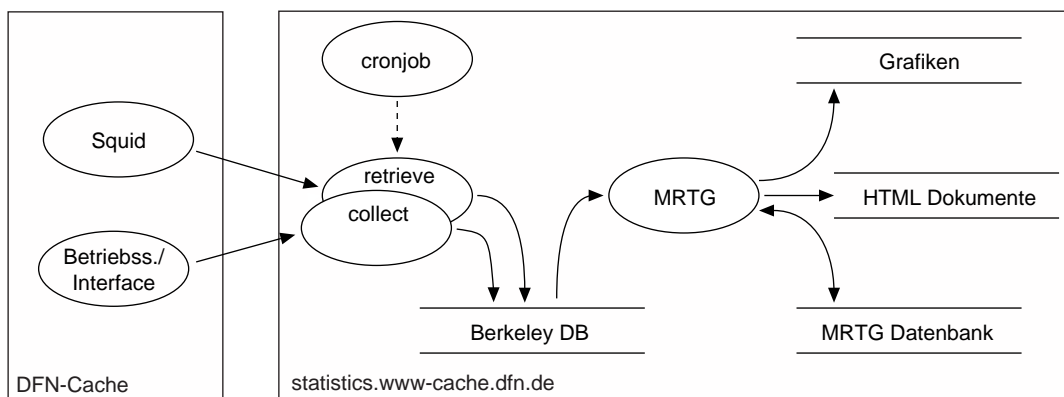


Bild 2: Datenflußdiagramm zur Sammlung und Visualisierung der Betriebsparameter

Abbildung 2 zeigt das vollständige Datenflußdiagramm. Zwei per cronjob gestartete Skripte (retrieve bzw. collect) rufen die entsprechenden Informationen der DFN-Caches ab (Squid-Prozeß bzw. Betriebssystem und Netzinterface) und legen die Daten in einer Datenbank ab. Tabelle 1 zeigt eine Übersicht über die gesammelten Parameter.

Squid-Prozeß	Betriebssystem/Netzinterface
DNS Abfragen	Kollisionen und Eingabefehler
Filedeskriptoren	Ein- und Ausgabeframes
HTTP und ICP Datenraten	Workload des Systems
HTTP und ICP Anfragen	TCP <i>active</i> und <i>passive opens</i>
Anzahl Objekte im Cache	TCP fehlgeschlagene Verbindungen
Antwortzeit HTTP	TCP Datenrate
Antwortzeit ICMP	TCP Fehlerdatenrate
Antwortzeit TCP	TCP Verbindungen/CLOSE_WAIT
	TCP Segmentrate
	UDP Datagrammrate

Tabelle 1: Überwachte Parameter der Caches.

Für jeden Parameter von jedem DFN-Cache wird eine eigene Datenbank angelegt. Als Datenbank wurde die auf nahezu allen Betriebssystemen vorhandene *Berkeley DB* eingesetzt. Neben der hohen Verfügbarkeit zeichnet sich die Berkeley DB durch eine mächtige Perl-Schnittstelle aus, die für die weitere Verarbeitung der Daten mit MRTG notwendig war.

In der zweiten Stufe der Verarbeitung entnimmt MRTG der Berkeley DB die für die grafische Darstellung relevanten Informationen. Dabei werden die Daten konvertiert und in einer zweiten Datenbank abgelegt (MRTG-Datenbank). Aus dieser eigenen Datenbank generiert MRTG die Grafiken und HTML-Dokumente, die für die WWW-basierte Visualisierung notwendig sind. Für jeden Parameter von jedem DFN-Cache werden jeweils vier zeitliche Verläufe protokolliert. Ausgehend von dem jeweils aktuellen Zeitpunkt sind dies die Werte der vergangenen

- zwei Tage (im 5 Minuten Mittel)
- zwei Wochen (im 30 Minuten Mittel)
- zwei Monate (im 2 Stunden Mittel)
- zwei Jahre (im Tagesmittel).

Abbildung 3 zeigt als Beispiel das Ergebnis der mit MRTG generierten grafischen Ausgabe. Zu sehen ist hier die Darstellung der Filedeskriptoren, die von dem Squid-Prozeß auf dem DFN-Cache in Hannover genutzt werden.

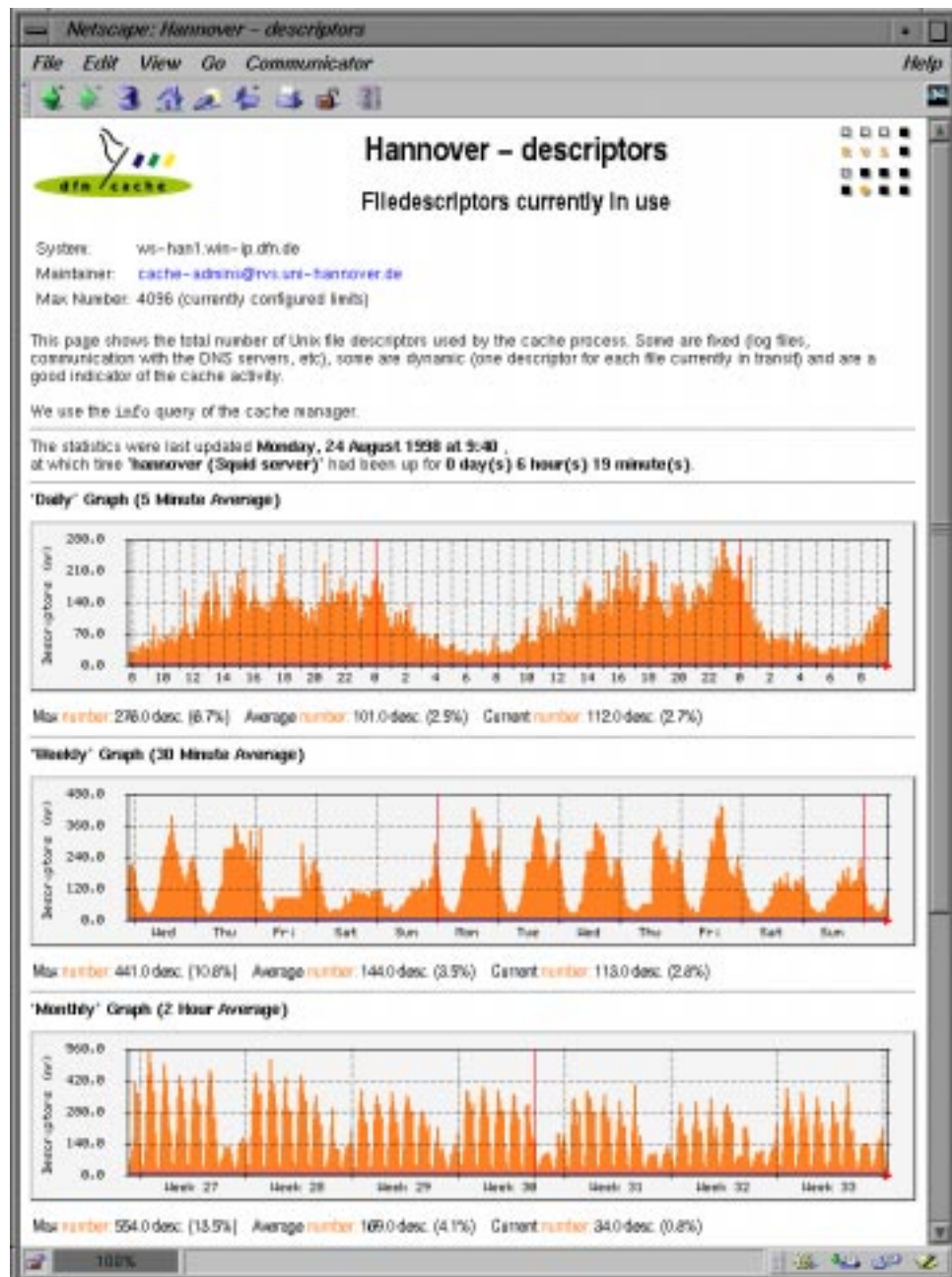


Bild 3: Ausschnitt einer mit MRTG generierten HTML-Seite mit Grafiken.

Um eine einfache und schnelle Übersicht über den momentanen Systemstatus zu erhalten, wurden weitere Ansichten der Grafiken erzeugt. So wurde in der gleichzeitigen Darstellung der Tagesverläufe aller Systemparameter eines DFN-Caches eine Möglichkeit gefunden, die Abhängigkeiten der Systemparameter untereinander besser beurteilen zu können. So konnte auf einfache Weise dargestellt werden, daß ein deutlicher Anstieg der HTTP-Requests an einen DFN-Cache in der Regel mit einem Anstieg der Kollisionen auf dem Ethernet sowie einem Abfallen der HTTP-Datenrate einherging.

Eine dritte Variante der grafischen Ausgabe lag in der gleichzeitigen Darstellung eines Systemparameters aller DFN-Cache in einer Grafik. Hiermit konnte ein einfacher Vergleich der DFN-Caches untereinander vorgenommen werden. Da MRTG nicht in der Lage ist, mehr als zwei zeitliche Verläufe in einem Diagramm gleichzeitig darzustellen, wurde für die Generierung der Grafiken das Tool *drawlog* eingesetzt. Abbildung 4 zeigt als Beispiel die Ausgabe der genutzten Filedeskriptoren aller 10 DFN-Caches.

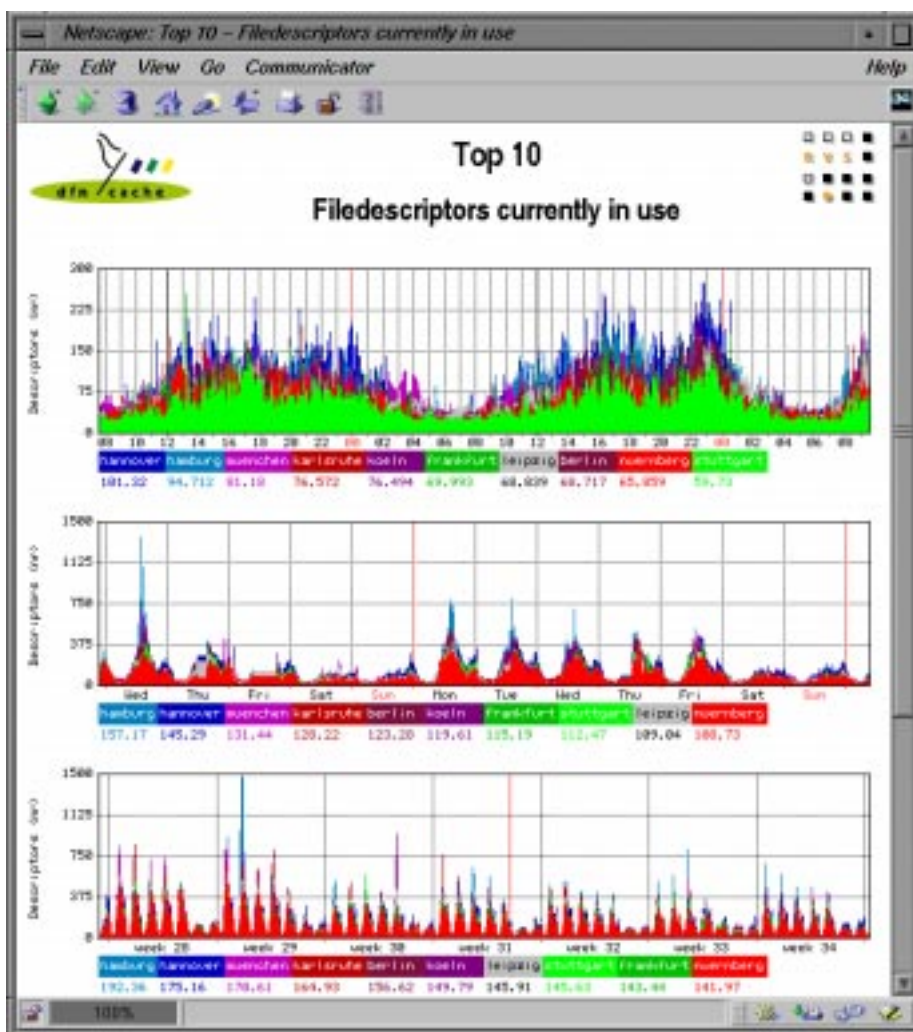


Bild 4: Gleichzeitige Darstellung von Systemparametern aller 10 DFN-Caches.

Mit den entwickelten Verfahren und Programmen stand am Ende des Projektes ein Satz von Werkzeugen zur Verfügung, der die gleichzeitige Überwachung von 10 verteilten Cache-Servern deutlich vereinfachte. Aufgrund des hohen Interesses anderer Cache-Administratoren

wurden die Werkzeuge mit ausführlichen Anleitungen zum Betrieb und Installation auf dem projekteigenen WWW-Server abgelegt.

### 4.3.2 Automatische Überwachung der Cache-Prozesse

Durch längere Beobachtung konnten aus den gesammelten Betriebsparametern diejenigen isoliert werden, die zuverlässige Hinweise auf den Zustand eines Caches erlaubten. In Verbindung mit zugeordneten Schwellwerten ließen sich Prozesse zur automatischen Überwachung bzw. Alarmierung (sog. Watchdogs) installieren. Die überwachten Parameter waren:

- Anzahl genutzter Filedesktoren,
- Verhältnis HTTP- zu ICP-Transferrate und
- HTTP-Antwortzeit (gemessen durch Abfrage eines HTML-Dokumentes).

Bei Überschreitung der Schwellwerte wurde durch den Watchdog eine EMail an die Administratoren der Caches gesendet. Die Schwellwerte wurden im einzelnen wie folgt festgelegt:

- War die Anzahl genutzter Filedesktoren oder die Wartezeit für ein HTML-Dokument aus dem Cache gleich 0, waren entsprechende Abfragen des Squid-Prozesses fehlgeschlagen. In diesen Fällen mußte von einer grundlegenden Betriebsstörung des betroffenen DFN-Caches ausgegangen werden.
- Betrug die Anzahl der genutzten Filedesktoren mehr als 1600, konnte der Squid-Prozeß eine Reihe von HTTP-Verbindungen nicht ordnungsgemäß ab- oder aufbauen. Diese Situation trat in der Regel dann ein, wenn Probleme im Routing innerhalb oder auch außerhalb des B-WiN auftraten.
- Überstieg die Wartezeit auf ein Testdokument aus einem DFN-Cache die Dauer von 5 Sekunden, trat aus Sicht den Nutzer eine nicht tolerierbare Betriebssituation ein. Dieser Wert wies in der Regel auf den Ausfall oder die totale Überlastung eines DFN-Caches hin.
- Da HTTP- und ICP-Transferrate über demselben Zeitabschnitt ermittelt wurden, ergab der Quotient aus beiden Werten das Verhältnis der übertragenen Datenmenge von HTTP- und ICP-Verkehr. Eine Alarmierung erfolgte dann, wenn der Quotient einen Wert von 1.08 unterschritt. Dieser Wert mußte in die Überwachung aufgenommen werden, da es in mehreren Fällen zu besonders schwerwiegenden Fehlfunktionen von DFN-Caches kam. Die Fehlfunktion äußerte sich in der ordnungsgemäßen Beantwortung von ICP-Anfragen, nicht jedoch aber von HTTP-Anfragen. Dieses Verhalten ist insbesondere in Cache-Verbänden nicht tragbar, da ein vermeintlich funktionierender Cache (Beantwortung von ICP-Anfragen) auf folgende HTTP-Anfragen nicht reagiert. Die Gründe für dieses Fehlverhalten konnten im Rahmen des Projektes nicht geklärt werden, traten aber auch bei Betreibern anderer Caches auf.

Der Einsatz eines Watchdog zur Überwachung der DFN-Caches und Alarmierung bei vermuteten Betriebsstörungen wurde im weiteren Verlauf des Projektes unverzichtbar. Es zeigte sich entsprechend den Erwartungen, daß sich zwar nicht alle kritischen Betriebssituationen bei der Überwachung berücksichtigt ließen, jedoch eine Vielzahl rechtzeitig gemeldet und behoben werden konnten.

### 4.3.3 Verkehrsstatistiken

Neben den Daten zur Überwachung der Server-Hardware und der Squid-Prozesse wurden weitere Informationen über das eigentliche Aufkommen an HTTP- und ICP-Verkehr auf den DFN-Caches gesammelt. Anhand dieser Daten konnten Aussagen über die Nutzung und den

Nutzen des DFN-Cache-Verbundes und die allgemeine Entwicklung des WWW-Verkehrs im Wissenschaftsnetz getroffen werden.

Die Verkehrsdaten wurden nicht wie die Betriebsstatistiken aus regelmäßigen Abfragen der momentanen Betriebssituation generiert, sondern durch nachträgliche Auswertung der täglich anfallenden Squid-Logfiles. Für das Extrahieren der Daten aus den Logfiles wurde kein eigenes Programm entwickelt, sondern das Perl-Skript *calamaris* eingesetzt. Bei der Erstellung der Verkehrsstatistiken traten im wesentlichen zwei Schwierigkeiten auf:

1. Die Datenmenge. Gegen Ende des Projektes umfaßte ein unkomprimiertes Logfile durchschnittlich mehr als 400 MByte, das täglich anfallende Datenvolumen der Logfiles aller 10 DFN-Cache betrug entsprechend über 4 GByte. Auch wenn durch Komprimierung eine Reduktion des gesamten Volumens auf ca. 700 MByte pro Tag möglich war, erforderte sowohl die Übertragung der Logfiles von den DFN-Caches auf den zentralen Server am RVS sowie die dauerhafte Speicherung der Logfiles auf dem Archivserver des RRZN ein hohes Maß an Ressourcen.
2. Die Dauer der Auswertung. Die Dauer einer vollständigen Auswertung aller 10 Logfiles betrug an Wochentagen ca. 14 Stunden. Auch wenn es noch keine annähernde Überschneidung der Auswertung aufeinanderfolgender Tage gab, war eine spontanes Extrahieren von Informationen nahezu unmöglich.

Die Darstellung der Verkehrsstatistiken in grafischer Form stellte ein weiteres Problem dar. Da die Ausgabe von *calamaris* aus einer umfangreichen Ansammlung von textbasierten Tabellen besteht, mußten in einem zweiten Schritt der Auswertung relevante Informationen aus den Tabellen extrahiert werden. Erst aus diesen Daten konnte in einem letzten Schritt mit Hilfe des Programms *gnuplot* Grafiken erzeugt werden, wie sie im Anhang A.1 aufgeführt sind. Da diese Form der Auswertung im Rahmen des Projektes nicht automatisiert wurde, konnten die Verkehrsstatistiken nur für bestimmte Anlässe erzeugt werden. Die Vorhaltung aktueller Verkehrsstatistiken im WWW wurde zu Beginn des Projektes konzipiert und prototypisch umgesetzt, konnte im weiteren Verlauf des Projektes jedoch nicht ausgebaut in den Dauerbetrieb übernommen werden.

#### 4.4 Nutzung der DFN-Caches

Der Zugang zu den DFN-Caches wurde über den gesamten Projektverlauf nicht eingeschränkt. Hintergrund dieses Vorgehens war es, interessierten Administratoren lokaler Caches zunächst eine schnelle und unkomplizierte Möglichkeit zum Testen der DFN-Caches zu geben. Die Administratoren wurden jedoch gebeten, ihre Caches nach einer nicht näher festgelegten Testphase registrieren zu lassen. Für eine Registrierung waren folgende Angaben erwünscht:

- Institution,
- DNS-Eintrag und IP-Adresse des Caches sowie
- Name und EMail-Adresse des Cache-Administrators.

Am 31.3.1998 umfaßte die Liste der registrierten lokalen Caches 127 Einträge, die sich wie folgt unterteilten:

- 91 Hochschulen, Fachhochschulen,
- 21 Forschungseinrichtungen und
- 15 Provider.

Anhand der im vorangegangenen Kapitel erläuterten Verkehrsstatistiken konnte auch verfolgt werden, daß die DFN-Caches von einer Reihe nicht-registrierter Caches dauerhaft befragt wurden. Versuche, Kontakt mit den Administratoren aufzunehmen um sie ggf. zu einer Registrierung ihres Caches aufzufordern, schlugen in der Regel fehl. Der Anteil am Verkehrsaufkommen, der im DFN-Cache-Verbund durch nicht-registrierte Caches verursacht wurde, betrug gegen Ende des Projektes ca. 10 %.

Im nachhinein zeigte sich, daß der dargestellte weitgehende Verzicht auf eine Zugangskontrolle zu den DFN-Caches nur für einen eingeschränkten Zeitraum empfehlenswert ist. Gegen Ende des Projektes wurden, neben den bereits erwähnten nicht-registrierten Caches innerhalb des Wissenschaftsnetzes, vermehrt Zugriffe von Caches oder Endsystemen außerhalb des Wissenschaftsnetzes festgestellt. In vielen Fällen konnten dauerhafte Zugriffe weitgehend unterbunden werden, indem in den DFN-Caches explizite Zugriffsverbote für bestimmte IP-Subnetze oder Domains konfiguriert wurden. Ein kontinuierliches Vorgehen auf der Basis von Vergleichen zwischen der Liste registrierter Nutzer und den Verkehrsstatistiken wäre jedoch mit einem hohen Zeitaufwand verbunden gewesen, so daß hier lediglich stichprobenartig vorgegangen werden konnte.

## 4.5 Konzeption DFN-Caches

Bereits vor der Inbetriebnahme der 10 DFN-Caches mußte die Frage geklärt werden, welche Konzeption dem DFN-Cache-Verbund zugrundeliegen sollte. Ausgangspunkt der Überlegungen war zunächst die Toplogie des Breitband Wissenschaftsnetzes (B-WiN). Abbildung 5 zeigt das B-WiN mit den Anbindungen an externe Netze in der Ausbaustufe zum Zeitpunkt des Projektendes. Zu Beginn des Projektes wies das B-WiN eine abweichende Konfiguration auf. Die Verbindung in die USA hatte ihren Aufpunkt in Köln, entsprechend waren die zentralen Knoten in Berlin und Köln miteinander verbunden. Mit der Inbetriebnahme der eigenen USA-Verbindung zwischen Frankfurt und Washington wurde auch die Anbindung des Knotens Berlin von Köln nach Frankfurt geschwenkt.

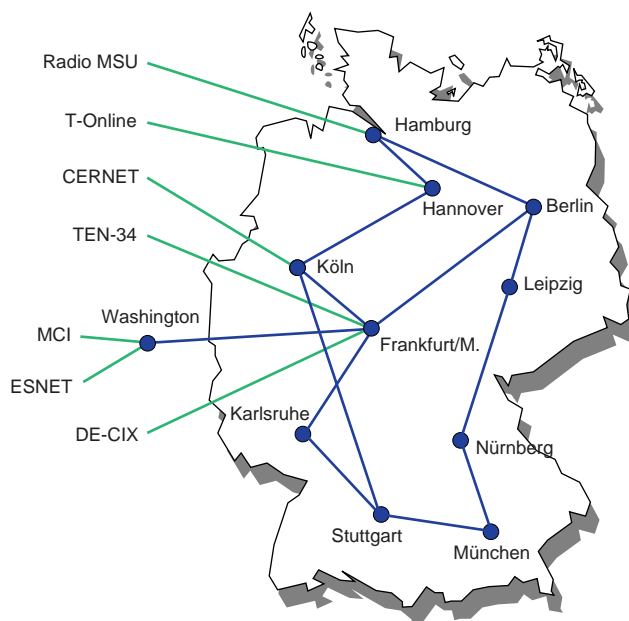


Bild 5: Breitband Wissenschaftsnetz mit externen Anbindungen (März 1998)

Weitgehend unabhängig von der exakten Gestaltung des B-WiN war jedoch die zentrale Fragestellung bei der Konzeption des DFN-Cache-Verbundes, wie eine möglichst gleichmäßige Verteilung der Last und des Verkehrsaufkommens auf die DFN-Caches erreicht werden konnte. Auf dem Wege zu einer unkomplizierten und dennoch leistungsfähigen Lösung wurden verschiedene Konzeptionen diskutiert. Zwei unterschiedliche Konfigurationen des DFN-Cache-Verbundes wurden schließlich umgesetzt, wobei die sich zweite in wesentlichen Teilen aus den Betriebserfahrungen mit dem ersten Ansatz ergab.

#### 4.5.1 Ursprüngliche Konzeption

Abgesehen von dem Projekt des NLANR existierte zu dem Zeitpunkt des Projektbeginns kein Cache-Verbund vergleichbarer Größe. Daher wurde in einem ersten Versuch die Konzeption des NLANR nahezu unverändert übernommen. Die Konzeption sah eine Verteilung der vermeintlich wichtigsten 50 Toplevel Domains auf die DFN-Caches vor (Tabelle 2).

Toplevel Domain	DFN Cache
.cz .hu .pl .ru .si .sk .su .tr	Berlin
.com	Frankfurt, Köln
.cn .id .in .jp .kr .my .sg .th .tw .vn	Hamburg
.dk .fi .gr .is .no .se .sp .pt	Hannover
.at .be .lu .ch .fr .it .nl	Karlsruhe
.de	Leipzig
.net	München
.edu .gov .int .mil .org .us	Nürnberg
.au .br .ca .ie .mx .nz .uk .za	Stuttgart

Tabelle 2: Verteilung von 50 Toplevel Domains auf die DFN-Caches

Zu beachten ist hier zunächst die Zuordnung der Toplevel Domain .com auf zwei DFN-Caches. Da zu Projektbeginn der Aufpunkt für die Transatlantikstrecke in Köln lag, wurde hier der erste DFN-Cache für .com plaziert. Im Frühjahr 1997 wurde der Aufpunkt nach Frankfurt verschoben, womit auch dem DFN-Cache in Frankfurt Objekte der Domain .com zugewiesen wurden. Da bereits zu diesem Zeitpunkt anhand der Verkehrsstatistiken belegt werden konnte, daß der .com-Verkehr einen großen Anteil am gesamten WWW-Verkehr ausmachte, wurden zwei DFN-Caches dauerhaft für die .com-Objekte bereitgestellt. Für die Toplevel Domains .de und .net wurde jeweils ein dedizierter DFN-Cache vorgesehen, die übrigen 47 Toplevel Domains wurden nahezu beliebig auf die übrigen 6 DFN-Caches verteilt.

Die Verteilung der Toplevel Domains mußte auch bei der Konfiguration der DFN-Caches untereinander berücksichtigt werden. Um die Ausprägung der DFN-Caches auf bestimmte Toplevel Domains gemäß Tabelle 2 zu unterstreichen, fiel die Entscheidung auf domain-abhängige Parent-Beziehungen. Somit wurden jegliche Requests lokaler Caches an den für die jeweilige Toplevel Domain zuständigen DFN-Cache durchgereicht.

Die lokalen Caches sollten die DFN-Caches unabhängig von deren Zuordnung zu bestimmten Toplevel Domains befragen. Auf den lokalen Caches sollten lediglich die zwei oder maximal drei DFN-Caches als Parents konfiguriert werden, die netztechnisch gesehen am günstigsten zu erreichen waren. Somit ergab sich die in Abbildung 6 dargestellte dreistufige Hierarchie im DFN-Cache-Verbund.

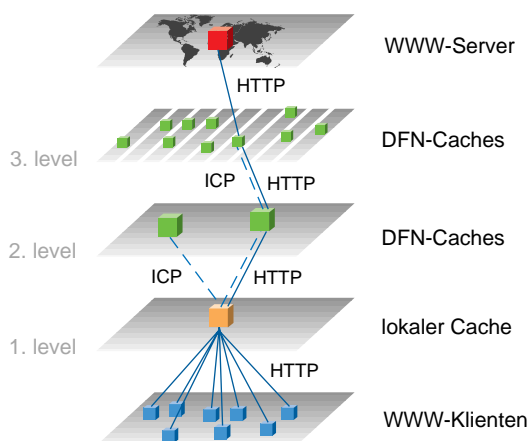


Bild 6: Dreistufige Cache-Hierarchie, ursprüngliche Konzeption

Bereits nach wenigen Wochen Dauerbetrieb zeigten sich die ersten Mängel in der gewählten Konzeption:

- Eine gleichmäßige Verteilung des Verkehrsaufkommens konnte anhand der vorgenommenen Zuweisung von Toplevel Domains gemäß Tabelle 2 nicht erreicht werden. Deutliche Ausprägungen der Verkehrsstatistiken bezüglich der Toplevel Domains (Anhang A.1.13) im .com- und .de-Verkehr zeigten, daß die gewählte breite Verteilung von 50 Toplevel Domains nicht sinnvoll war.
- Bei auftretenden Betriebsstörungen machte sich ebenfalls negativ bemerkbar, daß abgesehen von den .com-Caches kein Backup für eine Toplevel Domain vorhanden war. So führte ein längerer Ausfall des DFN-Caches in München dazu, daß innerhalb des DFN-Cache-Verbundes keine WWW-Objekte der Domain .net verfügbar waren und durch die anderen 9 DFN-Caches von den WWW-Servern abgerufen werden mußten. Da innerhalb der verbliebenen 9 DFN-Caches keine Nachbar-Beziehungen bezüglich der Toplevel Domain .net vorgesehen waren, konnte zudem kein interner Austausch von .net-Objekten vorgenommen werden. Da auch auf dem DFN-Cache in München keine Abfragen nach .net-Objekten auf anderen DFN-Caches vorgesehen war, mußten nach dessen Inbetriebnahme die betroffenen Objekte erneut von den WWW-Servern übertragen werden.
- Der Versuch, mit zwei DFN-Caches dem hohen Anteil .com-Verkehr wirksam entgegenzutreten zu können, mußte bald als falsch bemessen angesehen werden. Das Aufkommen an .com-Verkehr betrug ca. 50% des gesamten Verkehrs. Daher neigten gerade die beiden für .com-Verkehr zuständigen DFN-Caches in Frankfurt und Köln zu häufigen Ausfällen aufgrund von Überlastung. Besonders deutlich wurde diese Situation, als wegen eines Hardware-Defektes der DFN-Cache in Köln für einen längeren Zeitraum ausfiel. Bereits wenige Minuten nach Beginn des Ausfalls war der DFN-Cache in Frankfurt derart überlastet, daß er für die Dauer der Störung in Köln ebenfalls außer Betrieb genommen werden mußte.

- Im Fall eines Cache-Misses in der gesamten Cache-Hierarchie wurde der Request durch drei Caches geleitet. Hierdurch entstand insbesondere in Verbindung mit überlasteten DFN-Caches eine erhebliche Verzögerung, die aus Sicht vieler Nutzer nicht zu vertreten war.

Ausgehend von der dargestellten Situation wurden im Sommer 1997 Überlegungen für eine neue Konzeption des DFN-Cache-Verbundes vorangetrieben.

#### 4.5.2 Verbesserte Konzeption

Die Basis für eine neue Konzeption des DFN-Cache-Verbundes bildeten die Analysen der im bisherigen Projektverlauf gesammelten Verkehrsstatistiken. Die Statistiken wiesen darauf hin, daß ca. 50% des gesamten WWW-Verkehrs durch Objekte der Toplevel Domain .com verursacht wurde. Die verbleibende Hälfte des Verkehrs konnte zunächst sehr einfach einigen weiteren Toplevel Domains zugeordnet werden. Dennoch wurden keine weiteren Anstrengungen unternommen, diese Toplevel Domains weiter aufzuschlüsseln. Neben der Toplevel Domain .com wurde der restliche Verkehr der logischen Toplevel Domain !.com (nicht .com) zugewiesen. Entsprechend der Verkehrsverteilung wurden jeweils fünf DFN-Caches für jede der beiden Domains vorgesehen (Tabelle 3).

Toplevel Domain	DFN Cache
.com	Berlin, Frankfurt, Hannover, Nürnberg, Stuttgart
!.com	Köln, Hamburg, Karlsruhe, Leipzig, München

Tabelle 3: Verteilung Toplevel Domains, 2. Konzeption

Darüber hinaus konnte mit der neuen Verteilung auf die Parent-Beziehungen der DFN-Caches untereinander verzichtet werden. Alle fünf DFN-Caches innerhalb einer logischen Toplevel Domain unterhielten in der neuen Konzeption lediglich Sibling-Beziehungen untereinander. Zu der jeweils anderen Gruppe von DFN-Caches wurden keine Beziehungen konfiguriert. Somit bestand der DFN-Cache-Verbund aus einer einzigen Stufe, woraus sich eine Reduzierung der gesamten Cache-Hierarchie von drei auf zwei Stufen ergab (Abbildung 7).

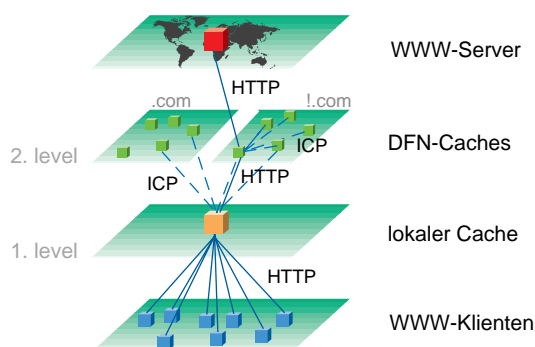


Bild 7: Zweistufige Cache-Hierarchie, verbesserte Konzeption

Für den Anschluß an den neuen DFN-Cache-Verbund sollten in den lokalen Caches vier DFN-Caches konfiguriert werden, jeweils zwei für eine logische Toplevel Domain. Die neue Konzeption wurde innerhalb der Mailingliste de-cache vorgestellt und nach kurzer Diskussion im Oktober 1997 umgesetzt. Anhand der Verkehrsstatistiken konnten sofort wesentliche Verbesserungen belegt werden:

- Das von den DFN-Caches an lokale Caches ausgesendete Datenvolumen konnte von ca. 30 GByte/Tag auf annähernd 60 GByte/Tag verdoppelt werden (Anhang A.1.12, sent).
- Das Verkehrsaufkommen zwischen den DFN-Caches untereinander sank deutlich ab (Anhang A.1.12, among).
- Der Ausfall eines DFN-Caches wurden durch die vier verbleibenden Caches der jeweiligen Gruppe kompensiert. Somit konnten die in Kap. 4.5.1 dargelegten Kettenreaktionen (Ausfall Köln, Zusammenbruch Frankfurt) nahezu ausgeschlossen werden.
- Die Reduzierung der Cache-Hierarchie auf zwei Stufen führte zu einer spürbaren Verringerung der Wartezeiten.

Wie erwartet kam es mit dem erhöhten Verkehrsaufkommen auch zu einem Anstieg der Last auf den DFN-Caches, die sich insbesondere durch übermäßige Allozierung von Hauptspeicher durch die Squid-Prozesse bemerkbar machte. Bereits nach wenigen Tagen mußten alle DFN-Caches auf eine Version der Squid-Software umgestellt werden, die durch einen deutlich geringeren Verbrauch an Hauptspeicher gekennzeichnet war. Der Nachteil dieser Version bestand in einer reduzierten Performance, die jedoch anhand der Verkehrstatistiken nicht nachgewiesen werden konnte.

### 4.5.3 Weiterführende Überlegungen

Trotz der überzeugenden Ergebnisse mit der neuen Konzeption gab es weitere Überlegungen, die Effektivität des DFN-Cache-Verbundes zu steigern. Die weitere Beobachtung der Zugriffsstatistiken wies einen zum Teil erheblichen Anteil an falsch konfigurierten lokalen Caches auf, die sich durch anhaltendes Abfragen von .com-Objekte an DFN-Caches der !.com-Domain bemerkbar machten. Obwohl die meisten Administratoren direkt angesprochen werden konnten, konnte dieser störende Mißstand nicht vollständig ausgeräumt werden. Neben falsch konfigurierten Caches führte auch Cache-Software ohne das Feature der Domain-basierten Konfiguration zu dieser Situation. Als Ausweg wurde die Konfiguration von Nachbar-Beziehungen zwischen .com- und !.com-Caches diskutiert. Eine nähere Untersuchung konnte jedoch im Rahmen des Projektes nicht mehr vorgenommen werden.

Eine weitere Möglichkeit zur potentiellen Verbesserung des DFN-Cache-Verbundes lag in der Verwendung der Option `proxy-only`. In der in Kap. 4.5.2 dargelegten Konstellation speicherte jeder DFN-Cache sämtliche Objekte, die er an die lokalen Caches weiterleitete. So war durchaus die Situation gegeben, daß große Binärdateien wie z.B. eine Version des Netscape Communicators gleichzeitig auf allen fünf .com-Caches gespeichert wurde. Hier hätte ein Feature von Squid sinnvoll eingesetzt werden können. Mit der Kennzeichnung eines Nachbarn als `proxy-only` in der eigenen Konfigurationsdatei wurden dessen Objekte zwar abgerufen und weitergeleitet, nicht aber in dem eigenen Speicher abgelegt. So hätten die zur Verfügung stehenden 60 GByte Plattenkapazität vollständig für *verschiedene* WWW-Objekte genutzt und Kopien bzw. Doubletten vermieden werden können. Auch diese Variante konnte im Rahmen des Projektes nicht mehr untersucht werden.

## 4.6 Weitere Arbeiten

### 4.6.1 Teilnahme am internationalen Cache-Verbund com-Mesh

In einem von der TERENA Caching Task Force gestarteten Experiment wurde ein Verbund europäischer Toplevel Caches gebildet. Innerhalb dieses Verbundes sollten lediglich WWW-Dokumente aus der Toplevel Domain .com austauscht werden. Da zu dem damaligen Zeitpunkt die ursprüngliche Konzeption des DFN-Cache-Verbundes in Betrieb war, wurden die

beiden .com-Caches in Frankfurt und Köln in das Experiment eingebunden. Als Nachbarn der beiden Caches wurden die Toplevel Caches von UNINETT (Norwegian academic network for research and education) und SWITCH (Swiss Academic & Reserach Network) konfiguriert. Im Gegenzug wurden die DFN-Caches von einer Reihe anderer europäischer Toplevel Caches befragt.

Es zeigte sich, daß bei dem Aufbau von Cache-Verbünde im Bereich der Weitverkehrsnetze verschiedene neue Aspekte beachtet werden müssen. Bei Berücksichtigung der hohen Zahl an Requests an die DFN-Caches war es sinnvoll, diese Anfragen nur an Nachbar-Caches ähnlicher Leistungsfähigkeit weiterzuleiten. Weiterhin führten Client-Caches mit schlechter Anbindung zu Problemen, da sie Objekte nur mit sehr geringen Übertragungsraten abrufen konnten. Dadurch wurden jedoch die Ressourcen auf den DFN-Caches entsprechend über einen längeren Zeitraum belegt.

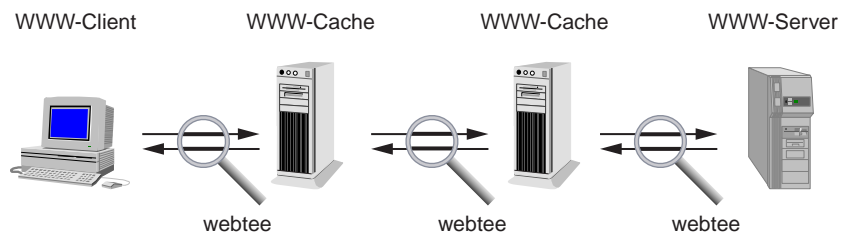
Nach alleiniger Betrachtung der Verkehrsstatistiken konnte das com-Mesh Experiment bei sinnvoller Konfiguration von Nachbar- und Client-Caches durchaus als positiv bewertet werden. Ausschlaggebend für den Abbruch der Teilnahme an dem com-Mesh Experiment war jedoch eine Tatsache, die die Bildung größerer Cache-Verbünde generell fraglich erscheinen läßt. Im Mai 1997 wurde zufällig durch die Mitarbeiter des Projektes bemerkt, daß der DFN-Cache in Frankfurt falsche Objekte auslieferte. Anfragen an <http://www.microsoft.com/> wurden mit der Auslieferung der Homepage einer englischen Hardware-Firma beantwortet. Anhand der Logfiles des DFN-Caches in Frankfurt konnte belegt werden, daß die offensichtlich falschen Objekte von dem SWITCH-Cache stammten. Durch Rücksprache mit dem entsprechenden Cache-Administrator konnte geklärt werden, daß der SWITCH-Cache wiederum die Objekte von einem seiner Nachbar-Cache erhalten hatte.

Um dieser Situation wirksam entgegenzutreten zu können, wurden sofort sämtliche Nachbar-Beziehungen zu Caches außerhalb des DFN-Projektes aufgelöst. Nach manuellem Löschen der entsprechenden Objekte in den DFN-Caches konnte innerhalb kurzer Zeit wieder ein normaler Betrieb gewährleistet werden. Es ließ sich jedoch im nachhinein nicht klären, wie die falschen Objekte in den com-Mesh Verbund gelangt waren. Folglich blieb auch die Frage, ob das Einbringen falscher Objekte durch Vorsatz, falsch konfigurierte Caches oder defekte Hardware verursacht worden war, unbeantwortet. Es konnte lediglich die Erkenntnis festgehalten werden, daß die Nutzung auch von internationalen Nachbar-Caches durchaus zu Einsparungen führen kann, jedoch bisher unbeantwortete Sicherheitsfragen aufwirft. Entsprechend formuliert wurde der dargestellte Sachverhalt mit "If you trust your neighbor cache, then you implicitly trust you neighbor's neighbor, etc...".

#### 4.6.2 Entwicklung von webtee

Im Rahmen des Projektes wurde das Programm *webtee* entwickelt, welches die Darstellung von Header-Informationen in HTTP-Requests und HTTP-Replies ermöglicht. Das Programm

ist so konzipiert, daß es an einem beliebigen Punkt auf der Übertragungsstrecke zwischen WWW-Client, WWW-Caches und WWW-Server installiert werden kann (Abbildung 8).



*Bild 8:* Möglichkeiten für den Einsatz von webtee

Motivation für die Entwicklung von webtee war die Notwendigkeit, im Rahmen einer Fehlersuche die von WWW-Browsern gesendeten HTTP-Informationen zu protokollieren. In der derzeitigen Version basiert webtee auf dem Protokoll HTTP/1.0, eine Erweiterung auf HTTP/1.1 ist geplant. Auf die Erläuterung von Details in der Implementierung von webtee soll an dieser Stelle verzichtet werden. Stattdessen werden im Anhang A.2 einige Beispiele aufgeführt, die den Nutzen von webtee verdeutlichen. Weitere Informationen sowie auch die Quellen von webtee sind von dem projekteigenen WWW-Server abrufbar.

## 5 Ergebnisse des Projektes

Mit dem im vorangegangenen Kapitel dargestellten Verlauf des Projektes wurden bereits einige Erkenntnisse und daraus getroffene Entscheidungen für das DFN-Cache-Projekt erläutert. Daher gilt es in diesem abschließenden Kapitel lediglich, die wesentlich Erkenntnisse zusammenzufassen.

### 5.1 Nutzen des DFN-Cache-Verbundes

Anhand der Verkehrsstatistiken ließ sich der Nutzen des DFN-Cache-Verbundes einfach bemessen. Eine Aufteilung der im DFN-Cache-Verbund beobachteten Verkehrsströme ist in Abbildung 9 dargestellt. Die aus den Verkehrsstatistiken ermittelten Größen waren:

- In*            Verkehr zwischen DFN-Caches und WWW-Servern.
- Out*            Verkehr zwischen DFN-Caches und 1. Level Caches.
- Among*        Verkehr zwischen DFN-Caches untereinander.

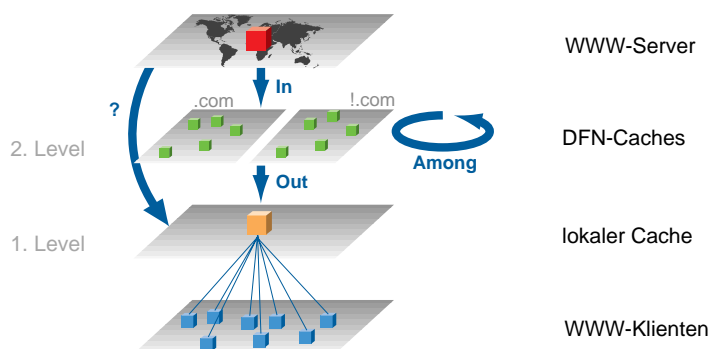


Bild 9: Datenströme in der Verkehrsbilanz

Die vierte Größe in Bild 9, der mit ? markierte Verkehr zwischen WWW-Servern und lokalen Caches bzw. WWW-Klienten, ließ sich nur anhand von Router-Statistiken im B-WiN ermitteln. Die entsprechenden Messungen mußten durch das DFN-NOC auf den ZRs durchgeführt werden. Im Rahmen des Projektes standen hier nur über kurze Zeiträume ermittelte Werte zur Verfügung, die demzufolge mit einer relativ hohen Unsicherheit behaftet waren.

In den Diagrammen in Anhang A.1.11 und A.1.12 sind die zeitlichen Verläufe der Verkehrsbilanzen im DFN-Cache-Verbund aufgetragen. Für die Ermittlung der Einsparungen ist die Bilanz über das Datenvolumen in Anhang A.1.12 entscheidend. Zunächst ist hier die deutliche Veränderung ab Mitte Oktober 1997 mit der Einführung der neuen Konzeption (Kap. 4.5.2) zu erkennen. Sowohl das Aufkommen von In- als auch Out-Verkehr konnte deutlich gesteigert werden, wogegen sich der Among-Verkehr stark reduzieren ließ. Das Verhältnis von Out- zu In-Verkehr blieb nahezu konstant bei 1,45.

Die durch den DFN-Cache-Verbund gewonnenen Einsparungen ergaben sich aus der Differenz zwischen Out- und In-Verkehr. Sie betrug an Wochentagen bis zu 20 GByte. Für eine Bewertung dieser 20 GByte ist weiterhin zu berücksichtigen, daß die Einsparungen an zwei Stellen im Wissenschaftsnetz gleichzeitig auftraten: zum einen wurden die 20 GByte fast vollständig aus dem Verkehr im Backbone des B-WiN ferngehalten, zum anderen wurde ein

beträchtlicher Teil davon (z. B. Verkehr von .com, .net, .edu) auf den externen Anbindungen eingespart.

Die bereits genannten Schwierigkeiten bei der Messung des gesamten WWW-Verkehrs auf den Leitungen im B-WiN und aus dem B-WiN heraus führten dazu, daß hier keine zuverlässigen Zahlen ermittelt werden konnten. Dennoch ließen sich anhand einiger Angaben des DFN-NOC Näherungen vornehmen. Über die Transatlantikleitung wurden im Frühjahr 1998 14 TByte pro Monat zwischen Washington und Frankfurt übertragen. Das entsprach einem Volumen pro Tag von ca. 470 GByte. Auf den ZRs gewonnenen Statistiken konnten belegen, daß der WWW-Anteil an dem Verkehr bis zu 70 % betrug. Daraus konnte für das Tagesvolumen an WWW-Verkehr eine grobe Schätzung von ca. 330 GByte angegeben werden.

Als weitere Orientierung ließ sich die Summe des übertragenen Datenvolumens innerhalb des B-WiN heranziehen. Für das Frühjahr 1998 wurde das Volumen mit 30 TByte pro Monat oder entsprechend 1 TByte pro Tag angegeben.

### 5.1.1 Bewertung

Verglichen mit den genannten Volumen von 330 GByte auf der Transatlantikstrecke und 1 TByte innerhalb des B-WiN fielen die 20 GByte, die durch den DFN-Cache-Verbund eingespart wurden, kaum ins Gewicht. Dieser Schluß wurde auch durch die Angabe des DFN-NOC gestützt, daß nur ein geringer Teil des Verkehrs auf der Transatlantikstrecke an die IP-Adressen der DFN-Caches gerichtet war.

Dieser zunächst negativen Bewertung steht die Tatsache gegenüber, daß der Gewinn, der durch den DFN-Cache-Service erzielt werden konnte, weitgehend durch dessen Nutzung bestimmt wurde. Zwei Aspekte, die eine deutliche Einschränkung bei der Nutzung des DFN-Cache-Verbundes belegten, mußten im Verlauf des Projektes beobachtet werden:

1. Bereits im 1. Zwischenbericht wurde ausführlich dargelegt, daß aufgrund der überlasteten Server keine wesentliche und langfristige Steigerung des Durchsatzes zu erwarten war. Somit konnte auch - unveränderte Hardware vorausgesetzt - der Gewinn aus dem DFN-Cache-Verbund nicht verbessert werden.
2. Es zeigte sich insbesondere gegen Ende des Projektes, daß der DFN-Cache-Verbund aufgrund der schlechten Performance für große Einrichtungen inattraktiv wurde. Da die Nutzung des DFN-Cache-Service freiwillig war, hätte nur ein besserer Service zu neuen Nutzern und dadurch mehr Gewinn führen können. Gerade große Einrichtungen hätten jedoch mit ihrem hohen Volumen an WWW-Verkehr dem DFN-Cache-Verbund zu erhöhtem Durchsatz verhelfen können.

Als weiteren, schwer meßbaren Gewinn des DFN-Cache-Verbundes waren die bei positiven Cache-Anfragen deutlich verringerten Antwortzeiten bei der Übertragung von WWW-Inhalten zu nennen. Eine erste tiefere Untersuchung dieses Aspektes der Einsparungen durch Caches liegt seit Ende 1997 in dem *Report on the costs and benefits of operating caching services*<sup>1</sup> vor. Dieser Report wurde im Rahmen des EU-Projektes DESIRE (Development of a European Service for Information on Research and Education) erstellt. Mitarbeiter des DFN-Cache-Projektes unterstützten die Studie als Revisor.

---

1. <http://www.surfnet.nl/surfnet/projects/desire/deliver/WP4/D4-2.html> (Dezember 1997)

## 5.2 Entwicklung WWW-Verkehr

Neben einer Betrachtung der Einsparungen durch den DFN-Cache-Verbund lassen die Verkehrsstatistiken weitere Beobachtungen des WWW-Verkehrs zu. Hierbei geht es weniger um das Wachstum des World-Wide Web an sich, sondern um eine inhaltliche Aufschlüsselung des WWW-Verkehrs. So geben die Abbildungen in Anhang A.1.14 und A.1.15 die Unterteilung des WWW-Verkehrs im DFN-Cache-Verbund auf der Basis von Mimetypes wieder.

Zunächst wird in Anhang A.1.14, die Vermutung belegt, daß der weitaus größte Teil an Anfragen im WWW auf HTML-Seiten (Mimetype Text) inklusive Grafiken (Mimetype Image) gerichtet ist. Die Zugriffszahlen auf Objekte anderer Mimetypes liegen annähernd zwei Größenordnungen niedriger. Ein Vergleich mit dem übertragenen Datenvolumen je Mimetype zeigt jedoch, daß der geringe Anteil an Requests nach Objekten des Mimetype Application annähernd das Datenvolumen generiert, daß für die Übertragung der fast 100-fachen Menge für Objekte des Mimetype Image aufgebracht werden muß. Belegt wird diese Beobachtung durch den Verlauf der durchschnittlichen Größe eines Objektes im WWW (Anhang A.1.15). Während die durchschnittliche Größe eines Objektes vom Mimetype Image ca. 10 KByte beträgt, liegt das Volumen eines Objektes vom Mimetype Application zwischen 400 und 1.000 KByte.

Aus dem zuletzt genannten Diagramm läßt sich eine weitere interessante Beobachtung ableiten. Die durchschnittliche Größe eines Objektes vom Mimetype Image sank während des Projektes von ca. 13 KByte auf annähernd 7 KByte ab. Eine Begründung für diese Entwicklung wird in verbesserten Verfahren zur Bildkompression vermutet. Weiterhin ist bekannt, daß zunehmend Bilddateien geringerer Größe (z. B. 1x1 Pixel) für das gestalterische Layout von WWW-Inhalten verwendet werden.

Die entgegengesetzte Entwicklung läßt sich für Objekte des Mimetype Text erkennen. Betrug die durchschnittliche Größe einer HTML-Seite zu Beginn des Projektes weniger als 5 KByte, ist über den gesamten Projektverlauf ein zwar leichter, aber dennoch kontinuierlicher Anstieg zu verzeichnen. Hier könnte die Begründung in der zunehmenden Verwendung von Tabellen oder Frames (aufwendiger HTML-Quellcode) sowie von Meta-Informationen liegen. Auch wenn diese Entwicklung nicht so ausgeprägt wie der Rückgang bei der Größe der Images ist, läßt sich anhand des Diagramms eine deutliche Annäherung in der durchschnittlichen Größe von Objekten der Mimetypes Image und Text aufzeigen.

Aus den genannten Beobachtungen können sich Ansätze für den Entwurf zielgerichteter Cache-Services ergeben. Steht die Einsparung von übertragenem Datenvolumen auf den Übertragungsstrecken im Vordergrund, sollten insbesondere Objekte der Mimetypes Image und Application bei der Konzipierung eines Cache-Verbundes berücksichtigt werden, da sie den weitaus größten Teil des Datenvolumens ausmachen (vgl. A.1.14, Datenvolumen). Ist jedoch die Reduzierung der Wartezeiten bei der Übertragung von WWW-Objekten von besonderem Interesse, sollten Objekte der Mimetypes Image und Text bevorzugt behandelt werden (vgl. A.1.14, Requests).

## 5.3 Fortführung des Projektes

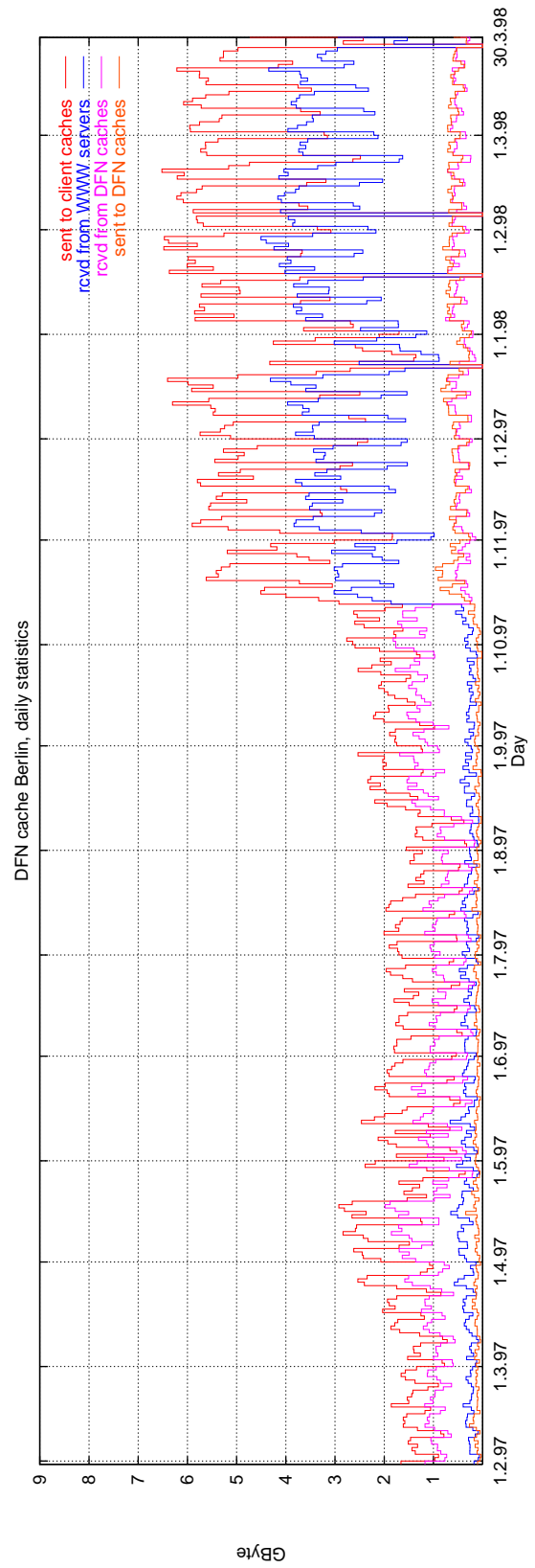
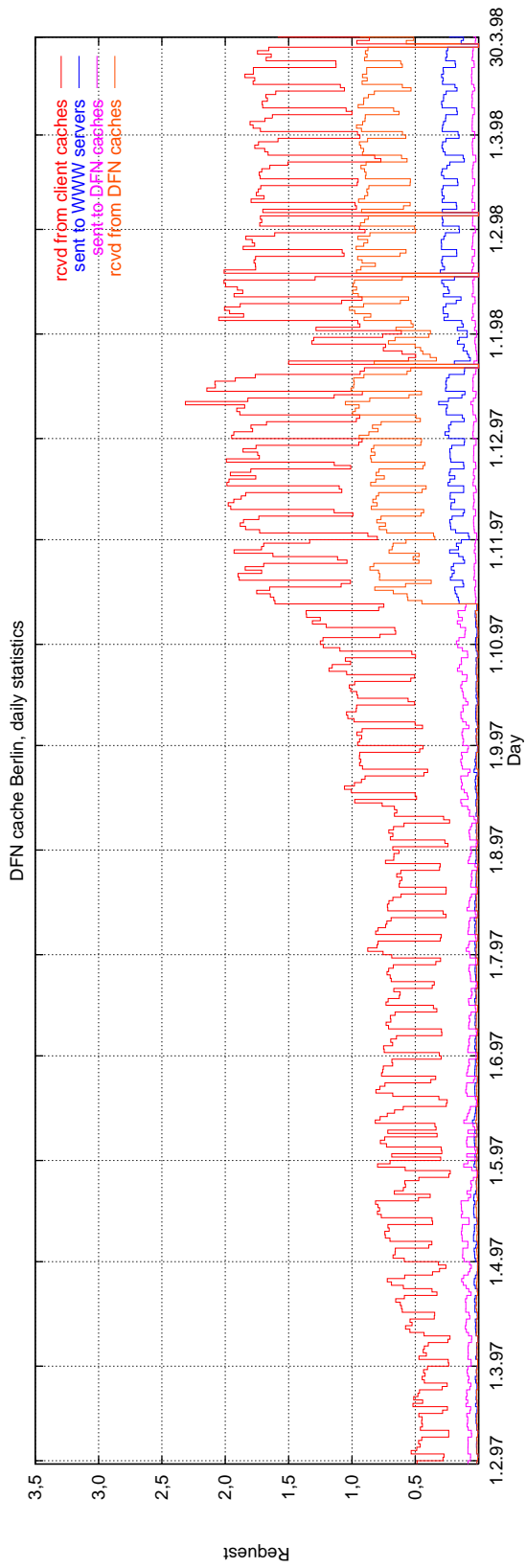
Aus den im vorliegenden Abschlußbericht aufgeführten Erkenntnissen im Aufbau und Betrieb eines Cache-Verbundes lassen sich eine Vielzahl von Vorschlägen für die Fortführung des Projektes zusammenstellen. Neben dem Ausbau der zur Verfügung stehenden Cache-Hardware sollte dabei auch die Formulierung einer konkreten Zielsetzung für einen neuen DFN-Cache-Verbund im Vordergrund stehen.

Als Konsequenz wurde im Frühjahr 1998 ein neuer Projektantrag an den DFN-Verein gestellt. Ziel des Antrags war es, daß Projekt *Konzeption einer Cache-Server-Infrastruktur auf dem Wissenschaftsnetz* in einer zweiten Stufe fortzuführen. Auf der Basis dedizierter Hardware soll ein neuer Verbund von DFN-Caches im Wissenschaftsnetz etabliert werden. Dieser neue DFN-Cache-Verbund wurde in der Antragstellung so konzipiert, daß er den bestehenden Verbund ablösen kann und auf die Reduzierung des WWW-Verkehrs auf wenigen ausgezeichneten externen Leitungen fokussiert ist. Als weiterer wichtiger Arbeitpunkt wird das Forcieren der Nutzung des DFN-Cache-Verbundes angesehen. Entsprechend den Ausführungen in Kap. 5.1.1 kann der Nutzen eines Cache-Verbundes nur durch verstärkte Nutzung signifikant verbessert werden. Da die Nutzung der DFN-Caches durch die lokalen Caches bestimmt wird, müssen deren Administratoren durch die Mitarbeiter des DFN-Cache-Projektes bei ihrer Arbeit unterstützt werden.

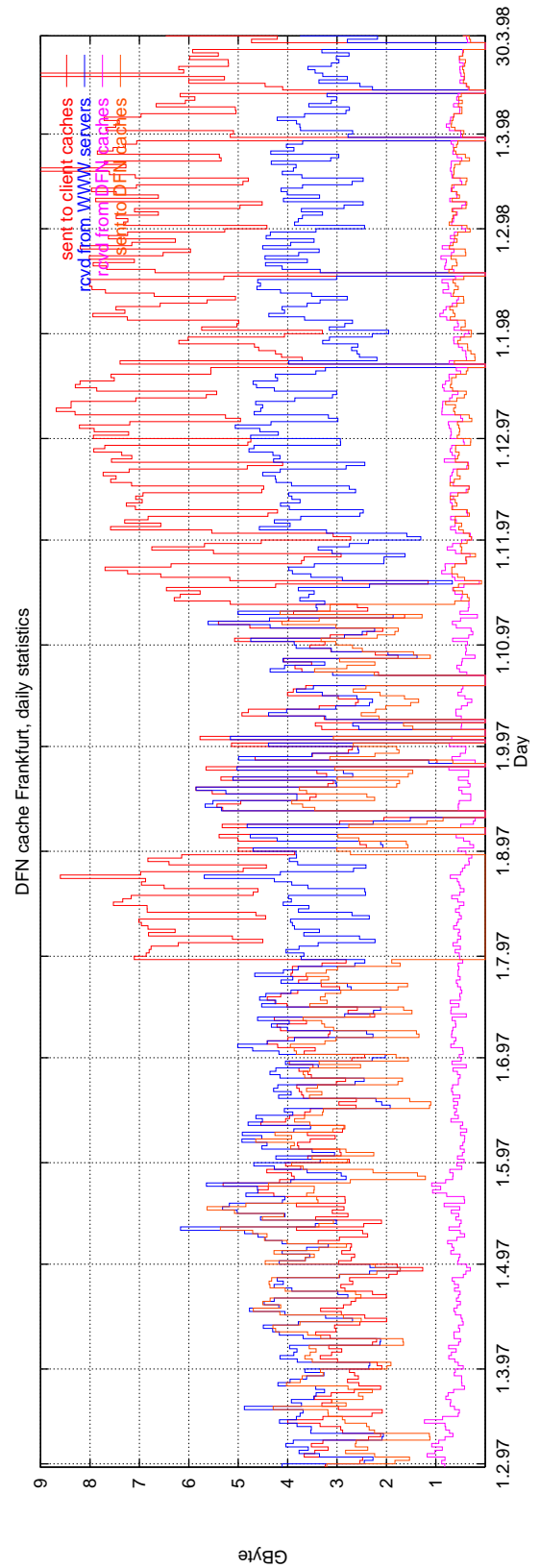
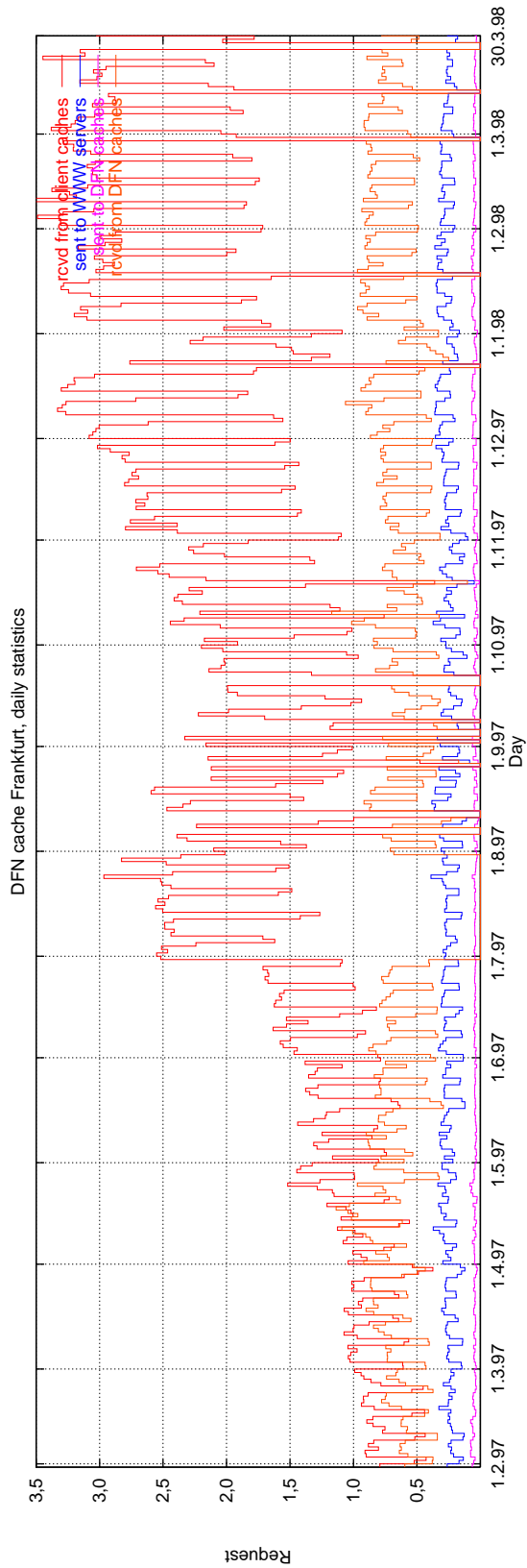
Nach erfolgreicher Annahme des Antrags begann die zweite Stufe des DFN-Cache-Projektes am 1.5.1998.

## **A.1 Verkehrsstatistiken**

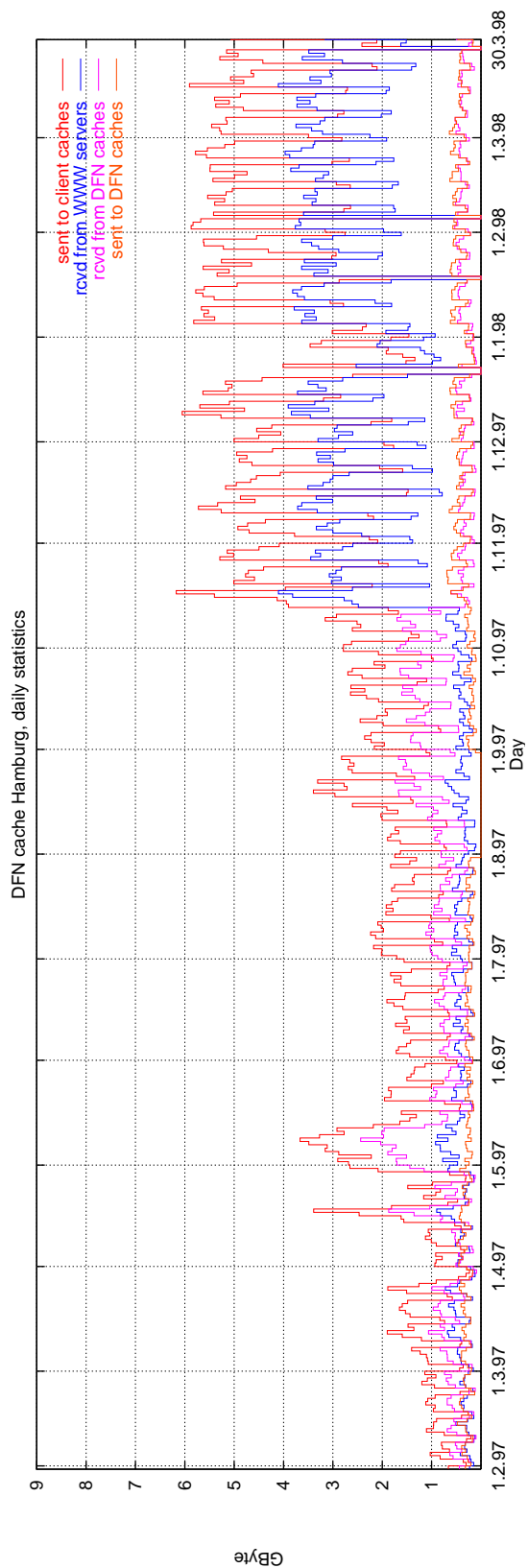
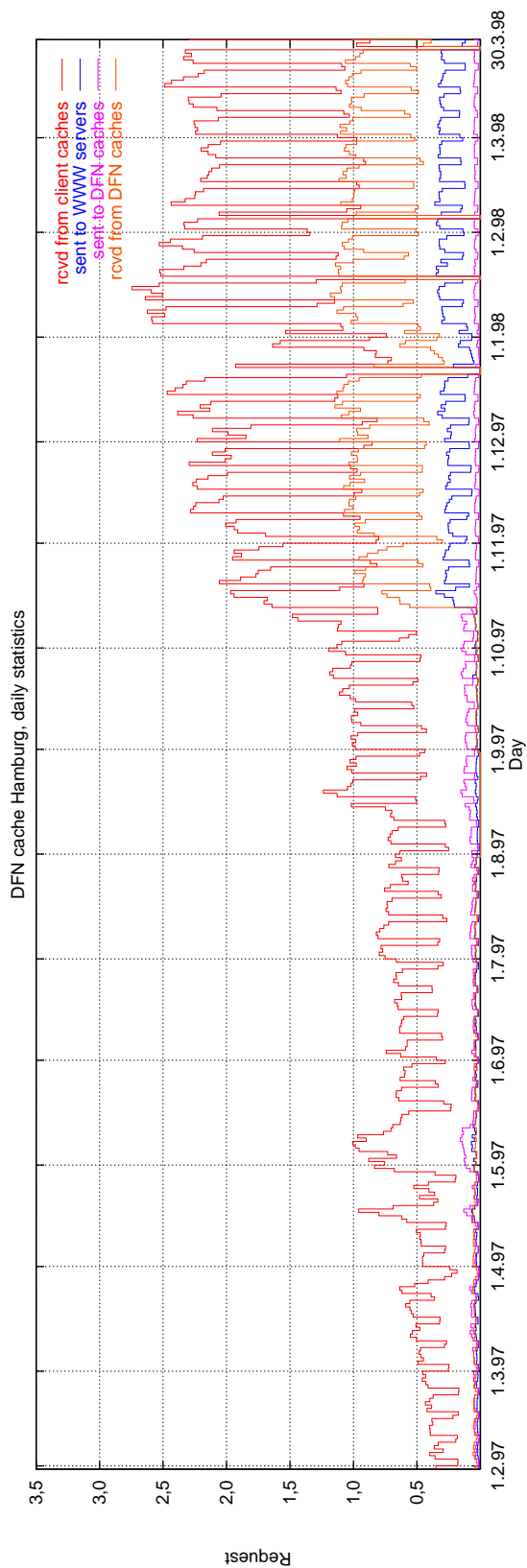
### A.1.1 Verkehrsbilanz Berlin, Requests und Datenvolumen



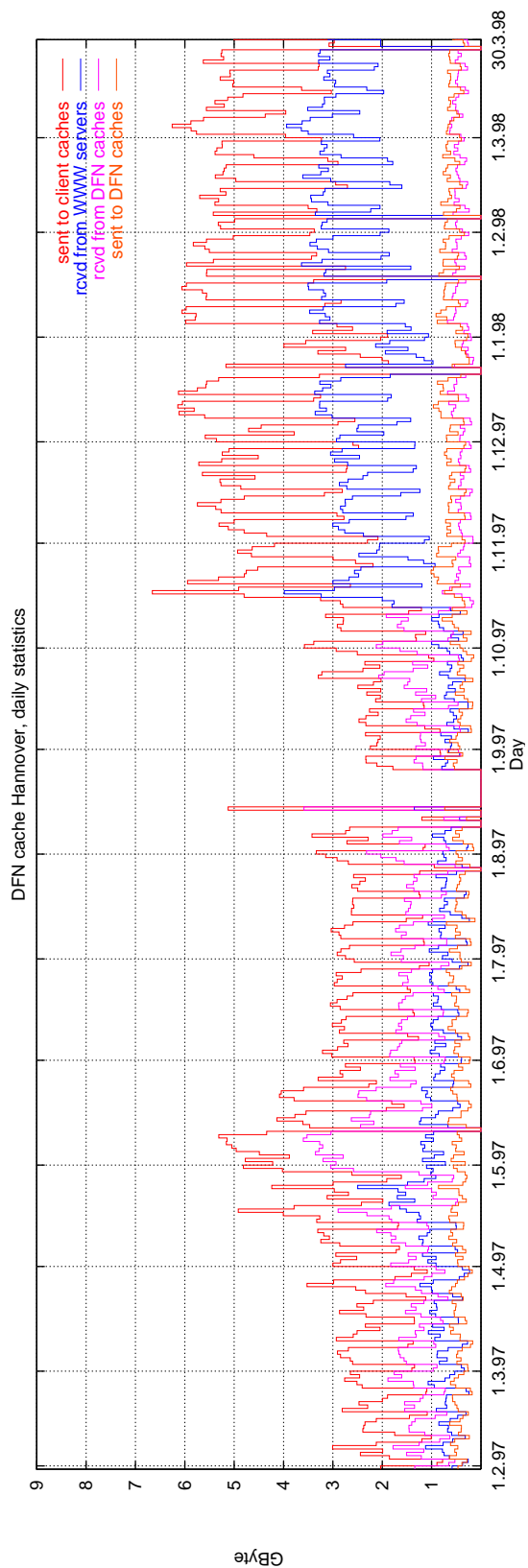
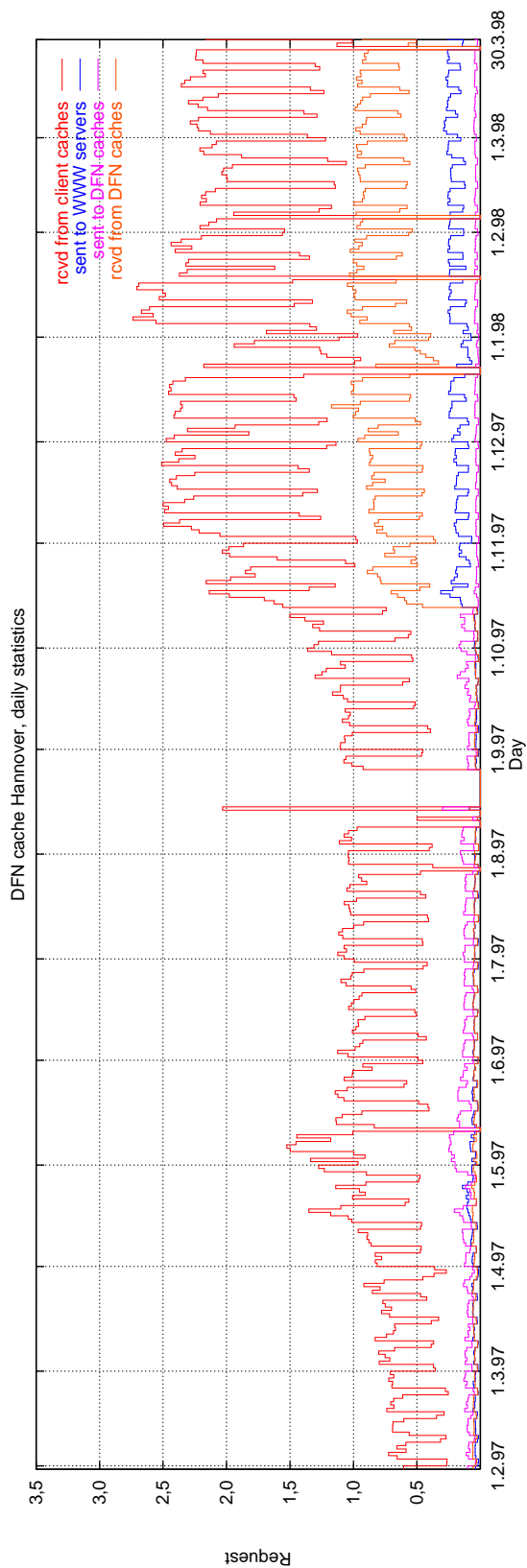
### A.1.2 Verkehrsbilanz Frankfurt, Requests und Datenvolumen



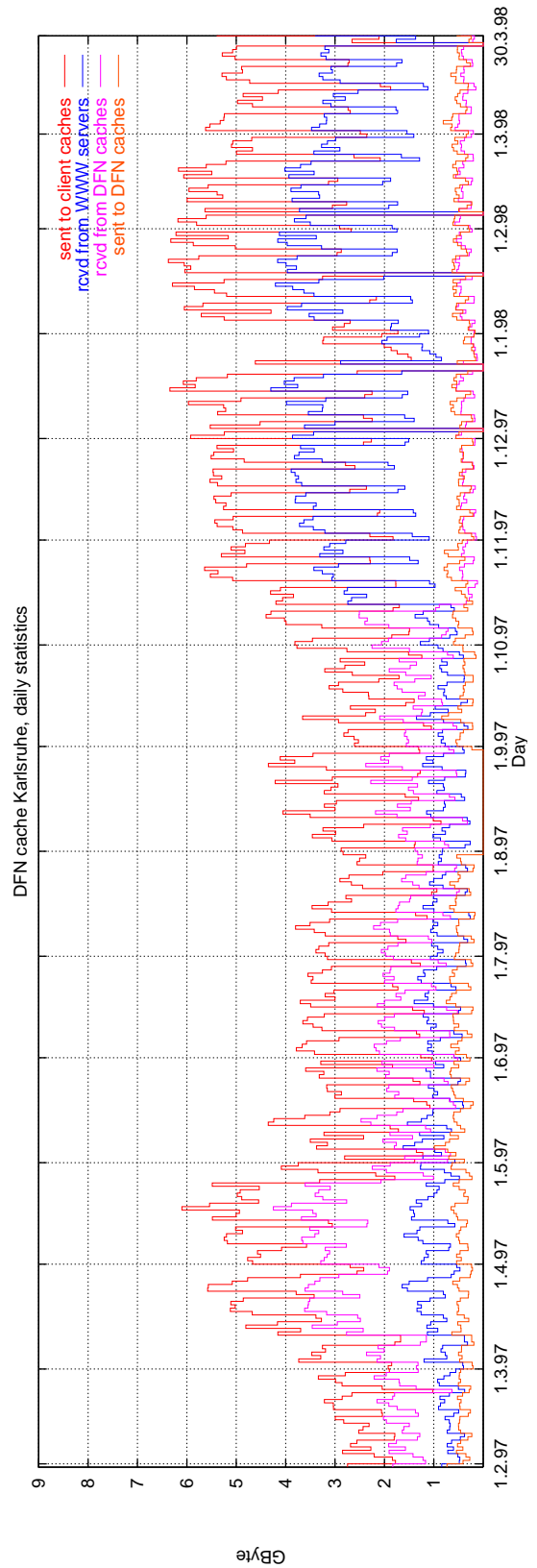
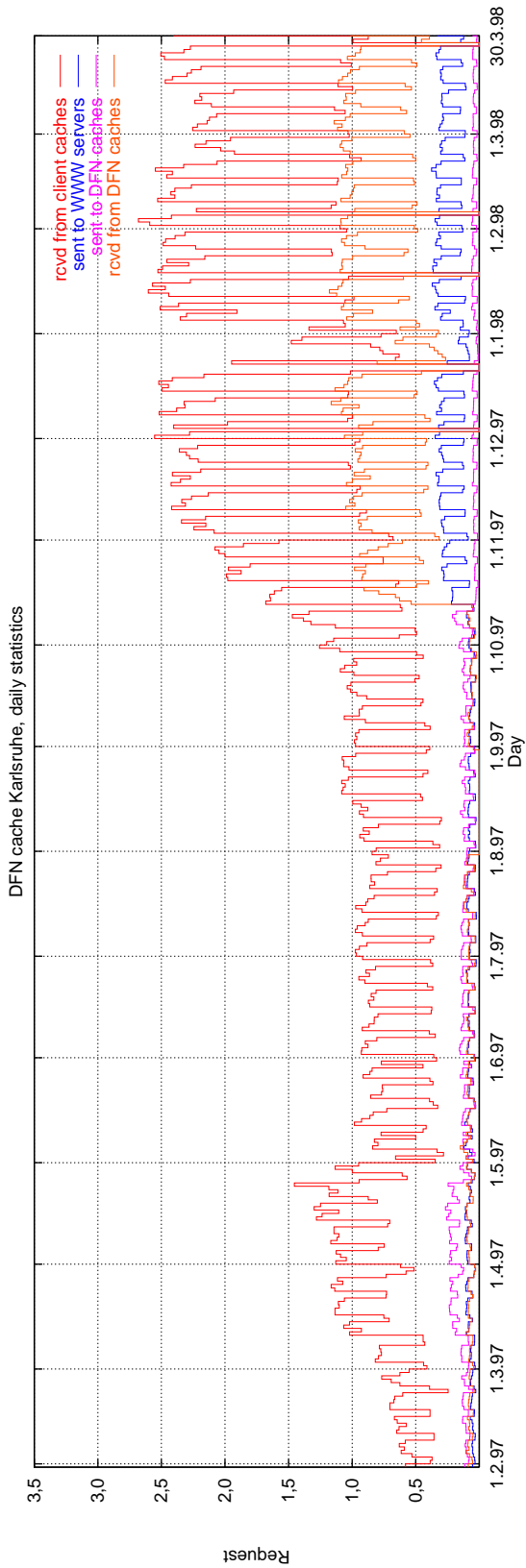
### A.1.3 Verkehrsbilanz Hamburg, Requests und Datenvolumen



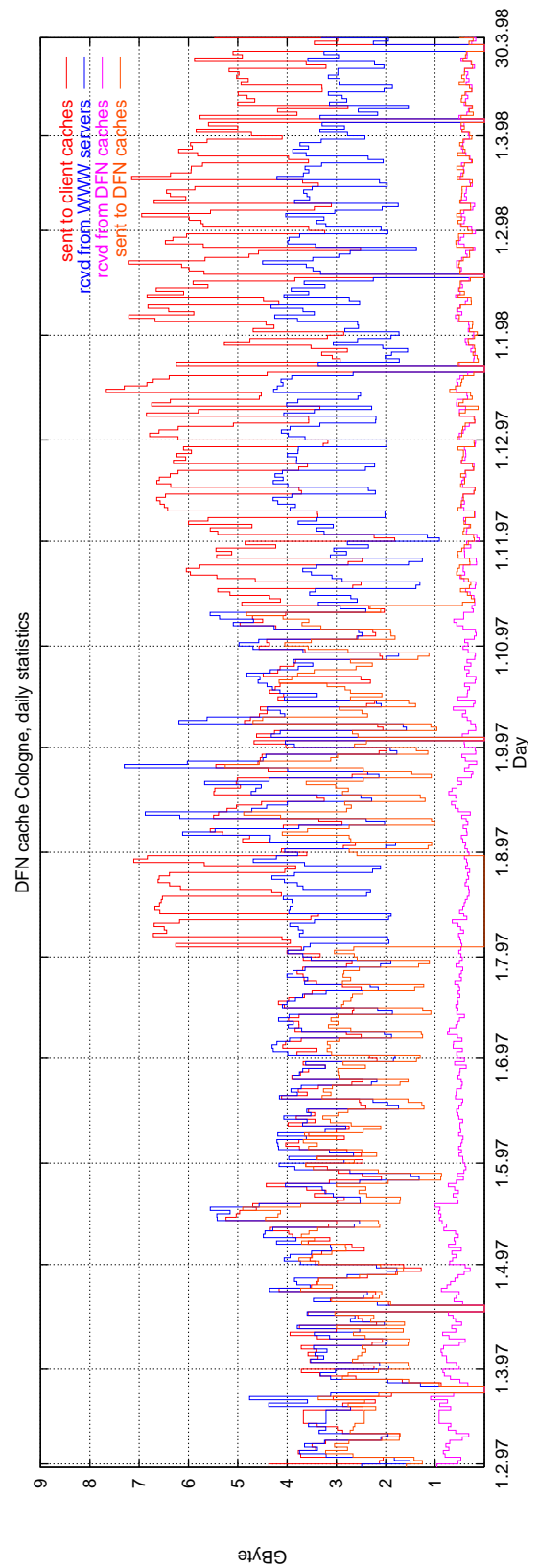
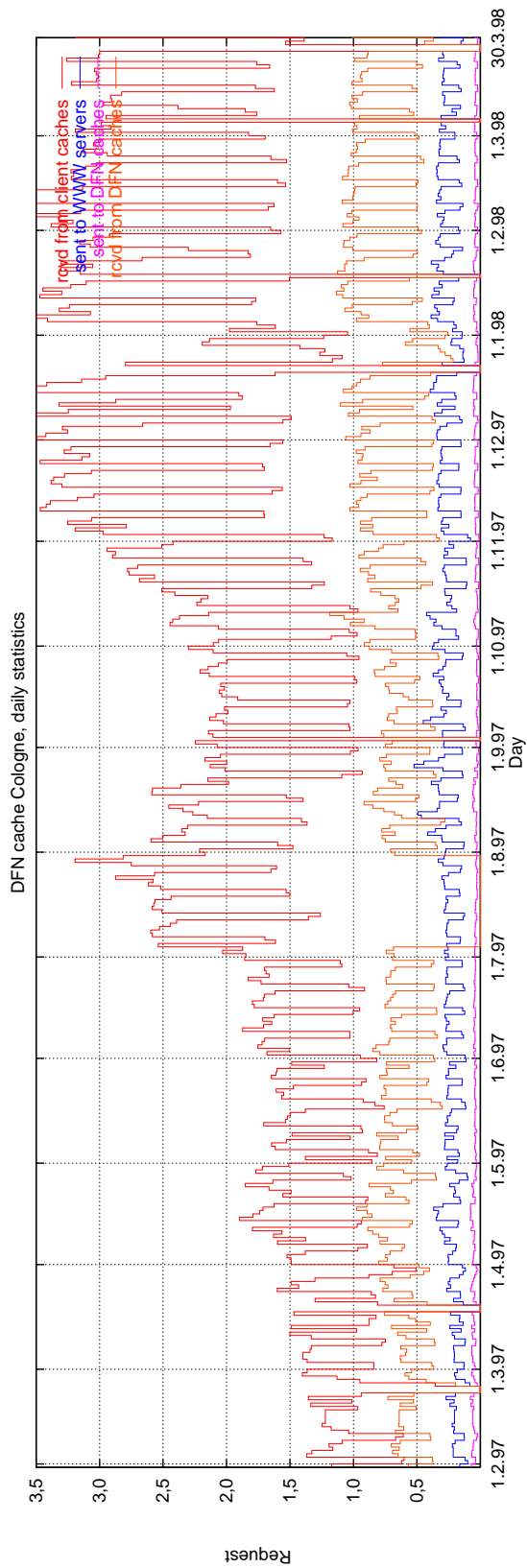
### A.1.4 Verkehrsbilanz Hannover, Requests und Datenvolumen



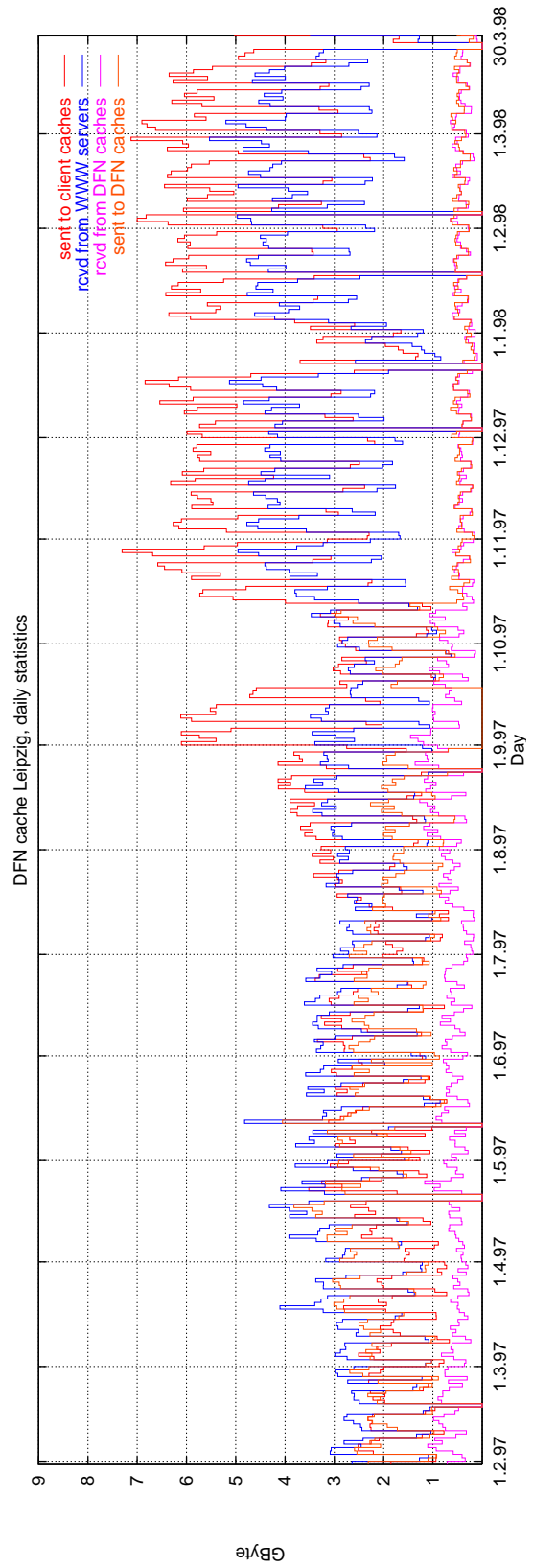
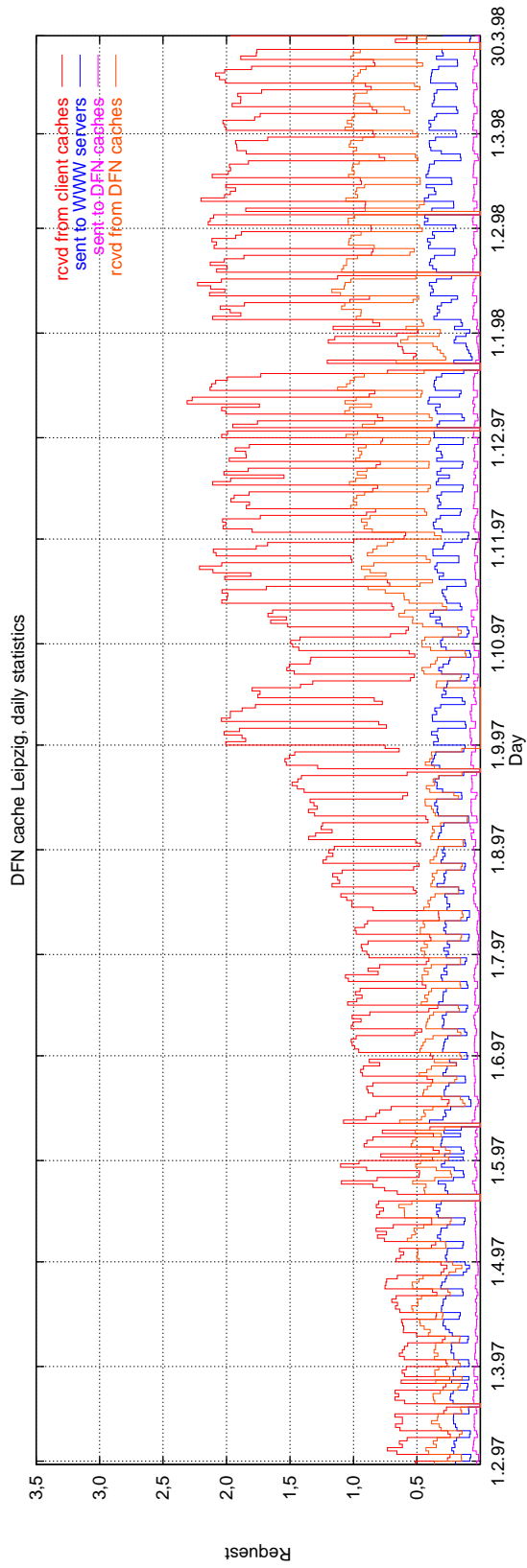
### A.1.5 Verkehrsbilanz Karlsruhe, Requests und Datenvolumen



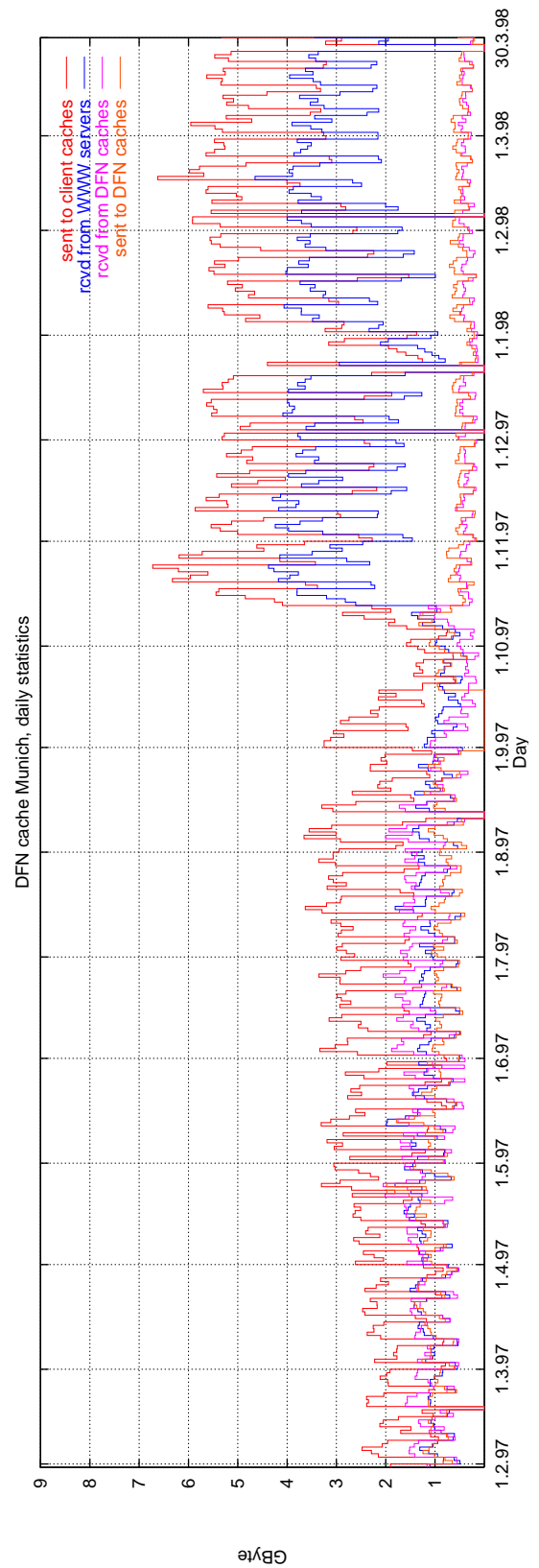
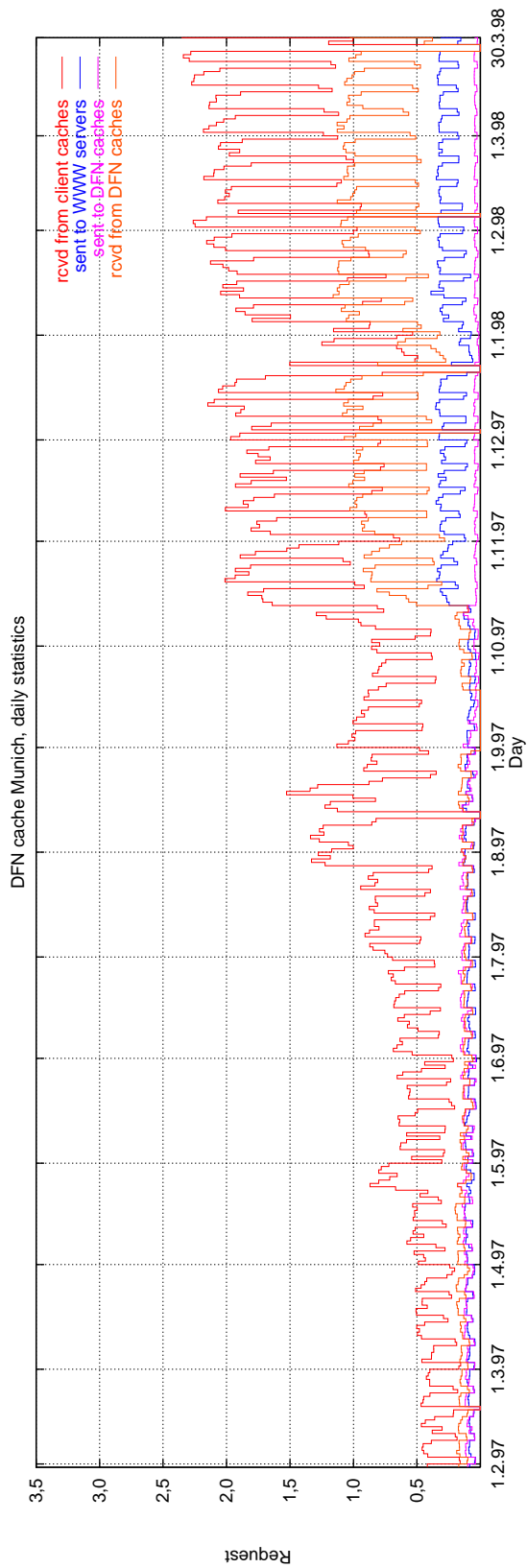
### A.1.6 Verkehrsbilanz Köln, Requests und Datenvolumen



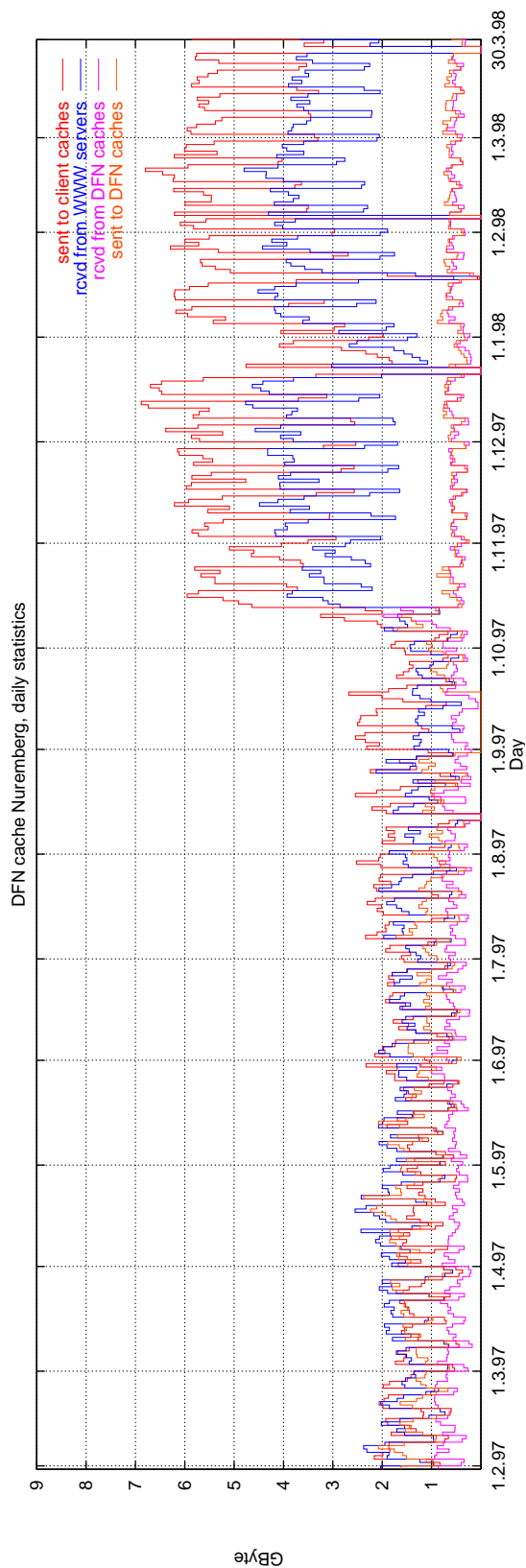
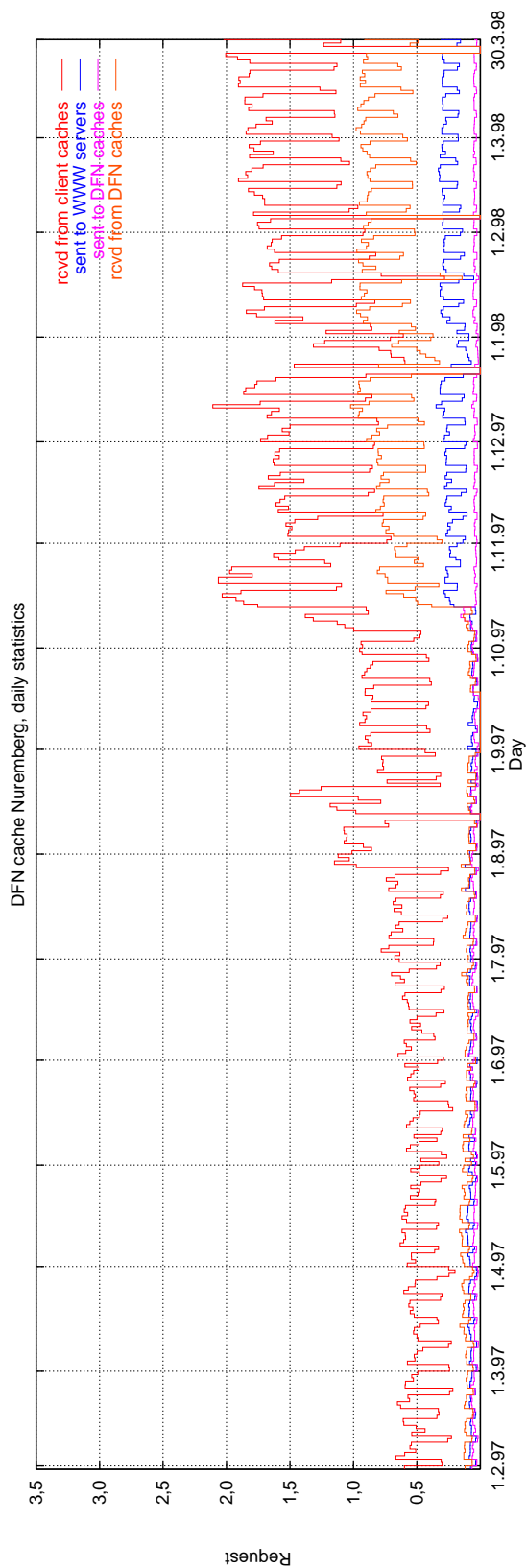
### A.1.7 Verkehrsbilanz Leipzig, Requests und Datenvolumen



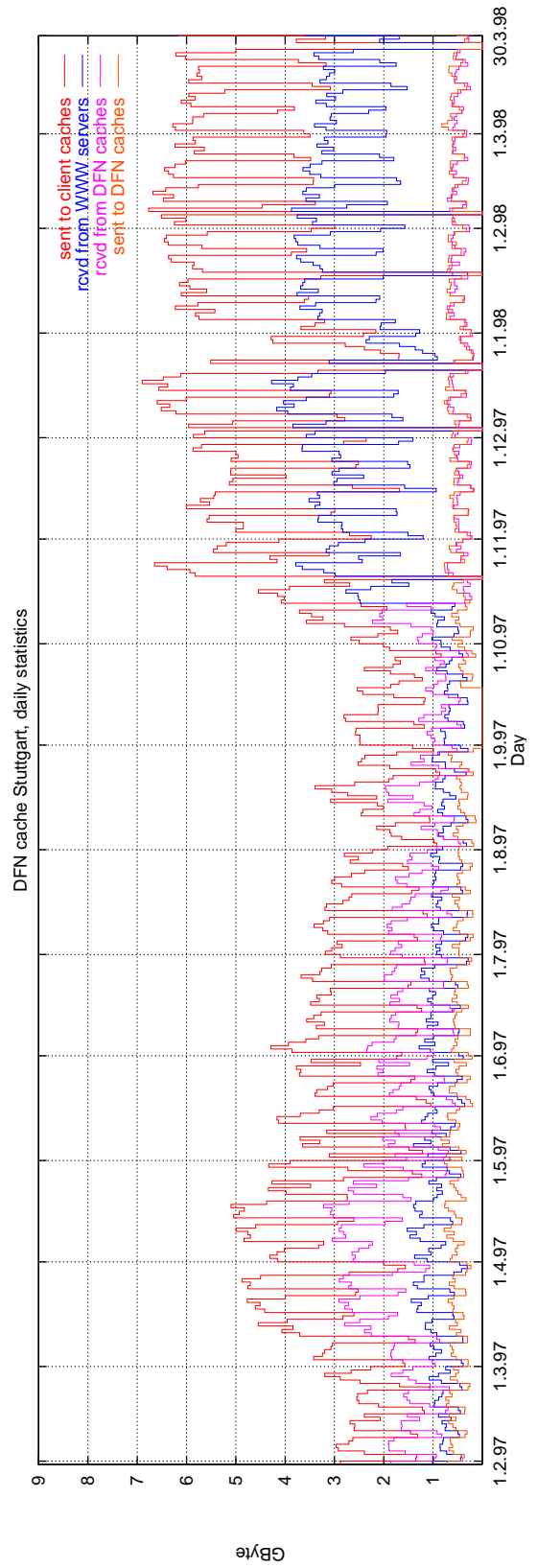
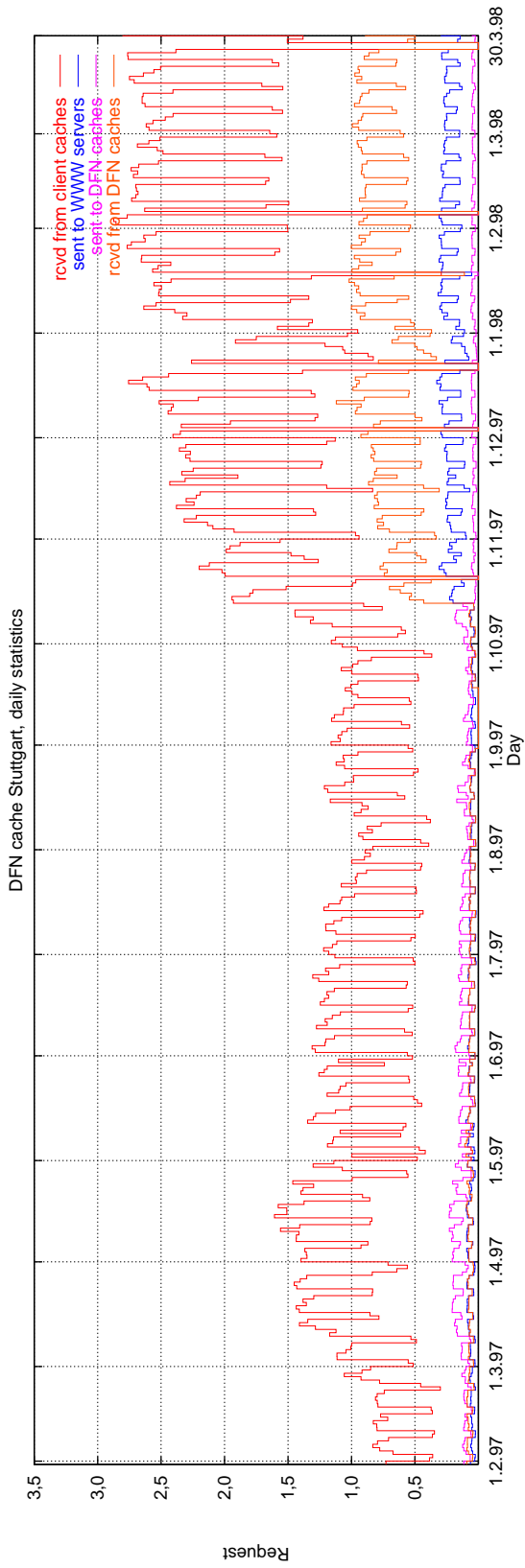
### A.1.8 Verkehrsbilanz München, Requests und Datenvolumen



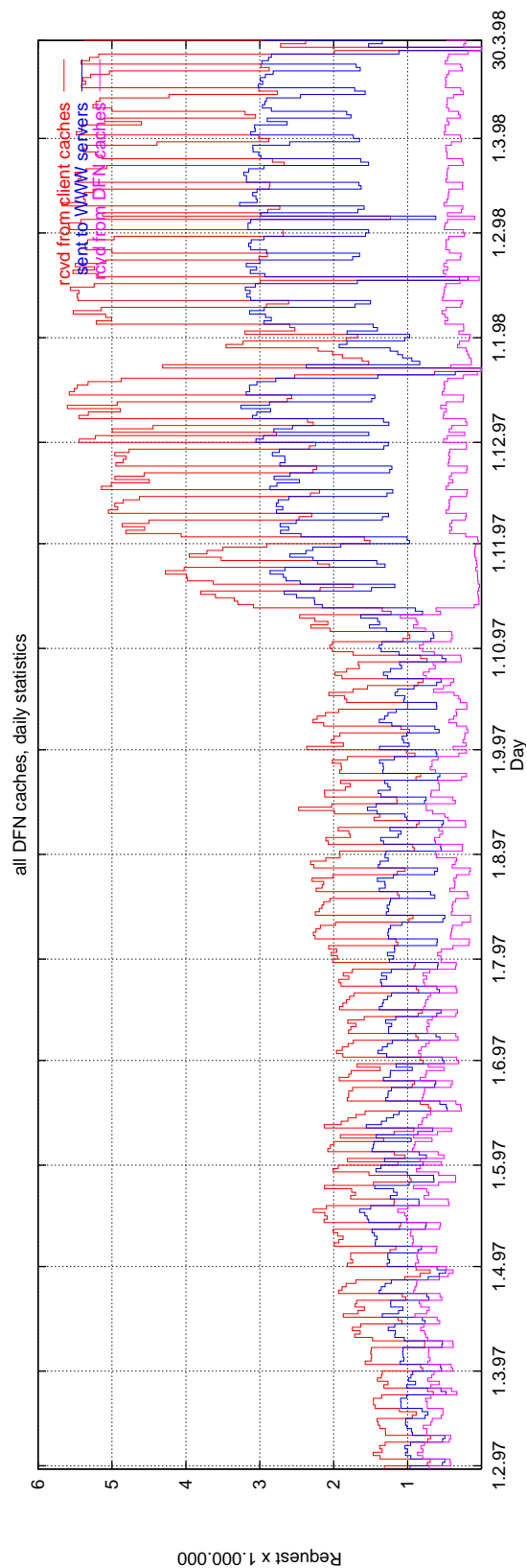
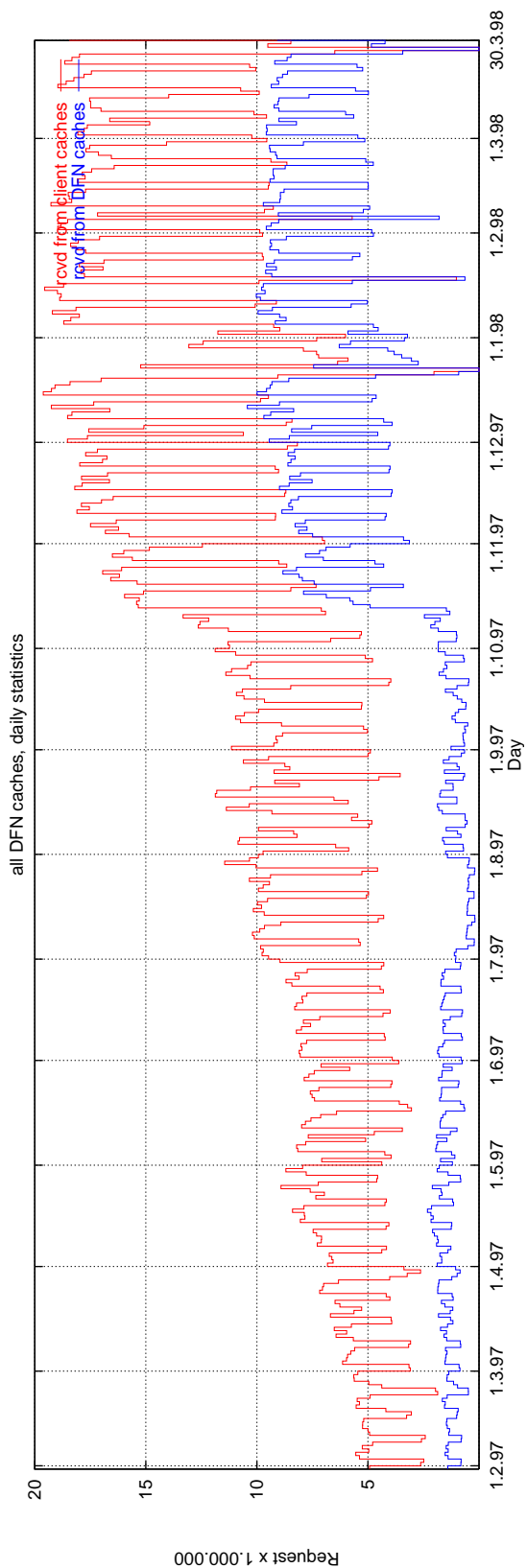
### A.1.9 Verkehrsbilanz Nürnberg, Requests und Datenvolumen



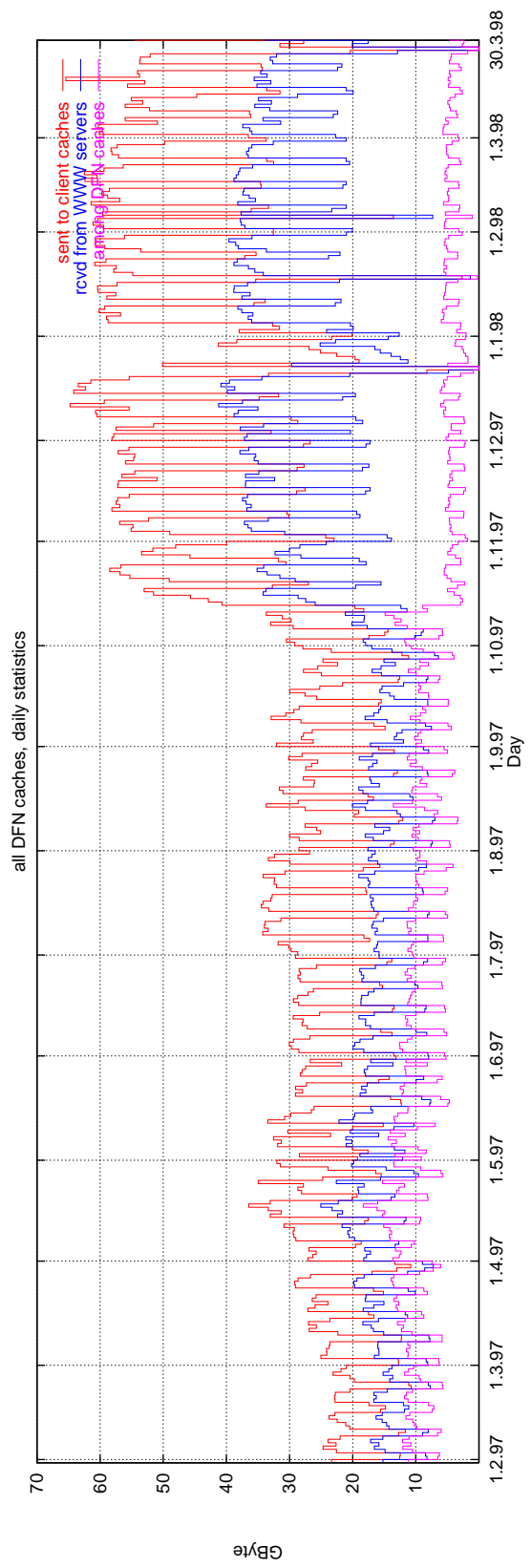
### A.1.10 Verkehrsbilanz Stuttgart, Requests und Datenvolumen



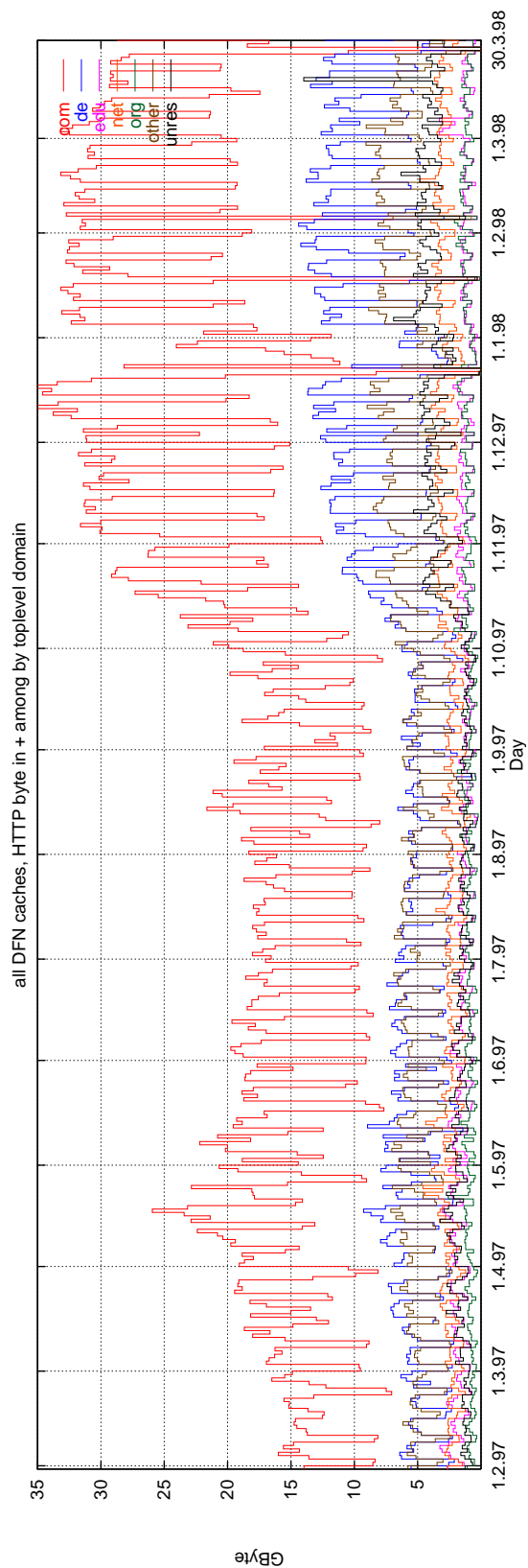
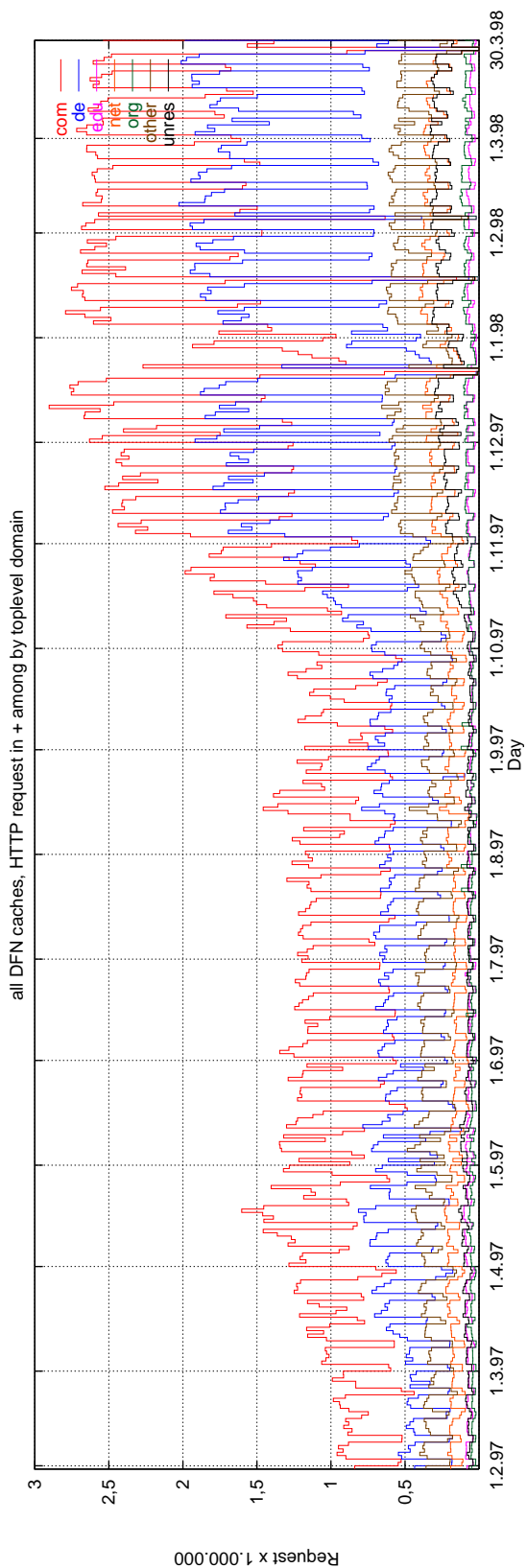
### A.1.11 Verkehrsbilanz Gesamt, ICP- und HTTP-Requests



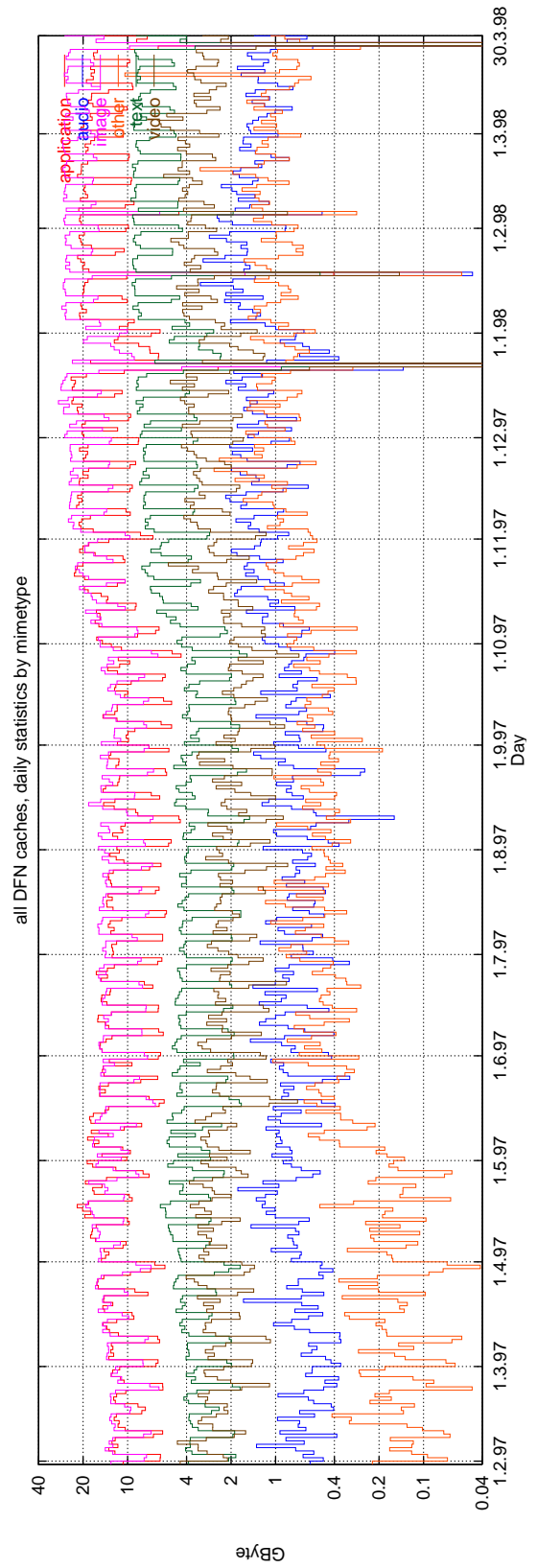
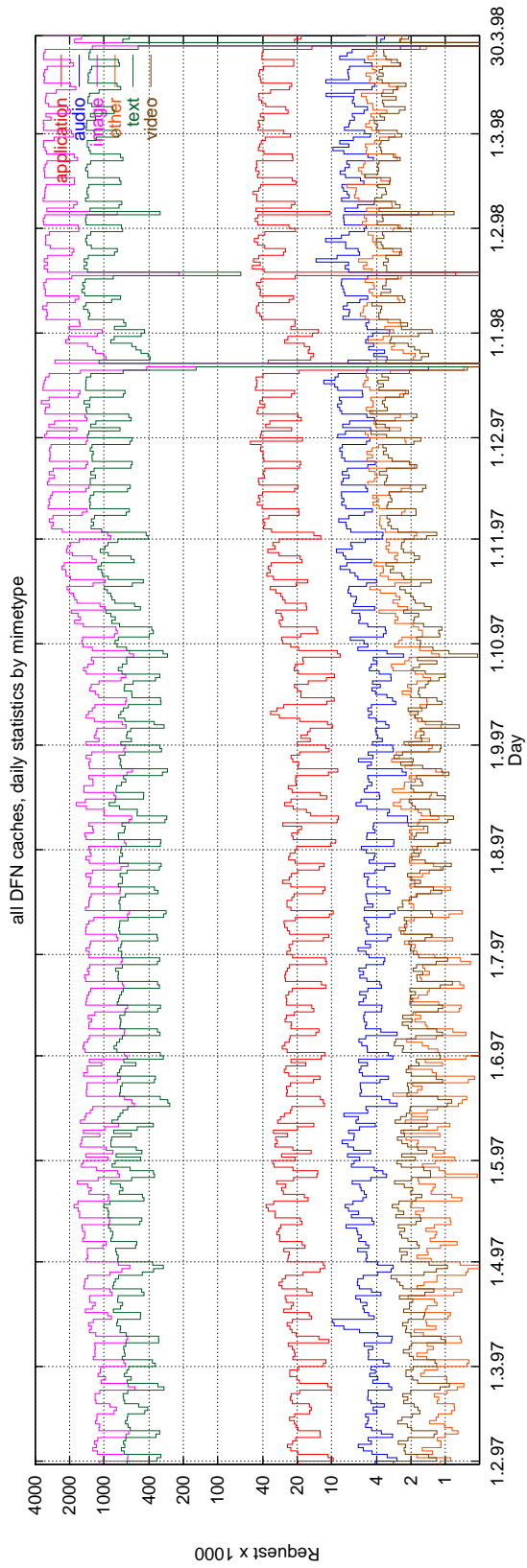
### A.1.12 Verkehrsbilanz Gesamt, , Datenvolumen



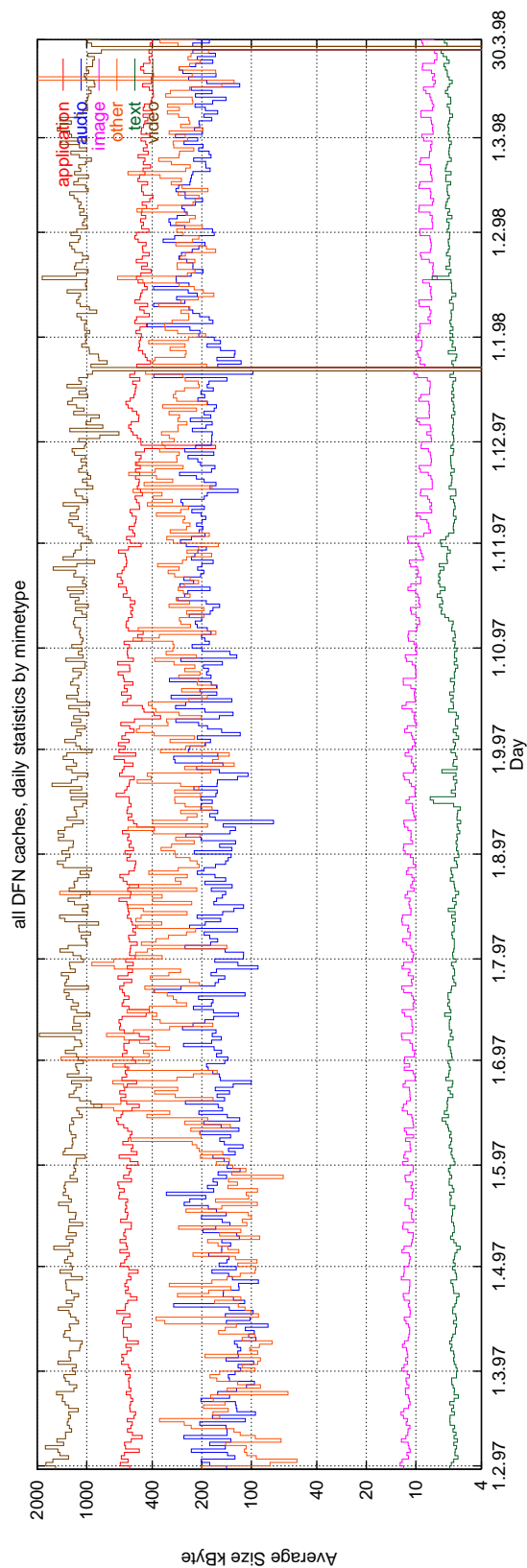
### A.1.13 Toplevel Domains Gesamt, Requests und Datenvolumen



### A.1.14 Mimetypes Gesamt, Requests und Datenvolumen



### A.1.15 Mimetypes Gesamt, Durchschnittliche Objektgröße



## A.2 Beispiele für die Nutzung von webtee

### A.2.1 Vergleich HTTP-Requests unterschiedlicher WWW-Browser

In dem folgenden Beispiel ist webtee zwischen WWW-Client und WWW-Server installiert, WWW-Caches sind nicht in die Übertragung der WWW-Objekte einbezogen (vgl. Kapitel 4.6.2, Abbildung 8). Dargestellt werden hier die HTTP-Requests, die von 5 unterschiedlichen WWW-Browsern für den Zugriff auf die Seite <http://www.rvs.uni-hannover.de/index.html> erzeugt werden. Auffällig ist der erhebliche Umfang der Header-Informationen, der von dem WWW-Browser Mosaic mit *jedem* HTTP-Requests generiert wird. Die anderen untersuchten WWW-Browser Netscape 3.04 und 4.05, Internet Explorer 4.0 sowie Lynx 2.7.2 unterscheiden sich nur unwesentlich in ihren deutlich knapperen HTTP-Requests.

#### 1. Mosaic 2.7b5 unter IRIX 6.2

```
> GET /index.html HTTP/1.0
> Accept: image/x-pcd-jpeg, text/plain, application/x-html, application/html,
text/x-html, text/html, application/postscript, audio/x-audio-on-demand,
application/pdf, application/x-frameset, audio/x-pn-realaudio
> Accept: application/x-ns-proxy-autoconfig, application/x-docshow-vr,
audio/basic, audio/x-aiff, image/gif, image/jpeg, image/png, image/x-png,
image/tiff, image/x-portable-anymap, image/x-portable-bitmap
> Accept: image/x-portable-graymap, image/x-portable-pixmap, image/x-rgb,
image/rgb, image/x-xbitmap, image/x-ypixmap, image/xwd, image/x-xwd,
image/x-xwindowdump, video/mpeg, video/quicktime, video/x-sgi-movie
> Accept: application/postscript, application/x-dvi, message/rfc822,
application/x-latex, application/x-tex, application/x-texinfo,
application/x-troff, application/x-troff-man, application/x-troff-me,
application/x-troff-ms
> Accept: text/richtext, text/tab-separated-values, text/x-setext, */*
> User-Agent: NCSA_Mosaic/2.7b5 (X11;IRIX 6.2 IP22) libwww/2.12 modified
> Host: www.rvs.uni-hannover.de
> Extension: Notify-Domain-Restriction
```

#### 2. Netscape 3.04 unter IRIX 6.2

```
> GET index.html HTTP/1.0
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/3.04 (X11; I; IRIX 6.2 IP22)
> Host: www.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
```

#### 3. Netscape 4.05 unter IRIX 6.2

```
> GET /index.html HTTP/1.0
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.05 [en] (X11; I; IRIX 6.2 IP22)
> Host: www.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
```

#### 4. Internet Explorer 4.0 unter Windows 95

```
> GET / HTTP/1.0
> Host: leonore.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/xbm, image*
> Accept-Language: de
```

```
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.0 (compatible; MSIE 4.0; Mac_PowerPC)
> UA-OS: MacOS
> UA-CPU: PPC
> Extension: Security/Remote-Passphrase
```

## 5. Lynx 2.7.2 unter Linux

```
> GET /index.html HTTP/1.0
> Host: www.rvs.uni-hannover.de
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.05 [en] (X11; I; IRIX 6.2 IP22)
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
```

In allen fünf Fällen wurde von dem adressierten WWW-Server `www.rvs.uni-hannover.de` ein HTTP-Reply mit folgenden Header-Informationen generiert:

```
< HTTP/1.1 200 OK
< Date: Fri, 28 Aug 1998 12:56:45 GMT
< Server: Apache/1.2.5 PHP/FI-2.0b13
< Last-Modified: Thu, 13 Aug 1998 14:04:51 GMT
< ETag: "82d-1979-35d2f283"
< Content-Length: 6521
< Accept-Ranges: bytes
< Connection: close
< Content-Type: text/html
```

## A.2.2 Reload-Mechanismus in Netscape 4.05 unter IRIX 6.2

In diesem Beispiel werden die unterschiedlichen HTTP-Header dargestellt, die von Netscape für das wiederholte Laden (Reload) desselben WWW-Objekts generiert werden. Zunächst wird der im vorangegangene Beispiel aufgeführte HTTP-Header für das erste Laden des WWW-Objektes dargestellt.

### 1. Erster Request auf das Objekt `http://www.rvs.uni-hannover.de/index.html`

```
> GET /index.html HTTP/1.0
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.05 [en] (X11; I; IRIX 6.2 IP22)
> Host: www.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
```

### 2. Erneuter Request durch wiederholte Eingabe des URL

```
> GET /index.html HTTP/1.0
> If-Modified-Since: Thursday, 13-Aug-98 14:04:51 GMT; length=6521
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.05 [en] (X11; I; IRIX 6.2 IP22)
> Host: www.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
```

### 3. Erneuter Request durch Betätigung des Reload-Button

```
> GET /index.html HTTP/1.0
> If-Modified-Since: Thursday, 13-Aug-98 14:04:51 GMT; length=6521
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.05 [en] (X11; I; IRIX 6.2 IP22)
> Pragma: no-cache
> Host: www.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
```

#### 4. Erneuter Request durch Betätigung von Shift + Reload-Button

```
> GET /index.html HTTP/1.0
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.05 [en] (X11; I; IRIX 6.2 IP22)
> Pragma: no-cache
> Host: www.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
```

Zu beachten sind die unterschiedlichen Kombinationen von `If-Modified-Since:` und `Pragma: no-cache` der HTTP-Header in den Fällen 2. bis 4. Weiterhin ist zu bemerken, daß in Fall 2. der WWW-Browser einen Request an den WWW-Cache bzw. WWW-Server sendet, da die Einstellung der Option "Bearbeiten->Einstellungen->Erweitert->Cache->Vergleich zwischen Online..." auf "Jedesmal" eingestellt war. Bei Angabe der Optionen "Einmal pro Sitzung" oder "Nie" in diesem Auswahlfeld wäre eine erneute Verbindungsaufnahme mit dem WWW-Server ausgeblieben. Stattdessen hätte Netscape das WWW-Objekt aus dem lokalen Platten- oder Memory-Cache ausgelesen und dargestellt.

In den Fällen 2. und 3. erfolgt die Beantwortung des HTTP-Request abhängig von der Auswertung der Angabe in der Zeile `If-Modified-Since:`. Wurde das WWW-Objekt verändert, wird es durch WWW-Cache bzw. WWW-Server ausgeliefert. Ist keine Veränderung seit dem mit `If-Modified-Since:` angegebenen Datum festgestellt worden, besteht die Antwort aus der Meldung `Not Modified`:

```
< HTTP/1.1 304 Not Modified
< Date: Fri, 28 Aug 1998 14:20:13 GMT
< Server: Apache/1.2.5 PHP/FI-2.0b13
< Connection: close
< ETag: "82d-1979-35d2f283"
```

In Fall 4. wird das WWW-Objekt in jedem Fall von dem WWW-Server übertragen. Auch ein zwischengeschalteter WWW-Cache sollte - verantwortungsbewußte Konfiguration vorausgesetzt - das Objekt nicht aus seinem Speicher ausliefern, sondern den HTTP-Request an den WWW-Server weiterleiten. Somit besteht in der Regel auch bei dem Einsatz von WWW-Caches die Möglichkeit, das Originaldokument von dem WWW-Server zu erhalten.

### A.2.3 Veränderung der HTTP-Header von WWW-Caches

In diesem Beispiel wird die Veränderung der HTTP-Header beim Durchlaufen einer Cache-Kaskade erläutert. Die Informationen sind insbesondere für die Betreiber von WWW-Servern wichtig, da in der Regel die Zugriffsstatistiken anhand dieser Informationen generiert werden. Das zugrundeliegende Szenario ist in Kapitel 4.6.2, Abbildung 8 dargestellt. Zwischen WWW-Client und WWW-Server sind zwei WWW-Caches geschaltet.

#### 1. Request von WWW-Client an WWW-Cache

```
> GET http://leonore.rvs.uni-hannover.de/ HTTP/1.0
> Proxy-Connection: Keep-Alive
> User-Agent: Mozilla/4.05 [en] (X11; I; Linux 2.0.33 i586)
> Host: leonore.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
```

## 2. Request von WWW-Cache an WWW-Cache

```
> GET http://leonore.rvs.uni-hannover.de/ HTTP/1.0
> User-Agent: Mozilla/4.05 [en] (X11; I; Linux 2.0.33 i586)
> Host: leonore.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
> Via: 1.0 lotta.rvs.uni-hannover.de:8080 (Squid/1.1.22)
> X-Forwarded-For: 192.168.0.3
> Cache-control: Max-age=259200
```

## 3. Request von WWW-Cache an WWW-Server

```
> GET /index.html HTTP/1.0
> Referer: http://leonore.rvs.uni-hannover.de/
> User-Agent: Mozilla/4.05 [en] (X11; I; Linux 2.0.33 i586)
> Host: leonore.rvs.uni-hannover.de
> Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png
> Accept-Language: en
> Accept-Charset: iso-8859-1,*,utf-8
> Cache-control: Max-age=259200
> Via: 1.0 www-cache.rrzn.uni-hannover.de:8080 (Squid/1.1.22), 1.0 koeln.www-
cache.dfn.de:8080 (Squid/1.1.22)
> X-Forwarded-For: 192.168.0.3, 192.168.0.3
```

Zu sehen sind in diesem Beispiel die Erweiterungen des HTTP-Header in den WWW-Caches durch die Einträge `Via:`, `X-Forwarded-For:` und `Cache-Control:`. Die HTTP-Reply hingegen wird in ihrem Umfang durch die WWW-Caches reduziert:

### 1. Reply von WWW-Server an WWW-Cache

```
< HTTP/1.1 200 OK
< Date: Sat, 29 Aug 1998 18:59:59 GMT
< Server: Apache/1.3.0 (Unix) PHP/3.0
< Last-Modified: Thu, 22 Feb 1996 11:45:54 GMT
< ETag: "177ab-94-312c5772"
< Accept-Ranges: bytes
< Content-Length: 148
< Connection: close
< Content-Type: image/gif
```

### 2. Reply von WWW-Cache an WWW-Cache

```
< HTTP/1.1 200 OK
< Date: Sat, 29 Aug 1998 18:59:59 GMT
< Server: Apache/1.3.0 (Unix) PHP/3.0
< Connection: close
```

```
< Content-Type: text/html
```

### 3. Reply von WWW-Cache an WWW-Client

```
< HTTP/1.1 200 OK  
< Date: Sat, 29 Aug 1998 18:59:59 GMT  
< Server: Apache/1.3.0 (Unix) PHP/3.0  
< Connection: close  
< Content-Type: text/html
```