

# Grundlagen und Probleme beim Nachweis von Selbstähnlichkeit und Long Range Dependance für Internetdatenverkehr

Johannes Formann  
<johannes@formann.de>

# Inhaltsverzeichnis

<b>I</b>	<b>Einleitung</b>	<b>1</b>
1	Ziele und Aufbau der Arbeit	1
<b>II</b>	<b>Grundlagen</b>	<b>1</b>
<b>2</b>	<b>Modellierungsebenen</b>	<b>1</b>
2.1	Paketebene . . . . .	1
2.2	Anwendungsebene . . . . .	2
<b>3</b>	<b>klassische Modellierungsansätze</b>	<b>2</b>
3.1	Probleme mit der Modellierung . . . . .	2
3.2	Auswirkung auf die Modellierung . . . . .	3
<b>4</b>	<b>Eigenschaften alternativer Verteilungsfunktionen</b>	<b>3</b>
4.1	Long Tail / Heavy Tail . . . . .	3
4.2	Selbstähnlichkeit . . . . .	4
4.3	Long Range Dependence . . . . .	5
4.3.1	Hurst-Parameter . . . . .	5
4.4	Verbindungen zwischen den Eigenschaften . . . . .	6
<b>5</b>	<b>Nachweis der Eigenschaften</b>	<b>6</b>
5.1	Long Tail / Heavy Tail . . . . .	6
5.2	Selbstähnlichkeit . . . . .	7
5.3	Long Range Dependance . . . . .	7
5.3.1	Hurst-Parameter Schätzer . . . . .	7
5.3.2	Probleme beim Schätzen des Hurst-Parameters . . . . .	9
<b>III</b>	<b>Experimentelles Nachweisen</b>	<b>10</b>
<b>6</b>	<b>Tools und Traces</b>	<b>10</b>
6.1	Trace . . . . .	10
6.2	Tools zum Schätzen des Hurst-Parameters . . . . .	11
<b>7</b>	<b>Vorgehen</b>	<b>12</b>
7.1	Datenaufbereitung & Verdichtung . . . . .	12
7.2	Analyse . . . . .	13
<b>8</b>	<b>Auswertung</b>	<b>13</b>
8.1	Anfragen pro Sekunde am 10. Mai 1998 5-21 Uhr . . . . .	13
8.2	Byte pro Sekunde am 10. Mai 1998 5-21 Uhr . . . . .	14

8.3	Anfragen pro Sekunde am 30 Juni 8-24 Uhr . . . . .	16
<b>IV</b>	<b>Zusammenfassung</b>	<b>17</b>
<b>9</b>	<b>Zusammenfassung</b>	<b>17</b>
9.1	Bedeutung für die Praxis . . . . .	17
9.2	Anmerkungen . . . . .	18
<b>V</b>	<b>Anhang</b>	<b>19</b>
<b>10</b>	<b>Anhaenge</b>	<b>19</b>
10.1	Datenformat der Worlc Cup 1998 Traces . . . . .	19
10.2	awk-Skript für Daten pro Sekunde . . . . .	20
10.3	Abbildungsverzeichnis . . . . .	21
10.4	Tabellenverzeichnis . . . . .	21
	<b>Literatur</b>	<b>22</b>

# Teil I

## Einleitung

### 1 Ziele und Aufbau der Arbeit

Diese Arbeit soll ein Überblick über die theoretischen Grundlagen zu den aktuell in der Netzwerkmodellierung verwendeten Modelle geben (Selbstähnlichkeit, Long Range Dependence, Long Tail), Analysemethoden vorstellen, und am Beispiel der Long Range Dependence-Eigenschaft exemplarisch untersuchen.

Das Verständniss und Kenntnisse der Modellierungstechniken sind vor allen zur Kapazitätsplanung wichtig.

Die Arbeit lässt sich in drei wesentliche Teile gliedern.

Im Grundlagen-Teil (Teil II) werden zunächst die verschiedenen Modellierungsansätze vorgestellt, und die Eigenschaften der dabei genutzten Funktionen. Im 5. Kapitel werden Methoden erklärt, mit denen die Eigenschaften<sup>1</sup> in vorhandenen Daten nachgewiesen werden können.

Der Praxis Teil (Teil III) verdeutlicht die gewonnenen Erkenntnisse an einem Praxisbeispiel, und zeigt die Probleme dabei auf.

Im letzten Teil werden die Ergebnisse dann zusammengefasst.

## Teil II

### Grundlagen

### 2 Modellierungsebenen

Bei der Modellierung von Internetdatenverkehr bietet sich eine Unterscheidung zwischen Paket- und Anwendungsebene an.

#### 2.1 Paketebene

Auf der Paketebene versucht man die Ankunft von Paketen zu modellieren.

Das kann sowohl allgemein für eine Verbindung geschehen (z.B. für die Dimensionierung von Puffern in Routern), oder auch Dienstspezifisch.

Die zu modellierenden Parameter sind dabei die Verteilungen der Zwischenankunftszeiten und die der Paketgrößen.

---

<sup>1</sup>Heavy-Tailed, Selbstähnlichkeit, Long Range Dependence

## 2.2 Anwendungsebene

Auf der Anwendungsebene gibt es deutlich mehr Parameter, die man je nach zu untersuchender Anwendung, modellieren kann.

Beispiele:

- Verbindungsankünfte und/oder Sitzungsankünfte<sup>2</sup>
- Verbindungsdauer und/oder Sitzungsdauern
- Ressourcenbedarf einer Verbindung (Übertragenes Datenvolumen, CPU-Zeit)
- Applikationsspezifische Werte (z.B. Dateigrößen)

In dieser Arbeit wird der Fokus auf der Anwendungsebene liegen, aber die Techniken lassen sich ohne Unterschied auf beide Bereiche anwenden.

## 3 klassische Modellierungsansätze

Bis etwa 1995[16, 20] wurde bei der Modellierung von Datenverkehr im Internet Poisson-Prozesse als adäquate Annäherung an echte Systeme gesehen.

Poisson-Prozesse sind gedächtnislos, und die Zwischenankunftszeiten entsprechen einer Exponentialverteilung.

Diese wurden genutzt, da die Poissonverteilung angenehme mathematische Eigenschaften hat<sup>3</sup> und Erfahrung vorhanden waren[5] (wurden bereits zur Dimensionierung von Telefonsystemen genutzt).

### 3.1 Probleme mit der Modellierung

Im Jahr 1995 wurden von Paxson und Floyd ein Paper[20] veröffentlicht, welches die Poisson-Modellierung als Modellierungsansatz für viele Anwendungsfälle als ungeeignet bezeichnet. Ihre Aussage beruhte auf der Analyse mehrerer Traces<sup>4</sup>, bei der sie u.A. die folgenden Beobachtungen machten:

- Benutzeraktivität ist weiterhin gut als Poisson-Prozess mit konstanter Rate modellierbar. (In den Traces Verbindungswischenankunftszeiten von Telnet und FTP-Session)  
Es ist aber gängige Praxis, längere Zeiträume, bei entsprechend sich verändernden Verhalten der Benutzer<sup>5</sup>, in verschiedene Abschnitte mit jeweils eigener Rate zu unterteilen.

---

<sup>2</sup>Eine Verbindung bezeichnet immer eine TCP-Verbindung, während eine Sitzung aus mehreren logisch zusammen gehörenden Verbindungen bestehen kann, und einer Benutzeraktivität zugeordnet wird.

<sup>3</sup>die gewünschten Leistungsmaße lassen sich leicht berechnen, auch bei gemultiplexten Systemen

<sup>4</sup>Aufzeichnungen von IP-Headern samt genauen Zeitstempel

<sup>5</sup>z.B. außerhalb der Arbeitszeit, Kernarbeitszeit, Mittagspause o.Ä)

- Vom Benutzer während seiner Aktivitätsphase ausgelösten Verbindungen/Datenverkehr (FTP-Data, Ankunft von Telnet-Paketen) sowie Datenverkehr der stark automatisiert abläuft (SMTP, NTP) weichen stark von Poisson modellierten Verhalten ab.

Sie stellten in Folge fest, dass sich der Datenverkehr besser mit Verteilungsfunktionen modellieren lässt, welche die später erläuterten Eigenschaften „Long Tail“, Selbstähnlichkeit und „Long Range Dependence“ (LRD) besitzen.

Diese Beobachtungen wurden später von anderen Autoren auch an anderen Datenquellen bestätigt [8], so dass zum aktuellen Zeitpunkt Konsens besteht, dass die Poissonmodellierten Ankünfte im Regelfall nicht geeignet sind, Internetdatenverkehr realistisch zu modellieren<sup>6</sup>.

### 3.2 Auswirkung auf die Modellierung

Dieser Paradigmenwechsel hat zwei Hauptkonsequenzen:

- Für den Modellierer bedeutet der Wechsel auf Verteilungsfunktionen mit den oben genannten Eigenschaften<sup>7</sup> einen deutlich erhöhten Aufwand bei der Berechnung von Werten, da die dazu notwendigen Funktionen deutlich komplizierter werden.
- Für den Betreiber bedeutet der Wechsel, das mit deutlich größeren und längeren Spitzen im Verkehrsmuster gerechnet werden muss, also die Puffer oder die Reservekapaazität deutlich erhöht werden muss. Dies wird in Abbildung 1 deutlich, bei dem die notwendige Puffergröße gegen die Auslastung bei konstanter Verlustrate dargestellt wird.

## 4 Eigenschaften alternativer Verteilungsfunktionen

### 4.1 Long Tail / Heavy Tail

Von „Long Tail“ oder auch „Heavy Tail“ wird bei Verteilungsfunktionen gesprochen, wenn auch sehr große Werte nicht zu vernachlässigende Wahrscheinlichkeiten haben. Davon spricht man, wenn die Verteilungsfunktion langsamer als exponentiell abnimmt.

Formal eine Zufallsvariable  $X$  ist heavy tailed wenn für alle  $\epsilon > 0$  gilt[5]:

$$\mathbb{P}[X > x]e^{\epsilon x} \rightarrow \infty, \text{ für } x \rightarrow \infty$$

Eine alternative Darstellung[4, 9] lautet: Eine Zufallsvariable  $X$  ist heavy tailed, wenn für große  $x$  gilt:

$$P(X > x) \sim cx^{-a}, \text{ für } x \rightarrow \infty$$

Dabei ist  $c$  ein Konstante und  $a$  der die Form beeinflussende Parameter. Für  $0 < a \leq 2$  ist die Long Tail Eigenschaft gegeben, welche zur Folge hat, das die Varianz unendlich ist.

<sup>6</sup>Es gibt allerdings noch Diskussionen ob Long Range Dependence der richtige Ansatz ist.

<sup>7</sup>z.B. Pareto

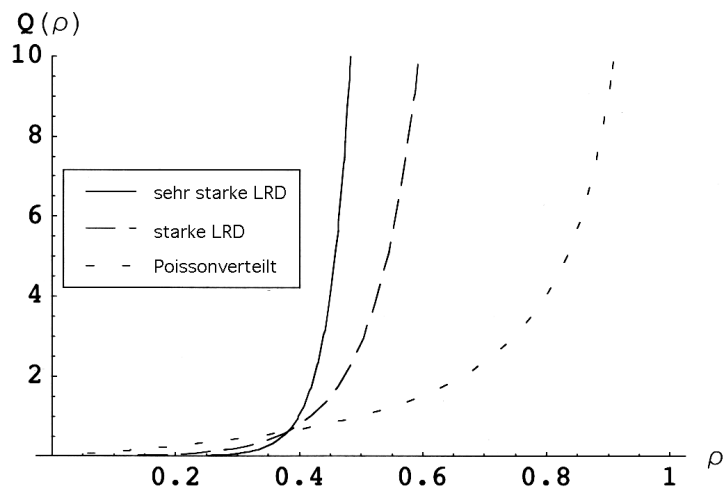


Abbildung 1: Benötigte Puffergröße Poisson vs. LRD. Aus [10]

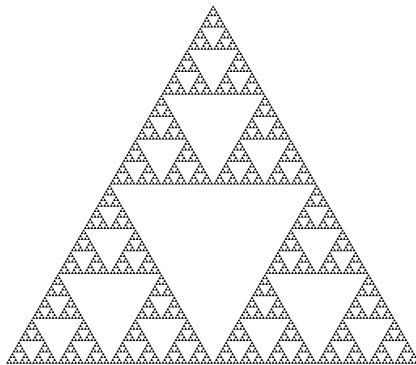


Abbildung 2: Sierpinski Dreieck (Quelle: Wikipedia Commons)

## 4.2 Selbstähnlichkeit

Selbstähnlichkeit bezeichnet im eigentlichen Sinne eine Sorte von Fraktalen, die ihr Aussehen nicht verändern, unabhängig von der Vergrößerung[8] wie das Sierpinski-Dreieck (siehe Abbildung 2).

Bei der Betrachtung von stochastischen zeitabhängigen Daten nutzt man die folgende Definition[5]:

Sei  $Y_t$  ein stochastischer Prozess mit dem kontinuierlichen Zeitparameter  $t$ , dann ist der Prozess selbstähnlich mit dem Selbstähnlichkeitsparameter  $H$ , falls für jede beliebige positive Konstante  $c$  gilt: Der skalierte Prozess  $c^{-H}Y_{ct}$  ist von der Verteilung gleich zu dem ursprünglichen Prozess  $Y_t$ .

Diese Definition kann man sich bildlich vorstellen, das der Prozess „gleich aussieht“, wenn man sowohl die x-Achse (Zeit) als auch die y-Achse passend skaliert. Ein Beispiel

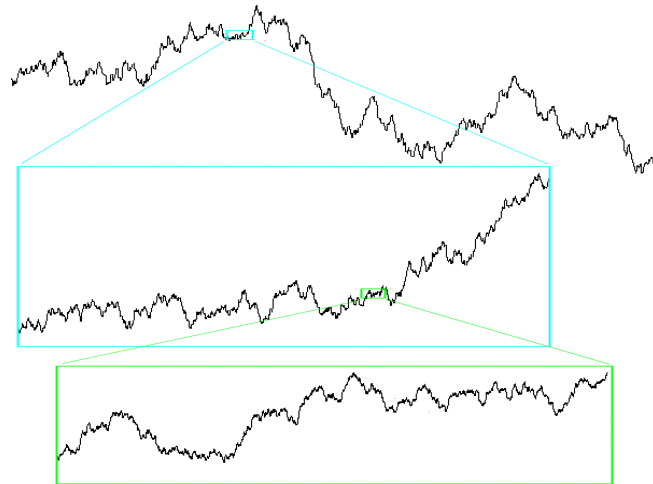


Abbildung 3: Selbstähnliche Verkehrskurve aus [11].

Ausschnitte aus der jeweils darüber liegenden Kurve haben jeweils ähnliches Verhalten.

ist in Abbildung 3 zu sehen.

### 4.3 Long Range Dependence

Die Long Range Dependence-Eigenschaft sagt aus, dass ein Prozess gedächtnisbehaftet ist, und damit über lange Zeiträume Korrelationen aufweist.

Formal gibt es verschiedene Definitionen:

- Ein stationärer Prozess ist Long Range Dependend, wenn seine Autokorrelationsfunktion  $\rho(k)$  nicht aufsummierbar ist. [20, 5] D.h.  $\sum_{k=0}^{\infty} \rho(k) = \infty$ .
- Ein stationärer Prozess  $X$  ist Long Range Dependend, wenn es eine Zahl  $\alpha \in (0, 1)$  und eine Konstante  $c_p > 0$  gibt, so dass  $\lim_{k \rightarrow \infty} \rho(k) / [c_p k^{-\alpha}] = 1$ . [13]
- Ein stationärer Prozess ist Long Range Dependend mit dem Grad  $0 < \alpha < 1$  falls es eine Konstante  $c_1$  gibt, so dass seine Autokovarianzfunktion  $\gamma_k \sim \frac{c_1}{k^\alpha}$ . [4]

Die Letzten beiden Definitionen sind fast identisch, und haben den Vorteil, dass sie eine quantitative Aussage über die Long Range Dependence Eigenschaft treffen mit dem Parameter  $\alpha$ .

Im weiteren Verlauf wird der Hurst-Parameter ( $H$ ) genutzt, statt  $\alpha$ , wobei gilt:  $H = 1 - \frac{\alpha}{2}$

#### 4.3.1 Hurst-Parameter

Der Hurst-Parameter ( $H$ ) wurde nach Harold Edwin Hurst benannt, der als Hydrologe den Wert entwickelte, um die optimale Dammgröße für die stark schwankenden Pegel des

Nils zu finden.

Definition:  $H := \frac{\log(R/S)}{\log T}$  wobei RS den Wert Rescaled Range des Prozesses entspricht, und T der Samplegröße. (Siehe auch 5.3.1)

Dabei gilt für das Verhalten eines Prozesses über die Zeit:

$0 \leq H < 0,5$ : Der Prozess bleibt stärker in der Nähe des Mittelwertes als bei einem „random walk“.

$H = 0,5$ : das Verhalten entspricht eines „random walks“. Wird auch mit Verteilungen wie der Poissonverteilung erreicht.

$0,5 < H \leq 1$ : das Verhalten legt teilweise größere „Strecken“ als bei einem „random walk“ zurück, und zeigt damit Long Range Dependence Eigenschaften.

$H > 1$ : Nicht definiert.

#### 4.4 Verbindungen zwischen den Eigenschaften

Unter der Annahme der sogenannten „weak Self Similarity“ sind selbstähnliche Prozesse und Long Range Dependend Prozesse äquivalent[10].

Im Allgemeinen kann man keinen Zusammenhang feststellen, da es sowohl Long Range Dependend Prozesse gibt die nicht selbstähnlich sind<sup>8</sup>[4] als auch selbstähnliche Prozesse<sup>9</sup> die keine Long Range Dependence besitzen.

Bei einem Long Range der durch das Multiplexen von On-Off Quellen mit jeweils konstanter Datenrate modelliert wird sind die Zufallsvariablen der Einschalt Dauern Heavy tailed, und andersherum[4].

## 5 Nachweis der Eigenschaften

### 5.1 Long Tail / Heavy Tail

Zur Bestimmung von Heavy Tail Eigenschaften wurden eine ganze Reihe von Verfahren entwickelt, die im folgenden kurz erwähnt werden sollen.

**Log-Log Diagram / ccdf-Test** In einem Diagramm werden auf die x-Achse, logarithmisch skaliert, der Wertebereich aufgetragen, und auf der y-Achse die komplementär-funktion der Verteilungsfunktion.[7, 4, 9]

In Abbildung 4 ist ein Beispiel zu sehen.

In dem Diagramm wird dann versucht ein möglichst kleines  $x_0$  zu finden, ab dem die Kurve wie einer Gerade aussieht. Aus der Steigung der Geraden lässt sich dann der Parameter  $\alpha$  schätzen.  $\frac{\Delta \log \bar{F}(x)}{\Delta \log x} \sim -\alpha$

---

<sup>8</sup>z.B. erzeugt vom Generator „fractionally integrated noise“

<sup>9</sup>z.B. der Short Range Dependend Prozess des „random walks“

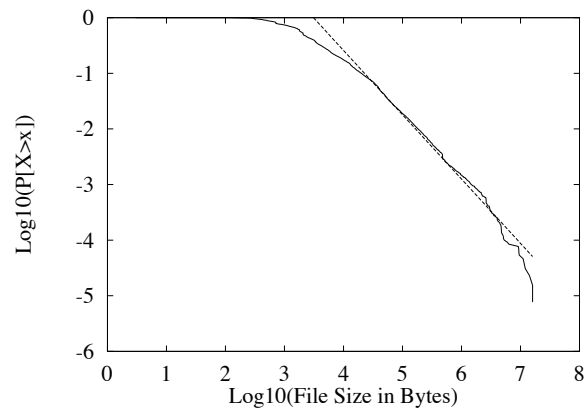


Abbildung 4: Beispiel für Größe von Dateien die über ein Netz übertragen wurden.[7]

**Hill Plot und Hill Schätzer** Bei den Hill Plot und dem Hill Schätzer wird eine Pareto-Verteilung angenommen, und die Long-Tail Eigenschaften indirekt über die geeignete Parametrisierung der Pareto-Verteilung bestimmt.

Siehe auch Kapitel 5.3.1.

**aest-Tool** Das von M. Crovella und M. Taquu entwickelte Tool aest[7] wertet den Kurvenverlauf bei unterschiedlichen Aggregationsgrößen aus.

Aus den Unterschieden der Kurven und dem Kurvenverlauf werden dann mehrere  $\alpha_i$  geschätzt, deren Mittelwert als geschätztes  $\alpha$  ausgegeben wird.

## 5.2 Selbstähnlichkeit

Selbstähnlichkeit wird nachgewiesen, indem der Hurst-Parameter  $H$  bestimmt wird. Wenn dieser sich in dem Bereich  $0,5 \leq H \leq 1$  befindet, so wird Selbstähnlichkeit angenommen.[8, 16]

Zur Bestimmung des Hurst-Parameters siehe Kapitel 5.3.1.

## 5.3 Long Range Dependence

Long Range Dependence wird auch über die Bestimmung des Hurst-Parameters nachgewiesen und dabei gleichzeitig quantifiziert.[16, 6, 4]

Damit Long Range Dependence vorliegt, muss der Hurst-Parameter  $H$  die folgende Bedingung erfüllen:  $0,5 < H \leq 1$

### 5.3.1 Hurst-Parameter Schätzer

Da sowohl die Long Range Dependence als auch die Selbstähnlichkeit über asymptotische Definitionen verfügen, also für sichere Aussagen unendlich lange Datenreihen benötigen,

können die Parameter in der Praxis nur geschätzt werden.

Im folgenden werden die gebräuchlichsten Hurst-Parameter Schätzer jeweils kurz vorgestellt. Bei denen mit einem einfachen mathematischen Hintergrund wird dabei das Funktionsprinzip näher erläutert, bei den Restlichen würde das den Umfang dieser Arbeit sprengen, und es wird auf die weiterführende Literatur verwiesen.

**Absolute Moments / Absolute Value** Dieser Schätzer gehört zu den grafischen zeitbasierten Schätzern.

Er basiert darauf für verschiedene stark aggregierte (Blockgröße  $m$ ) Varianten der zu untersuchenden Zeitserie das jeweilige erste absolute Moment zu berechnen, und dann beides logarithmisch skaliert in einem Graphen darzustellen. (X-Achse  $m$ , Y-Achse erstes absolutes Moment)[24, 15, 25]

Berechnung:

$$AM^{(m)} = \frac{1}{N/m} \sum_{k=1}^{N/m} |X^{(m)}(k) - \bar{X}|$$

Mit  $X^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} x_i$ ,  $k = 1, 2, \dots, [N/m]$ ,  $N$  als Länge der Serie und  $\bar{X}$  als Mittelwert der Serie.

Zum Schätzen des Hurst Parameters wird eine Gerade mit der Steigung  $s$  durch die Punkte gelegt. Dann ist:  $H = 1 - s$

**Aggregate Variance / Variance Time Plot** Dieser Schätzer gehört zu den grafischen zeitbasierten Schätzern. Er ist ähnlich aufgebaut wie der Absolute Moments Schätzer.

Er basiert darauf für verschiedene stark aggregierte (Blockgröße  $m$ ) Varianten der zu untersuchenden Zeitserie die jeweilige Stichprobenvarianz zu berechnen, und dann beides logarithmisch skaliert in einem Graphen darzustellen. [24, 15, 25]

Berechnung:

$$\hat{Var}X^{(m)} = \frac{1}{N/m} \sum_{k=1}^{N/m} \left( X^{(m)}(k) - \bar{X} \right)^2$$

$X^{(m)}(k)$ ,  $N$  und  $\bar{X}$  wie bei Absolute Moments.

Zum Schätzen des Hurst Parameters wird eine Gerade mit der Steigung  $s$  durch die Punkte gelegt. Dann ist:  $H = 1 + \frac{s}{2}$

**Periodogram** Dieser Schätzer gehört zu den grafischen frequenzbasierten Schätzern.

Er basiert darauf, die spektrale Dichte für die verschiedenen Frequenzen zu berechnen und beides logarithmisch skaliert in einem Graphen darzustellen. (X-Achse Frequenz, Y-Achse spektrale Dichte)[24, 8, 15, 25]

Berechnung der spektralen Dichte:

$$I(\lambda) = \frac{1}{2\pi N} \left| \sum_{j=1}^N X(j) e^{ij\lambda} \right|^2$$

mit  $\lambda$  als Frequenz,  $N$  als Länge der Serie und  $X$  die Zeitserie.

Die Schätzung des Hurst Parameters erfolgt über die Steigung  $s$  einer eingepassten Geraden.<sup>10</sup> Dann ist  $H = -\frac{1-s}{2}$

**Rescaled Range Statistics (R/S)** Dieser Schätzer gehört zu den grafischen zeitbasierten Schätzern. Es dürfte auch, aufgrund der Nähe zur Definition von Selbstähnlichkeit, einer der bekanntesten und verbreitetsten Schätzer sein.

Er basiert darauf für verschiedene stark aggregierte (Blockgröße  $m$ ) Varianten der zu untersuchenden Zeitserie das jeweilige Ergebnis der R/S Statistik zu berechnen und beides logarithmisch skaliert in einem Graphen darzustellen.

Für die Berechnung der R/S Statistik sei auf die Literatur verwiesen.[24, 25, 19]

Die Schätzung des Hurst Parameters erfolgt über die Steigung  $s$  einer eingepassten Geraden.  $H = s$

**Variance of Residuals** Dieser Schätzer gehört zu den grafischen zeitbasierten Schätzern.

Er basiert darauf für verschiedene stark aggregierte (Blockgröße  $m$ ) Varianten der zu untersuchenden Zeitserie den jeweiligen Mittelwert der Varianz der Verbleibenden Werte zu berechnen und beides logarithmisch skaliert in einem Graphen darzustellen.[24, 25, 8]

Für das genaue Vorgehen bei der Berechnung sei auf die Literatur verwiesen.[24, 25]

Die Schätzung des Hurst Parameters erfolgt über die Steigung  $s$  einer eingepassten Geraden.  $H = \frac{s}{2}$

**Wavelett / Abry-Veith** Dieser Schätzer gehört zu den frequenzbasierten Schätzern. Aufgrund der Art der Bestimmung lassen sich Konfidenzintervalle angeben.

Er basiert auf den Parametern einer Waveletanalyse der zu analysierenden Serie.

Die mathematischen Grundlagen dazu finden sich in der Literatur.[26, 23, 17]

**Whittle** Dieser Schätzer gehört zu den frequenzbasierten Schätzern. Er setzt Long Range Dependence voraus, darf deshalb nicht zum testen auf Long Range Dependence verwendet werden, aber bei bekannter Long Range Dependence-Eigenschaft zur Bestimmung des Hurst-Parameters.

Die Funktionsweise besteht dadrin, dass durch geeignete Parametrisierung eine Verteilungsfunktion der Serie angenähert wird. Liefert aufgrund des Vorgehens auch Konfidenzintervalle.[15, 8]

### 5.3.2 Probleme beim Schätzen des Hurst-Parameters

Bei dem Schätzen des Hurst-Parameters treten in der Literatur[19, 14, 17, 21] beschriebene Probleme auf, die sich teilweise damit erklären lassen, dass in der Praxis es eigentlich nie perfekt eingeschwungene Systeme ohne Störeinflüsse, ohne Trends und Messreihen mit einer Länge (nahe) an unendlich existieren, auf denen die Definitionen basieren.

---

<sup>10</sup>häufig werden nur die niedrigeren Frequenzbereiche betrachtet, da diese wohl weniger Störungen aufweisen

Mit synthetisch erzeugten Messdaten, die sehr nahe am ideal sein sollten, konnten vereinzelt Abweichungen des Hurst-Parameters von fast 20% vom Sollwert beobachtet werden, so das man die Eignung des jeweiligen Schätzers für die gegebenen Daten überprüfen muss.[17]

Als sehr robust gegen Störungen und recht genau gilt der Abry-Veith Schätzer[21, 4], aber in [17] wird gezeigt, dass der Algorithmus recht empfindlich auf Störungen durch weißes Rauschen reagiert.

Viele Schätzer<sup>11</sup> reagieren auf periodische, verrauschte Kurven mit einer falschen Erkennung einer Long Range Dependence Eigenschaft. Störungen von Messreihen mit Long Range Dependence Eigenschaft können je nach Schätzer stark abweichende Werte liefern.[14, 17]

Als Lösung wird bisher nur angeboten verschiedene Schätzer jeweils parallel zu nutzen, und bei stark abweichenden Ergebnissen die Ursachen zu Untersuchen, und danach die Ergebnisse von dem vermeintlich Geeignetsten zu nehmen.

## Teil III

# Experimentelles Nachweisen

## 6 Tools und Traces

### 6.1 Trace

Ein Trace wird eine Messreihe genannt, bei der die zu erfassenden Ereignisse<sup>12</sup> mit den jeweiligen Parametern<sup>13</sup> und mit einem Zeitstempel versehen gespeichert werden. Der Zeitstempel ermöglicht dabei Analysen über das Zeitverhalten zu erstellen.

Zur exemplarischen Auswertung wurden für diese Arbeit Webserverlogfiles ausgewählt, da diese sich aufgrund der Struktur<sup>14</sup> gut Auswerten lassen, und dabei notwendige Konvertierungen selber durchgeführt werden können.

Von den stark verbreiteten Traces, die auf der Analyse von Ethernetpacketen basieren, wurde aufgrund der notwendigen Toolchain abgesehen.

**Analysierter Trace - Fifa World Cup 1998** Als zu analysierender Trace wurden die Logfiles der Fifa Fußball Weltmeisterschaft 1998[2] genutzt, da diese frei verfügbar sind, ausreichende Stichprobengröße für verschiedene Analysen gegeben ist und es auch eine umfassende Auswertung[3] davon vorhanden ist.

Eckdaten aus [3]:

**Zeitraum:** 1. Mai 1998 bis 23. July 1998

---

<sup>11</sup>Whittle, Periodogram, R/S, Wavelett

<sup>12</sup>z.B. Anfragen an einen Server, Paketankünfte

<sup>13</sup>z.B. Paketgröße, übertragene Datenmenge, angeforderte Datei

<sup>14</sup>ASCII-Dateien, einfache Struktur

**Erhobene Daten:** Common Log Format (IP-Adresse, Benutzer falls authentifiziert, Zeitstempel, Anfrage, HTTP-Status-Response, Größe der Antwort)

**Anzahl der Anfragen:** im gesamten Beobachtungszeitraum 1.352.804.107

**Anzahl der IP-Adressen:** 2.770.108

Der Aufbau des Gesamtsystem bestand dabei aus 30 Server, die in 4 Cluster<sup>15</sup> zusammengefasst waren. Die Anfragen wurden dabei mittels Loadbalancer auf möglichst Netztopologisch nahe Cluster verteilt, und dort auf die Server.

## 6.2 Tools zum Schätzen des Hurst-Parameters

Es konnten in der Literatur vier Tools gefunden werden, die Schätzer für den Hurst-Parameter implementieren.

**Hurst Estimator 2[12]** ist ein frei verfügbares in C++ geschriebenes Programm, welches einen Hurst-Parameterschätzer auf wavelett-Basis mittels Rescaled-Range-Statistic implementiert.

Aufgrund von der Verwendung von absoluten Pfaden in den Makefiles und dem Voraussetzen von Bibliotheken einer bestimmten IDE, lässt es sich, trotz gegenteiliger Aussage des Autors<sup>16</sup>, nicht mit akzeptablem Zeitaufwand kompilieren, so dass bei dieser Arbeit nicht weiter berücksichtigt wurde.

**LASS[23]** ist eine Erweiterung für die Software MATLAB. Die Erweiterung implementiert einen Hurst-Parameterschätzer auf wavelett-Basis, mit ausgefeilten grafischen Auswertungsmöglichkeiten.

Sie ist über die Autoren zu beziehen, und wurde in dieser Arbeit nicht weiter betrachtet, da die Ausgangsbasis MATLAB nicht vorhanden ist.

**Hurst Exponent estimators v2.0[18]** ist eine Erweiterung für die Scilab-Toolbox[22]. Sie implementiert eine Reihe von Hurst-Parameter-Schätzer (Variance of Residuals, Aggregated Variance, Absolute Moments, Rescale Range(R/S) und Periodogram) und ist wie die Scilab-Toolbox frei verfügbar.

Da diese Erweiterung nur Methoden zur Berechnung liefert, deren Nutzung nur so rudimentär dokumentiert ist, dass ohne Wissen über die Funktionsweise der scilab-Toolbox nicht einmal die Beispiele nachvollzogen werden konnten, wurde sie nicht weiter in der Arbeit berücksichtigt.

---

<sup>15</sup>einer in Paris, drei andere in den USA

<sup>16</sup>„It should not be too difficult to convert these Makefiles for UNIX“

**Selfis[15]** ist ein in Java geschriebenes Programm welches eine Reihe von von Hurst-Parameterschätzer implementiert ( Absolute Value, Aggregate Variance, R/S, Variance of Residuals, Periodogram, Whittle und Abry-Veitch/wavelett).

Es ist als Binärversion frei Verfügbar, und unterstützt Datensätze mit bis zu 65.535 Einträge.

Es wurde im folgenden als Tool zur Bestimmung des Hurst-Parameters verwendet.

## 7 Vorgehen

Im Folgenden wird das Vorgehen beschrieben um die Daten geeignet analysieren zu können, welches im wesentlichen für alle Auswertungen gleich war<sup>17</sup>.

### 7.1 Datenaufbereitung & Verdichtung

**Datenaufbereitung** Da die Traces in einem Binärformat (siehe 10.1) konvertiert wurden, bevor sie zum Download bereitgestellt wurden<sup>18</sup>, müssen die Daten zunächst geeignet aufbereitet werden.

In der Toolsammlung[1], die zu den analysierten Traces[2] gehört, befinden sich mehrere C-Programme, welche die Arbeit mit den Binärdaten vereinfachen.

Darunter befindet sich das Tool **recreate**, welches aus den Binärdaten wieder ein Logfile im Common Logfile Format erstellt. Dieses wurde für die Analysen geändert, so das statt eines Zeitstempels in der Form „[DD/MMM/YYYY:HH:MM:SS +-ZZZZ]“ der Unix-Timestamp ausgegeben wurde.

Die Verwendung des Unix-Zeitstempels hat den Vorteil, das im Verlauf der weiteren Analysen die Daten leichter sortiert werden können, und auch leichter auf Lücken mit 0 Datensätzen zu einem Zeitpunkt kontrolliert werden können.

**Dattenverdichtung** Um die Daten auf die gewünschten Werte zu verdichten werden von dem Programm **awk** angebotene assoziative Arrays genutzt.

Die Funktionsweise wird anhand eines Beispiels, welches die Daten auf die Anzahl der Requests pro Sekunde verdichtet, erklärt.

```
1 awk -v start=894772800 -v ende=894830400 '{
2   sum[$4] ++
3 }';
4 END{for (i=start;i<ende;i++) {
5   print sum[i]+0
6   }
7 }' Input.logfile >Output.data
```

In Zeile 1 wird das Programm **awk** aufgerufen, und nach dem „-v“ jeweils eine Variable übergeben. Die Variablen **start** und **ende** werden in dem Beispiel genutzt, um den Zeitraum der ausgegeben werden soll zu definieren.

In den Hochkommata eingeschlossen befindet sich das eigentliche Programm.

---

<sup>17</sup>Besonderheiten werden bei den einzelnen Auswertungen benannt

<sup>18</sup>diente hauptsächlich der Datenreduktion, und der besseren Verarbeitbarkeit mit eigenen Programmen.

Innerhalb der ersten geschweiften Klammern (Zeile 1-3) befindet sich der Befehl der pro Zeile ausgeführt werden soll. Er besteht in diesem Fall aus dem Erhöhen eines Zählers des assoziativen Arrays „sum“ an der Position, die durch das vierte Datenfeld der Zeile definiert wird. Die Datenfelder werden durch Tabstops getrennt, so das das vierte Feld bei dem Common-Logfile Format dem Zeitstempel entspricht.

In den geschweiften Klammern nach dem Schlüsselwort „END“ (Zeile 4-7) wird die Ausgabe realisiert, die nach dem Einlesen aller Zeilen durchgeführt wird. Diese besteht aus einer Schleife, die für jeden Zeitpunkt zwischen „start“ und „ende“ den Zähler ausgibt, und falls keine Daten vorhanden sind 0.

In der 7. Zeile werden noch die Eingabedaten definiert, und die Ausgabe in eine Datei umgelenkt.

## 7.2 Analyse

Die Analyse fand dann durch das Laden der entsprechend vorbereiteten Daten in das Tool selfis statt, welches den Hurst-Parameter mit den gewünschten Schätzer bestimmte.

## 8 Auswertung

In diesem Kapitel werden exemplarisch mehrere Messungen ausgewertet, und bei den Ergebnissen evtl. vorhandene Besonderheiten vorgestellt.

### 8.1 Anfragen pro Sekunde am 10. Mai 1998 5-21 Uhr

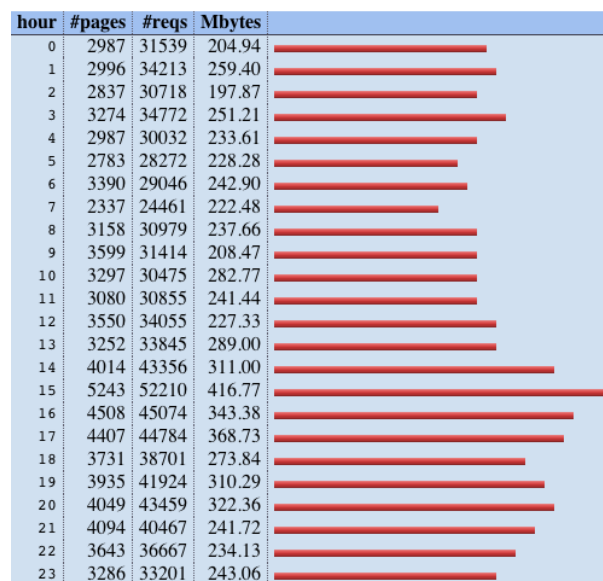


Abbildung 5: Logfileauswertung 10. Mai mit analog

Wie in Abbildung 5 zu erkennen, zeigt der Verlauf der Anfragen pro Stunden keine Besonderheiten.

Eine Analyse wie in Kapitel 7 beschrieben lieferte die in Tabelle 1 abgebildeten Ergebnisse.

Schätzer	Hurst Exponent	Korrelationskoeffizient	95% Konfidenzintervall
Absolute Moments	0,756	86,20%	-
Aggregate Variance	0,819	97,6%	-
Periodogram	0,752	-	-
R/S	0,697	99,16%	-
Variance of Residuals	0,825	99,31%	-
Wavelett / Abry-Veith	0,786	-	0,778-0,794
Whittle	0,718	-	0,711-0,726

Tabelle 1: Ergebnisse der Auswertung AnfragenSekunde vom 10. Mai

Diese liegen dabei alle relativ nah um 0,75, so das man von Long Range Dependence ausgehen kann, mit einem Hurst Parameter  $H \approx 0,75$ .

## 8.2 Byte pro Sekunde am 10. Mai 1998 5-21 Uhr

Um die Verteilung der Daten zu analysieren wird folgende vereinfachende Annahme gemacht:

- alle Daten werden zu dem Zeitpunkt der jeweiligen Anfrage übertragen

Diese Annahme ist bei einer Auflösung von 1 Sekunde, gerade bei größeren Dateien, natürlich nicht genau, aber da es keine Möglichkeiten gibt die wahren Übertragungszeiten zu rekonstruieren muss diese so bestehen bleiben.

Das awk-Skript zur Datenaggregation wird dafür auch leicht modifiziert, um statt der Anfragen die übertragenen Daten in Byte zu zählen. Siehe Anlage 10.2.

Bei der Auswertung, die in Tabelle 2 abgebildet ist, fallen im Vergleich zu den Anfragen/Sekunde für den gleichen Zeitraum zwei Dinge ins Auge:

1. Der Hurst Parameter scheint für die Verteilung des Datentransfers mit  $H \approx 0,55$  deutlich geringer zu sein. Also die Verteilung der Übertragenen Daten pro Sekunde zwar eine Long Range Dependence aufzuweisen ( $H > 0,5$ ), welche aber deutlich geringer ausgeprägt ist.
2. Der Absolute Moments und der Variance of Residuals Schätzer liefern deutlich abweichende Werte.

Das zwei Schätzer stark abweichende Werte haben, bestärkt die schon in Kapitel 5.3.2 genannte Verhaltensmaßregel, immer mehrere Schätzer zu nutzen. Bei den Diagramm für

Schätzer	Hurst Exponent	Korrelationskoeffizient	95% Konfidenzintervall
Absolute Moments	0,102	67,73%	-
Aggregate Variance	0,535	99,46%	-
Periodogram	0,594	-	-
R/S	0,586	99,86%	-
Variance of Residuals	0,805	97,68%	-
Wavelett / Abry-Veith	0,561	-	0,553-0,569
Whittle	0,521	-	0,514-0,528

Tabelle 2: Ergebnisse der Auswertung Daten/Sekunde vom 10. Mai

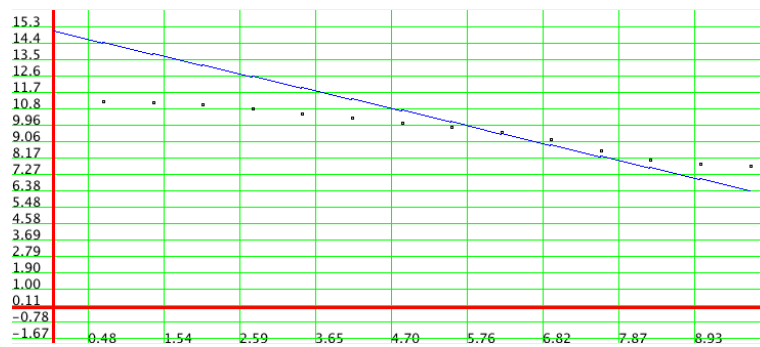


Abbildung 6: Absolute Moments Schätzer für Daten/Sekunde vom 10. Mai

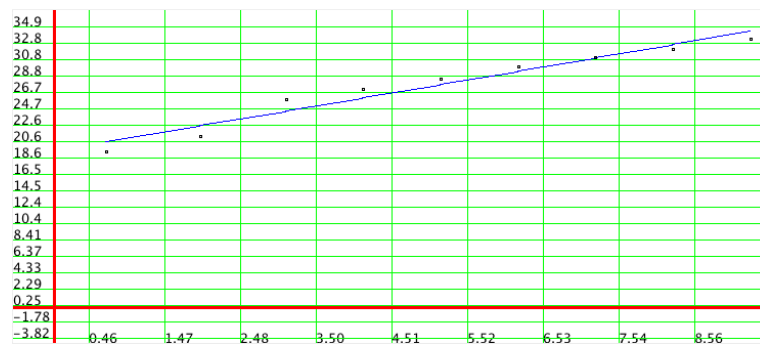


Abbildung 7: Variance of Residuals Schätzer Estimator für Daten/Sekunde vom 10. Mai

den Absolute Moments Schätzer kann man noch anhand der schlechten Einpassung der Geraden, siehe Abbildung 6, auf ein nicht unbedingt passendes Ergebnis schließen. Bei dem Diagramm für den Variance of Residuals Schätzer in Abbildung 7 lassen sich keine solchen Fehler erkennen, so dass man ohne den Vergleich zu anderen Schätzern von einem korrekten Hurst Parameter ausgegangen wäre.

### 8.3 Anfragen pro Sekunde am 30 Juni 8-24 Uhr

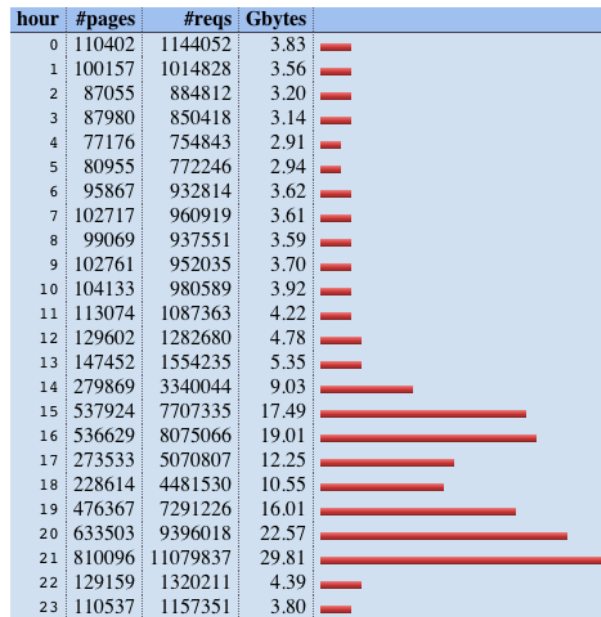


Abbildung 8: Logfileauswertung 30. Juni mit analog

Wie in Abbildung 5 zu erkennen, zeigt der Verlauf der Anfragen pro Stunden am 30. Juni am späten Nachmittag bis Abend einen hohen Peak, so dass anhand dieses Diagramms schon zu vermuten ist, das aufgrund der vermutlich nicht vorhandenen annähernden Stationarität, die Ergebnisse stärker streuen können.

Eine Analyse wie in Kapitel 7 beschrieben lieferte die in Tabelle 3 abgebildeten Ergebnisse.

Bei diesen Ergebnissen fällt zuerst auf, dass einige Schätzer (Periodogram, Variance of Residuals, Abry-Veith) einen Hurst-Parameter  $> 1$  bestimmt haben, welcher nicht definiert ist. Dies liegt, wie schon als Vermutung geäußert, vermutlich an den sehr starken abweichen von dem Ideal<sup>19</sup>.

Die anderen Schätzer verteilen sich auf die Werte knapp unter 1 ( $H \approx 0,99$ ) und Werte zwischen 0,7 und 0,76.

Anhand dieses Beispiels kann man sehen, dass sich bei Datenreihen die stark von den Annahmen abweichen, die Schätzung des Hurst Parameters faktisch unmöglich werden

<sup>19</sup>keine Stationarität, eindeutige Tendenzen, und Perioden

Schätzer	Hurst Exponent	Korrelationskoeffizient	95% Konfidenzintervall
Absolute Moments	0,763	44,35%	-
Aggregate Variance	0,993	35,33%	-
Periodogram	1,172	-	-
R/S	0,705	93,49%	-
Variance of Residuals	1,445	97,305	-
Wavelett / Abry-Veith	1,010	-	1,004-1,015
Whittle	0,999	-	0,998-1,001

Tabelle 3: Ergebnisse der Auswertung Requets/Sekunde vom 30. Juni

kann. In so einem Fall muss man überprüfen, ob nicht andere Erklärungsmuster<sup>20</sup> für die Ankunftsrate zu wählen sind, und man dann die Verteilungsfunktionen für kürzere Zeiträume<sup>21</sup> analysiert.

## Teil IV

# Zusammenfassung

## 9 Zusammenfassung

Selbstähnliche und Long Range Dependend Prozesse modellieren einige Prozesse in Weitverkehrsnetzen, insbesondere im Internet, deutlich besser als die klassischen Poisson-Prozesse.

Es gibt aber gerade bei der Bestimmung der Parameter in der Praxis noch deutlichen Verbesserungsbedarf, da die existierenden Schätzer sich häufig in die Irre führen lassen, und auch noch relativ ungenau sind.

### 9.1 Bedeutung für die Praxis

Die Auswirkungen auf die Praxis dürften geringer sein, als z.B. das Diagramm 1 in Kapitel 3.2 suggerieren aus verschiedenen Gründen:

- in der Praxis werden ohnehin meist größere Reservekapazitäten bereitgehalten um mögliche ungeplante Peaks die durch das Benutzerverhalten ausgelöst werden kompensieren zu können.[10] Diese können im Alltag natürlich auch die Peaks, welche die Long Range Dependence impliziert abfangen.

<sup>20</sup>in diesem Fall atypisches Nutzerverhalten durch wichtiges Fußballspiel

<sup>21</sup>also z.B. nur für den Zeitraum des Peaks

- im Backbonebereich ist durch das sehr starke Multiplexen nach einigen Studien das Verhalten wieder nahe einer Poissonverteilung.[16]
- Die meisten Dienste nutzen als Übertragungsprotokoll TCP, welches die Datenrate aufgrund der steigenden Latenz bei sich füllenden Puffer reduziert, und mit Paketverlusten umgehen kann. Deshalb ist es in der Praxis im Netzwerkbereich aktuell<sup>22</sup> noch kein wichtiges Thema.[10]

## 9.2 Anmerkungen

Bei der Erstellung der Arbeit fielen die folgenden Punkte auf:

- es existieren wenige Paper bei denen die Auswertung gut nachvollziehbar ist (Datenquelle über Aufbereitung bis zur Analyse).
- Für verschiedene Quellen (z.B. die Webserverlogfiles, allgemein) keine an vorhandenen Daten nachprüfbar Aussagen auffindbar.
- Toollandschaft sehr eingeschränkt.
- Häufig sehr schwierig Material zu finden welches zwischen den „Dreizeiler“-Zusammenfassungen liegt und den Originalpaper (welches teilweise von Mathematikern geschrieben wurde, und damit nicht unbedingt intuitiv verständlich ist), um gewisse Zusammenhänge/Definitionen zu verstehen.

---

<sup>22</sup>mit der steigenden Verbreitung von Diensten die auf geringe Verlustraten angewiesen sind und keine ausgefeilte Flusskontrolle haben, z.B. Live-Videostreams über UDP, könnte sich das ändern.

# Teil V

## Anhang

### 10 Anhaenge

#### 10.1 Datenformat der Worlc Cup 1998 Traces

```
struct request
{
  uint32_t timestamp;
  uint32_t clientID;
  uint32_t objectID;
  uint32_t size;
  uint8_t method;
  uint8_t status;
  uint8_t type;
  uint8_t server;
};
```

The fields of the request structure contain the following information:

**timestamp** the time of the request, stored as the number of seconds since the Epoch. The timestamp has been converted to GMT to allow for portability. During the World Cup the local time was 2 hours ahead of GMT (+0200). In order to determine the local time, each timestamp must be adjusted by this amount.

**clientID** a unique integer identifier for the client that issued the request (this may be a proxy); due to privacy concerns these mappings cannot be released; note that each clientID maps to exactly one IP address, and the mappings are preserved across the entire data set - that is if IP address 0.0.0.0 mapped to clientID X on day Y then any request in any of the data sets containing clientID X also came from IP address 0.0.0.0

**objectID** a unique integer identifier for the requested URL; these mappings are also 1-to-1 and are preserved across the entire data set.

**size** the number of bytes in the response

**method** the method contained in the client's request (e.g., GET). Mappings for this are contained in src/\*/definitions.h

**status** this field contains two pieces of information; the 2 highest order bits contain the HTTP version indicated in the client's request (e.g., HTTP/1.0); the remaining 6 bits indicate the response status code (e.g., 200 OK). Mappings for the HTTP version and the status codes are contained in src/\*/definitions.h.

**type** the type of file requested (e.g., HTML, IMAGE, etc), generally based on the file extension (.html), or the presence of a parameter list (e.g., '?' indicates a DYNAMIC request). If the url ends with '/', it is considered a DIRECTORY. Mappings

from the integer ID to the generic file type are contained in definitions.h. If more specific mappings are required this information can be obtained from analyzing the object mappings file (state/object\_mappings.sort).

**server** indicates which server handled the request. The upper 3 bits indicate which region the server was at (e.g., SANTA CLARA, PLANO, HERNDON, PARIS); the remaining bits indicate which server at the site handled the request. All 8 bits can also be used to determine a unique server. Mappings for the region are contained in src/\*/definitions.h.

## 10.2 awk-Skript für Daten pro Sekunde

```
1 awk -v start=894772800 -v ende=894830400 '{
2   sum[$4] += $9
3 }';
4 END{for (i=start;i<ende;i++) {
5     print sum[i]+0
6   }
7 }' Input.logfile >Output.data
```

## 10.3 Abbildungsverzeichnis

### Abbildungsverzeichnis

1	Benötigte Puffergröße Poisson vs. LRD. Aus [10]	4
2	Sierpinski Dreieck (Quelle: Wikipedia Commons)	4
3	Selbstähnliche Verkehrskurve aus [11]. Ausschnitte aus der jeweils darüber liegenden Kurve haben jeweils ähnliches Verhalten.	5
4	Beispiel für Größe von Dateien die über ein Netz übertragen wurden.[7]	7
5	Logfileauswertung 10. Mai mit analog	13
6	Absolute Moments Schätzer für Daten/Sekunde vom 10. Mai	15
7	Variance of Residuals Schätzer Estimator für Daten/Sekunde vom 10. Mai	15
8	Logfileauswertung 30. Juni mit analog	16

## 10.4 Tabellenverzeichnis

### Tabellenverzeichnis

1	Ergebnisse der Auswertung AnfragenSekunde vom 10. Mai	14
2	Ergebnisse der Auswertung Daten/Sekunde vom 10. Mai	15
3	Ergebnisse der Auswertung Requensts/Sekunde vom 30. Juni	17

## Literatur

- [1] Worldcup98 tools. URL <http://ita.ee.lbl.gov/html/software.html>.
- [2] M. Arlitt and T. Jin. 1998 World Cup Web Site Access Logs, August 1998. URL <http://ita.ee.lbl.gov/html/contrib/WorldCup.html>.
- [3] M. Arlitt and T. Jin. Workload characterization of the 1998 world cup web site. HP Labs Technical Reports 1999-35R1, 1999.
- [4] J.-Y. L. Boudec. PERFORMANCE EVALUATION OF COMPUTER AND COMMUNICATION SYSTEMS, May 2007. URL <http://perfeval.epfl.ch>.
- [5] C. D. Cairano-Gilfedder and R. G. Clegg. A decade of internet research: advances in models and practices, 09 2007.
- [6] O. Cappe, E. Jean-Christophe, P. Petropulu, and X. Yang. Long-range dependence and heavy-tail modeling for teletraffic data, 2002.
- [7] M. Crovella and M. Taqqu. Estimating the heavy tail index from scaling properties, 1999.
- [8] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.
- [9] A. Downey. Evidence for long-tailed distributions in the Internet, 2001.
- [10] N. J. Gunther. *Guerrilla Capacity Planning*. Springer Verlag Berlin-Heidelberg, 2007. ISBN 978-3-540-26138-4.
- [11] A. Gupta. Self-similar traffic. Lecture Advanced Network Technologies.
- [12] I. Kaplan. Hurst estimator 2, Mai 2003. URL [http://www.bearcave.com/misl/misl\\_tech/wavelets/hurst/doc/index.html](http://www.bearcave.com/misl/misl_tech/wavelets/hurst/doc/index.html).
- [13] T. Karagiannis. SELFIS: A Short Tutorial, November 2002.
- [14] T. Karagiannis, M. Faloutsos, and R. Riedi. Longrange dependence: now you see it, now you don't, 2002.
- [15] T. Karagiannis, M. Faloutsos, and M. Molle. A User-Friendly Self-Similarity Analysis Tool, 2003.
- [16] T. Karagiannis, M. Molle, and M. Faloutsos. Long-Range Dependence - Ten Years of Internet Traffic Modeling. *IEEE Internet Computing*, September \* October 2004: 57–64, 2004.
- [17] T. Karagiannis, M. Molle, and M. Faloutsos. Understanding the Limitations of Estimation Methods for Long-Range Dependence, 2006.

- [18] F. Melakessou. Hurst exponent estimators v2.0, July 2007. URL [http://www.scilab.org/contrib/index\\_contrib.php?page=displayContribution&fileID=988](http://www.scilab.org/contrib/index_contrib.php?page=displayContribution&fileID=988).
- [19] S. Molnar and T. Dang. Pitfalls in long range dependence testing and estimation, 2000.
- [20] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.
- [21] M. Roughan and D. Veitch. Measuring long-range dependence under changing traffic conditions. In *INFOCOM (3)*, pages 1513–1521, 1999.
- [22] Scilab Consortium. Scilab. URL <http://www.scilab.org/>.
- [23] S. Stoev, M. S. Taqqu, C. Park, G. Michailidis, and J. S. Marron. Lass: a tool for the local analysis of self-similarity. *Computational Statistics & Data Analysis*, 50(9):2447–2471, 2006.
- [24] M. Taqqu and V. Teverovsky. On estimating the intensity of long-range dependence in finite and infinite variance time series, 1996.
- [25] M. Taqqu, V. Teverovsky, and W. Willinger. Estimators for long-range dependence: an empirical study, 1995.
- [26] D. Veitch and P. Abry. A wavelet based joint estimator of the parameters of long-range dependence. *IEEE Transactions on Information Theory*, 45(3):878–897, 1999.