Technische Universität Hamburg-Harburg

Arbeitsbereich Digitale Kommunikationssysteme

Prof. Dr. rer. nat. Ulrich Killat

Erhöhung des Nutzungsgrades eines ATM Netzes für den Wissenschaftsbereich (ERNANI)

Abschlussbericht

Kai Below, Carsten Schwill

März 2000

(Überarbeitete Fassung, September 2001)

Inhaltsverzeichnis

1 Einleitung **1**

2 Knotenmodelle **5**

2.1 Modell des Netzwerk-Knotens 5

2.1.1 Switching 6

2.1.2 Routing 7

2.1.3 Rufannahme 9

2.2 Modell eines IP-Routers 11

2.2.1 Notwendige Kommunikationsschichten 11

2.2.2 ATM Adaptation Layer (AAL) 11

2.2.3 Internet Protocol (IP) 13

2.2.4 Gegenwärtige Benutzung von Routing im B-WiN 14

2.3 Realisierung der Regelung für ABR 15

2.3.1 Rückkopplung durch RM-Zellen 15

2.3.2 Available Bit Rate (ABR) Algorithmen 17

2.3.3 Verifikation und Veranschaulichungen 21

3 Quellenmodelle **25**

3.1 Quellen ohne Flusskontrolle 25

3.1.1 N-Burst . . . 25
3.1.2 SupFRP . . . 26
3.2 TCP On/Off Quelle . . . 28
3.3 Validierung der Quellenmodelle . . . 31
3.3.1 Quellen ohne Flusskontrolle . . . 31
3.3.2 TCP On-Off Quelle . . . 32
3.4 Parametrisierung der TCP On/Off-Quelle . . . 34
3.4.1 Variation des power-tail Index α der Dateilängenverteilung . . . 36
3.4.2 Variation der Quellenzahl bei näherungsweise konstanter Last . . . 39
3.5 Erweiterung der TCP On/Off-Quelle um ein HTTP-Modell . . . 40
3.6 Parametrisierung der Quelle im B-WiN Modell . . . 41
3.7 Zusammenfassung . . . 43

4 Netzmodell **45**
4.1 Geometrie und Parameter des Netzmodells . . . 46
4.2 Verteilung der Quellen auf Virtuellen Pfade . . . 49
4.3 IP Netzmodell . . . 50
4.4 Parameter der B-WiN Simulationen . . . 51

5 Warteschlangen-Dimensionierung bei selbstähnlichem Verkehr **55**
5.1 Qualitäts-Parameter für den Leistungs-Vergleich . . . 56
5.2 Poisson Referenz Verkehr . . . 57
5.3 TPT On/Off TCP Modell . . . 58
5.4 Ergebnisse . . . 59
5.4.1 M/D/1/B Referenz System . . . 59
5.4.2 Poisson On/Off TCP Referenz . . . 60
5.4.3 Zusammenfassung . . . 62

6 IP Netzmodell: Ergebnisse **65**

6.1 Lasterzeugung 65

6.2 Verschiedene Lastfälle 68

6.3 Aufteilung der Raten 69

7 Einfluss von Early Packet Discard (EPD) **75**

8 Netzmodell mit ABR Regelung **79**

8.1 Modellierung 79

8.2 Ergebnisse der Simulation 80

9 Einfluss von dynamischem Routing **87**

9.1 Verbindungsaufbau und -abbau im ATM Netz 88

9.2 Ergebnisse 91

10 Zusammenfassung **95**

A Selbstähnlichkeit und Hurst Parameter **97**

A.1 Der Zufallsprozess 97

A.2 Selbstähnlichkeit 98

A.3 Selbstähnlichkeit Zweiter Ordnung 98

A.4 Heavy- / Power- / Long-Tailedness 99

A.5 Umrechnung von Hurst Parameter H / α / β 100

A.6 Benutzte Hurst Parameter Schätzer 100

A.6.1 Variance-Time Plot 100

A.6.2 R/S-Plot 100

B Überlastabwehr in TCP **103**

B.1 Slow Start und Congestion Avoidance 103

B.1.1 Slow Start 103

B.1.2 Congestion Avoidance 104

B.1.3 Slow Start Threshold 105

B.2 Fast Retransmit und Fast Recovery 105

C Die “Free Bit Rate” (FBR) Dienstkategorie **113**

C.1 Datenbasis für die Regelung von FBR-Quellen 115

C.2 Einfluss auf den Rufaufbau “normaler” ATM-Verbindungen 116

D Abkürzungen **119**

Literaturverzeichnis **123**

Abbildungsverzeichnis

2.1 B-WiN Knoten. 6

2.2 ATM Prioritäts-Warteschlangen Modul. 7

2.3 VCI Switching im Modell “SWITCH” mit einer Routingtabelle je Eingangsport. 8

2.4 Simulierte Kommunikationsschichten für Internetverkehr mit maximalen Datagramm-Größen. 12

2.5 Aufteilung eines AAL5 Rahmens auf mehrere ATM-Zellen. 12

2.6 Aufteilung eines IP-Datagramms auf mehrere IP-Fragmente. 13

2.7 ABR-Regelschleife für eine Verbindung von A nach B. 16

2.8 EFCI: Pufferbelegung beim Einschalten einer CBR-Quelle. 22

2.9 ERICA: Pufferbelegung beim Einschalten einer CBR-Quelle. 23

3.1 Überlagerunsgsschema der N-Burst Quelle. 26

3.2 Komplementäre Verteilungsfunktion der TPT-Verteilung. 27

3.3 Zwischenankunftszeiten der SupFRP. 27

3.4 Schema der Aggregierung von TCP On/Off-Quellen. 30

3.5 Schätzung der Selbstähnlichkeit zweiter Ordnung der 20-FRP Zwischenankunftszeiten. 32

3.6 Schätzung der Selbstähnlichkeit zweiter Ordnung des 20-FRP Zählprozesses. 33

3.7 Schätzung der Selbstähnlichkeit zweiter Ordnung der 20-Burst Zwischenankunftszeiten. 33

3.8 Schätzung der Selbstähnlichkeit zweiter Ordnung des 20-Burst Zählprozesses. 34

3.9 Schätzung der Selbstähnlichkeit zweiter Ordnung der 20-TCP Zwischenankunftszeiten. 35

3.10 Schätzung der Selbstähnlichkeit zweiter Ordnung des 20-TCP Zählprozesses. 35

3.11 Schätzung des Hurst Parameters bei Variation des Tail-Index α der Dateilängenverteilung, 25 Quellen, Mittelwerte und Standardabweichungen aus jeweils 10 Messungen. 37

3.12 Quadrierter Variations-Koeffizient und mittlere Last ρ als Funktion des tail-Index α der Dateilängenverteilung, 25 Quellen, Mittelwerte und Standardabweichungen aus jeweils 10 Messungen. 37

3.13 Schätzung des Hurst Parameters bei Variation des Tail-Index α der Dateilängenverteilung, 10 Quellen, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen. 38

3.14 Quadrierter Variations-Koeffizient und mittlere Last ρ als Funktion des Tail-Index α der Dateilängenverteilung, 10 Quellen, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen. 39

3.15 Quadrierter Variations-Koeffizient und mittlere Last ρ als Funktion der Quellen-Anzahl, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen. 40

3.16 Schätzung des Hurst Parameters bei Variation der Quellen-Anzahl unter näherungsweise konstanter Last, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen. 41

4.1 Linkraten im B-WiN. 47

4.2 Das B-WiN Netzmodell. 48

4.3 B-WiN Knoten. 48

4.4 B-WiN Quellen-Senken Modul. 49

4.5 ATM Prioritäts-Warteschlangen Modul. 49

4.6 Berechnete Auslastung der Links in (a) Mbit/s bzw. (b) %, als Resultat von Verkehrs-Matrix, Link-Dimensionierung und Routing Tabelle. 51

4.7 B-WiN Knoten IP Netzmodell. 52

4.8 B-WiN Quellen-Senken Modul IP Netzmodell. 52

4.9 Warteschlangen Modul IP Netzmodell. 53

5.1 $Z = f(G,\alpha)$ USA $\rightarrow$ Köln, TPT On/Off TCP mit M/D/1/B als Referenzsystem (oben links bzw. Ausschnitt oben rechts), Goodput (unten links) und Durchsatz (unten rechts). 60

5.2 $Z = f(I,\alpha)$ für $I = 10$ (links) und $I = 100$ (rechts), USA $\rightarrow$ Köln. 61

5.3 $Z = f(I,\alpha)$ für $I = 10$ (links) und $I = 100$ (rechts), Frankfurt $\rightarrow$ Köln. . . 61

5.4 $Z = f(I,\alpha)$ für $I = 10$ (links) und $I = 100$ (rechts), Frankfurt $\rightarrow$ München. 62

6.1 Auslastung der Ports (oben) im Vergleich zu der ursprünglichen Verkehrsmatrix (unten), Messwerte links in Mbit/s, rechts in Prozent der verfügbaren Bandbreite. 66

6.2 Durchsatz in Prozent von der gemessenen Verkehrsmatrix, erster Ansatz. 67

6.3 Last auf den Links (links), Durchsatz in Prozent der gemessenen Verkehrsmatrix (rechts), zweiter Ansatz. 67

6.4 Goodput und Ende-zu-Ende Verzögerung für $I = 2$, $\alpha = 1.3$ (Oben), $\alpha = 1.5$ (Mitte), $\alpha = 1.9$ (Unten). 69

6.5 Goodput und Ende-zu-Ende Verzögerung für $I = 100$, $\alpha = 1.3$ (Oben), $\alpha = 1.5$ (Mitte), $\alpha = 1.9$ (Unten). 70

6.6 Mittlere Ende-zu-Ende Verzögerung (links) und deren Standardabweichung (rechts) für $\alpha = 1.5$, $I = 4$ (Oben), $I = 5.55$ (Mitte) bzw. $I = 10$ (Unten). 71

6.7 Durchsatz individueller TCP Verbindungen am Knoten Frankfurt, Port 4 (München) für $I = 2$ (oben links), $I = 4$ (oben rechts), $I = 10$ (unten links) und $I = 100$ (unten rechts). 73

7.1 Goodput, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 10$. . . 76

7.2 Mittlere Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 10$ (man beachte die unterschiedlich skalierten Ordinaten). 77

7.3 Standardabweichung der Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 10$. 77

7.4 Goodput, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 100$. . 78

7.5 Mittlere Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 100$. 78

7.6 Standardabweichung der Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 100$. 78

8.1 Verwendete Kennlinie des Nutzungsfaktors $f(Q, Q_0)$ bei ERICA+. . . . 80

8.2 Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5, I = 2$. 81

8.3 Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5, I = 4$. 82

8.4 Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5, I = 6$. 83

8.5 Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5, I = 10$. 84

8.6 Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5, I = 100$. 85

9.1 Schematische Darstellung des Verbindungsaufbaus. 88

9.2 Schematische Darstellung des Verbindungabbaus. 89

9.3 Zustandsdiagramm eines Teilnehmers für den Verbindungsauf- und -abbau. 90

9.4 Relative Linklast auf dem ersten bzw. zweiten Link von 11 Knoten des B-WiN; Simulationsergebnisse mit 440 Quellen und 440 Senken für statisches und dynamisches Routing. 91

9.5 Vergleich des durch das Netz übertragenen Datenvolumens in Megabyte (Goodput). 92

9.6 Vergleich der absoluten Netzlast im Megabyte (Throughput). 92

9.7 Vergleich der Anzahl der aufgebauten Verbindungen in Abhängigkeit von der Quellenzahl. 93

A.1 Beispiel eines Variance-Time Plots. 101

A.2 Beispiel eines R/S-Plots. 102

B.1 TCP Tahoe und TCP Reno ohne Begrenzung des Fast Retransmit Algorithmus. 109

B.2 TCP Tahoe und TCP Reno mit Begrenzung des Fast Retransmit Algorithmus. 111

B.3 TCP Tahoe ohne/mit Begrenzung des Fast Retransmit Algorithmus. . . . 112

C.1 Pufferbelegung beim Einschalten % einer CBR-Quelle ohne Verzögerung. 117

Tabellenverzeichnis

3.1 Messwerte am Knoten München (Port 1, 3 und 5), Anzahl der Quellen N (die Verkehr über diesen Port schicken) bei $I = 1$ HTTP-Includes: Hurst Parameter H_{VT} (VT-Plot), Länge der Folge L, Mittelwert und Standardabweichung der Zwischenankunftszeiten ($\mu\sigma$), quadrierter Variationskoeffizient $C^2(X) = \sigma^2/\mu^2$ und Linkauslastung ρ. 42

3.2 Messwerte am Knoten München (Port 1, 3 und 5), Anzahl der Quellen N (die Verkehr über diesen Port schicken) bei $I = 2$ HTTP-Includes: Hurst Parameter H_{VT} (VT-Plot), Länge der Folge L, Mittelwert und Standardabweichung der Zwischenankunftszeiten ($\mu\sigma$), quadrierter Variationskoeffizient $C^2(X) = \sigma^2/\mu^2$ und Linkauslastung ρ. 42

4.1 Gemessene Verkehrs-Matrix auf IP-Ebene, Raten in Mbit/sec. 50

4.2 Wegelenkung auf der Basis von Knotennummern. 50

4.3 Quellen-Verteilung bei insgesamt 1000 Quellen. 51

4.4 Gemeinsame Parameter aller Simulationen. 53

6.1 Mittelwerte der Leistungskenngrößen im IP Netzmodell; G: Goodput, T: Throughput, R: Paket-Verlustrate in %, μ_D: mittlere Ende-zu-Ende Verzögerung, σ_D: Standard Abweichung der Ende-zu-Ende Verzögerung bei einer Puffergröße von $B = 625$ IP Paketen. 72

8.1 Mittlerer Goodput G und Durchsatz T im UBR- und ABR-Fall in Prozent in Abhängigkeit von der Anzahl der HTTP “Includes” I. 83

Kapitel 1

Einleitung

Dieser Bericht entstand im Rahmen des vom DFN Verein in Auftrag gegebenen Projektes zur "Erhöhung des Nutzungsgrades eines ATM Netzes für den Wissenschaftsbereich (ERNANI"), dessen Zielsetzung im folgenden kurz beschrieben wird: Das Breitband-Wissenschafts-Netz (B-WiN) stellt seinen Kunden einen Anschluss an das weltweite IP-Netz zur Verfügung. Dazu stützt es sich auf IP Router als Netzknoten ab, die ihrerseits über eine ATM Infrastruktur miteinander kommunizieren. Diese Art der Kommunikation lässt sich am besten durch das Konzept "IP über ATM" beschreiben, wobei zwischen benachbarten Routern permanente ATM Verbindungen geschaltet sind und die Router im übrigen die Wegewahl nach dem BGP Protokoll vornehmen. Die in diesem Bericht untersuchte Fragestellung ist, ob Verkehrsmanagementfunktionen aus dem Bereich der ATM Technik geeignet sind, die Effizienz der Ressourcennutzung zu steigern. Dazu ist als erstes der Einfluss der Funktion des "Early Packet Discard" (EPD) zu betrachten, das für den "Unspecified Bit Rate" (UBR) Dienst spezifiziert ist. In dem Szenario des B-WiN macht es ohne die Funktion des EPD dann keinen Sinn mehr, zwischen UBR und CBR zu unterscheiden. Während UBR und CBR keinerlei protokolltechnische Unterstützung erfordern, handelt es sich beim ABR-Dienst um eine flussgesteuerte Datenübertragung, die sowohl ein Protokoll zur Realisierung der Regelschleife als auch erhebliche Pufferressourcen im inneren Netz erfordert, um die gewünschte Dienstgüte in puncto Zell-Verlustwahrscheinlichkeit zu erreichen. Alle genannten Dienstklassen eignen sich prinzipiell zur Etablierung von virtuellen Pfaden zwischen Routern für die immer wichtiger

werdende Übertragung von Daten im "World Wide Web" (WWW). Diese Daten werden zwar in großer Menge, jedoch im Vergleich zu Echtzeit-Anwendungen mit hoher Toleranz gegenüber kurzzeitigen Verzögerungen übertragen. Sie verfügen in der Regel über eigene Mechanismen zur Flusssteuerung und Überlastabwehr (z.B. TCP/IP basierte Anwendungen). Damit stellt sich die Frage, ob mit der Einführung des ABR-Dienstes eine wesentliche Verbesserung des erzielbaren Netzdurchsatzes ("Goodput") erreicht werden kann und welcher Aufwand an Puffern dafür ggf. zu treiben wäre. Ein statisches Routing kann definitionsgemäß die momentane Auslastung der Links nicht berücksichtigen. Mit dem PNNI Protokoll werden Möglichkeiten eines dynamischen Routing in die ATM Technik eingeführt. Auch hier stellt sich wiederum die Frage, wie lohnenswert die Einführung dieser Möglichkeit beim Betrieb des B-WiN ist Zur Beantwortung der angeschnittenen Fragen wurde im Rahmen dieses Projektes ein Simulationsmodell geschaffen, das die Verkehrsverhältnisse im B-WiN möglichst genau nachzustellen versucht, um auf dieser Basis zu verlässlichen Ergebnissen bzgl. effektivem Netzdurchsatz, Verzögerungen auf Anwendungsebene und ggf. Fairness zu kommen.

Während der Bearbeitung des Projektes haben sich einige Verschiebungen der Schwerpunkte ergeben:

- Die Idee, Messungen an einzelnen TCP Verbindungen vorzunehmen, die in einen "Hintergrundverkehr" eingebettet sind, wurde fallen gelassen, da der Hintergrundverkehr keine verlässliche Modellierung des Verkehrsflusses in den Netzknoten erlaubt. Stattdessen wurde mit hohem Aufwand ein Netz mit 1000 TCP/IP Quellen gelegt. Jede dieser 1000 Quellen besitzt einen eigenen Protokoll-Stack, d.h. es wurden einzelne Netzwerk-Klienten modelliert. Auf die Modellierung von CBR Quellen, die letztlich nur die verfügbare Bandbreite auf den Links reduzieren, wurde verzichtet.

- Die von der Fachwelt inzwischen heiß diskutierten Eigenschaften von Quellen, die "heavy tails" und/oder Selbstähnlichkeit zeigen, wurde zusätzlich in die Modellierung einbezogen. Die entsprechenden Parameterstudien wurden an einem reinen TCP/IP Netzmodell vorgenommen, das auch für das G-WiN eine solide Modellierungsbasis liefert.

Der vorliegende Bericht gliedert sich wie folgt:

In Kapitel 2 werden die eingesetzten Knotenmodelle und in Kapitel 3 die entwickelten Quellenmodelle vorgestellt. Kapitel 4 fasst die wesentlichen Modellannahmen und Parameter des Netzmodells zusammen. Das Kapitel 6 beschreibt den Großteil der Untersuchungen, die an einem reinen IP-Netzmodell durchgeführt wurden. Es werden Ergebnisse von Parameterstudien präsentiert, die keine Abhängigkeit von der ATM Schicht haben. Im Kapitel 7 wird der Einfluss von "Early Packet Discard" untersucht, während Kapitel 8 die Option einer ABR basierten Infrastruktur bewertet. Im Kapitel 9 wird schließlich ein ATM Netzmodell mit dynamischen Routing vorgestellt und in seinen Auswirkungen analysiert. Kapitel 10 fasst die erzielten Ergebnisse zusammen.

Der Bericht wird abgeschlossen mit drei Anhängen:

- A: Selbstähnlichkeit und Hurst Parameter
- B: Überlastabwehr im TCP Protokoll
- C: Vorschlag für eine Dienstkategorie. "Free Bit Rate" (FBR)

Kapitel 2

Knotenmodelle

Je nach Sichtweise können unter dem Begriff Netzwerknoten verschiedene Netzwerkkomponenten bzw. -funktionalitäten verstanden werden. So stellt das B-WiN unter anderem den Endkunden einen Anschluss an das weltweite IP-Netz zur Verfügung, so dass aus Sicht dieser Kunden die IP-Router im B-WiN die Netzknoten sind. Aus Sicht des dem B-WiN zugrunde liegenden ATM-Netzes sind dagegen die ATM-Vermittlungen Netzknoten und die IP-Router Endgeräte. Ob sich im Rahmen dieser Arbeit hinter dem Begriff Netzknoten eine ATM-Vermittlung, ein IP-Router oder einfach nur ein geographischer Ort verbirgt, ist darum jeweils dem Kontext zu entnehmen.

In diesem Kapitel wird zunächst in Abschnitt 2.1 das der Simulation zugrunde liegende Modell eines ATM-Netzknotens erläutert. Daran schließt sich mit Abschnitt 2.2 die Modellierung des IP-Routers an. Die Realisierung des ATM-Dienstes ABR in Quellen und Vermittlungen sowie die Verifikation der Implementierung in statischen und dynamischen Szenarien wird in Abschnitt 2.3 erläutert.

2.1 Modell des Netzwerk-Knotens

Der Netzwerk-Knoten in Bild 2.1 wird zusätzlich zu Ein- und Ausgang durch vier wesentliche Elemente gekennzeichnet: Das Quellen-Senken Modul (oben links), den ATM-Switch (mitte), das ATM-Prioritätswarteschlangen Modul (oben rechts) und die Higher

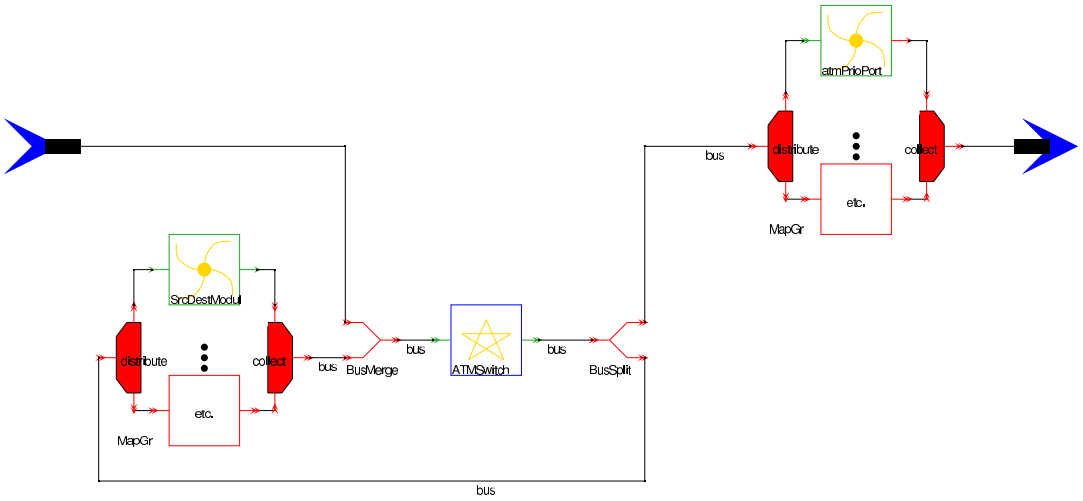


Abbildung 2.1: B-WiN Knoten.

Order Functions (HOF, unten links und rechts), die benutzt werden, um die Anzahl der benötigten Quellen bzw. Warteschlangen zu skalieren. Der ATM-Switch hat die Aufgabe der Abwicklung der Rufannahme bzw. -ablehnung sowie das “switchen” der einzelnen Datenzellen (siehe Abschnitt 2.1.1).

Das ATM-Prioritäts-Warteschlangen Modul in Bild 2.2 sorgt für eine der Verkehrsklasse angemessenen Einordnung in die Warteschlangen, die mit Hilfe des Schedulers (rechts oben) von oben nach unten entsprechend der Priorität abgearbeitet werden. Dabei haben reguläre ATM-Zellen (CBR, VBR) im mit “ATM” gekennzeichneten Puffer die höchste Priorität 0. Erst wenn dieser Puffer leer ist, werden die zu den in Abschnitt 2.3 näher beschriebenen ABR-Verbindungen gehörenden BRM-Zellen verschickt. Ohne diese bevorzugte Behandlung von BRM-Zellen vor den eigentlichen ABR-Zellen würde die Belegung der ABR-Puffer unter ungünstigen Umständen ins Unendliche wachsen, beziehungsweise Zellen verloren gehen. Sollten sich schließlich weder im ATM-, BRM- noch im ABR-Puffer Zellen befinden, werden UBR-Zellen versendet.

2.1.1 Switching

Die eigentliche Hauptaufgabe des Switches gliedert sich in zwei grundsätzliche Aspekte, nämlich des Verteilen ankommender Zellen auf die korrekten Ausgänge und das Ersetzen der nur auf jeweils einem Link gültigen lokalen Kennzeichner zwischen Ein- und Aus-

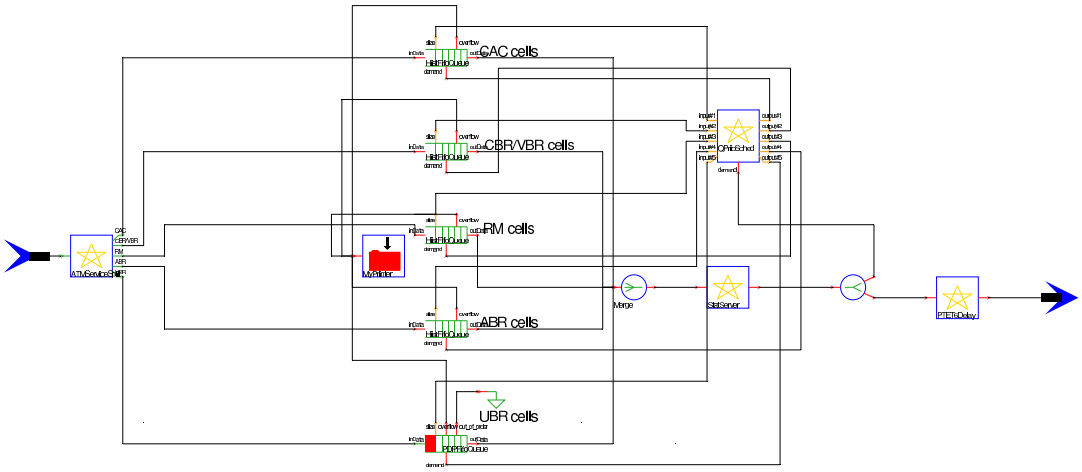


Abbildung 2.2: ATM Prioritäts-Warteschlangen Modul.

gang. In realen Netzen wird dabei zwischen dem übergeordneten “Virtual Path Identifier” (VPI) und dem “Virtual Connection Identifier” (VCI) unterschieden.

In Bild 2.3 ist die Vermittlung einer virtuellen Verbindung von der Quelle zum Ziel mit den zugehörigen VCIs und Ports der “Switches” dargestellt. Im Simulator ist nicht eine global gültige, sondern für jeden Eingangsport eine separate Routingtabelle vorgesehen. Dadurch ist sichergestellt, dass bei dem Aufbau neuer Verbindungen bereits der jeweilige Sender auf einem Link den zugehörigen VCI festlegen kann, da Sender und Empfänger über dieselbe Tabelle verfügen.

2.1.2 Routing

Für das beim Verbindungsaufbau stattfindende Routing aufgrund der Empfängeradresse wird im Simulator auf eine innerhalb eines Switches für alle Ports gültige Routingtabelle zurückgegriffen, die vom Benutzer zuvor gesetzt werden muss. Um die mögliche Leistungsfähigkeit von PNNI im Sinne einer besseren Auslastung des Netzes abzuschätzen, wird das Routing anhand des Dijkstra-Algorithmus durchgeführt, der auch dem PNNI zugrunde liegt. Das dynamische Routing erfolgt auf der Basis fester, zu Rufaufbauzeiten aktualisierter Linkgewichte.

Die Dynamik des Routing kommt durch die Aktualisierung der Linkgewichte zustande, wobei der entscheidende Punkt hier die Lastabhängigkeit ist. Einem Vorschlag der Firma

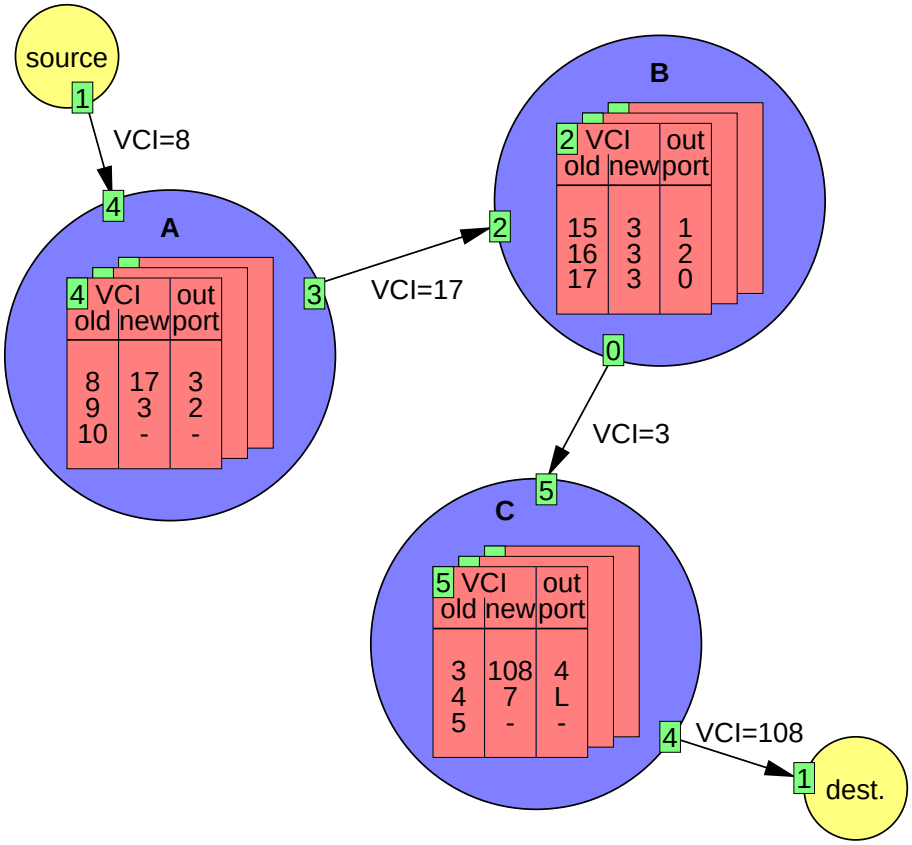


Abbildung 2.3: VCI Switching im Modell “SWITCH” mit einer Routingtabelle je Eingangsport.

Cisco folgend [CIS] wurde für das dynamische Routing die folgende Metrik verwendet:

$$C = \left[\frac{k_1}{bw/10^4} + \frac{k_2}{bw/10^4(256-load)} + \frac{k_3 \cdot delay}{10^4} \right] \cdot \frac{k_5}{reliability + k_4} \qquad (2.1)$$

Hierbei wird mit *bw* die minimale Linkbandbreite des Pfades in Mbit/s bezeichnet, mit *load* ein auf den Bereich $[0,255]$ normierter Messwert der momentanen Last auf dem Link und mit *delay* die gemessene Verzögerung in msec auf den Link. Die Verlässlichkeit dieser Messungen geht mit dem Parameter *reliability* in die Formel ein. Die Konstanten $k_1 \ldots k_5$ ermöglichen eine individuelle Gewichtung der Parameter. Die Default-Einstellungen gibt Cisco für IGRP mit $k_1 = k_3 = 1$ und $k_2 = k_4 = k_5 = 0$ an. Damit fällt die Last- und die Verlässlichkeitsmessung (*reliability*) weg. Bei Ethernet mit 10 Mbit/s mit einer angenommenen Verzögerung von 1 ms ergibt sich mit $k_1 = k_3 = 1$ und $k_2 = k_4 = k_5 = 0$ z.B. $C = \frac{1}{10/10^4} + \frac{1}{10^4} = 10^3 + 10^{-4} \approx 1000$.

Durch die Simulation des Netzes unter Einschluss des dynamischen Routings nach dem Dijkstra-Algorithmus wird eine obere Schranke für den erzielbaren Gewinn durch den Einsatz von PNNI mit dynamisches Routing im Vergleich zum statischen Routing bestimmt. Als Maß hierfür dient die Erhöhung des “Goodputs”, der in den Simulationen anhand der Messung von Verkehrsmatrizen ermittelt wird.

2.1.3 Rufannahme

Die Rufannahme (Call Admission Control, CAC) entscheidet im ATM-Netz über die Annahme oder Ablehnung von Verbindungswünschen. Sie ist damit ein wesentlicher Faktor bezüglich eines für Kunden und Netzbetreiber effektiven Betriebs von ATM-Netzen.

Das Interesse des Kunden ist vor allem die Einhaltung der beim Verbindungsaufbau vereinbarten Dienstgüte (Quality of Service, QoS) als deren wichtigsten Aspekte vom ATM-Forum [SSO96] Zellverlustrate (Cell Loss Ratio), die größtmögliche Zellverzögerung bis zum Empfänger (Maximum Cell Delay) und die Variation der Zellverzögerung (Cell Delay Variation) sowie die Blockierungsrate zu nennen sind. Um eine entsprechende Dienstgüte gewährleisten zu können, muss der Betreiber das ATM-Netz ausreichend dimensionieren und darf es nicht überlasten.

Aus der Sicht des Netzbetreibers soll die Rufannahme eine möglichst hohe Auslastung bei gleichzeitigem Erreichen der notwendigen Dienstgüteparameter ermöglichen. Die hierfür notwendigen mathematischen Berechnungen bei jedem Verbindungsaufbau müssen dabei schnell von den Vermittlungen ausgeführt werden können. Die Rufannahme wird *jedem* Netzknoten auf dem Weg einer neu aufzubauenden Verbindung abschnittsweise durchgeführt. Das heißt, dass in jedem auf dem Weg liegenden Knoten lokal über deren Annahme oder Ablehnung entschieden werden muss. Vor dem Hintergrund, dass in diesem Projekt nur TCP/IP Quellen betrachtet werden, stellt sich die Frage, wie die auf der ATM-Ebene ablaufende Rufannahme mit dem Verbindungsgeschehen auf TCP-Ebene verknüpft werden soll.

Dazu wird hier so verfahren, dass jede TCP Verbindung den dynamischen Aufbau einer ATM Verbindung erfordert. Die zunächst vielleicht etwas willkürlich erscheinende resul-

tierende Rufannahme für TCP-Quellen ist eine durchaus ernst zu nehmende Alternative: Angesichts ernsthafter Überlastprobleme im Internet, wird der Ruf nach einer Rufannahme für TCP-Verbindungen laut [MR99]. Wird diese jetzt verknüpft mit der Rufannahme einer ATM Verbindung, so stellt sich die Frage nach dem Kriterium der Annahme. Eine Standardantwort im ATM Bereich wäre “peak rate allocation” oder “equivalent capacity allocation" [GAN91]. Die erste Möglichkeit schließt jeden Multiplexgewinn aus, ist also beliebig ineffizient. Die zweite Möglichkeit wie eine Vielzahl weiterer [DG96] krankt daran, dass so etwas wie eine von der Quelle vorgegebene Rate im Kontext von TCP-Quellen sinnlos ist, da diese Quellen ihr Sendeverhalten an die vorhandenen Ressourcen anpassen.

Sinnvoll erscheint daher die folgende Überlegung [Mor00]: TCP’s “fast retransmit” Mechanismus produziert bei einem Paket-Verlust einen “time out”, wenn die Fenstergröße weniger als 4 ist. Damit also TCP noch “vernünftig funktioniert”, sollte die aktuelle Fenstergröße immer größer als 4 sein (vgl. Anhang B).

Berücksichtigt man, dass Verluste zu einer Halbierung des Fensters führen, ist ein noch akzeptabler Arbeitspunkt der, wo sich das Fenster zwischen 4 und 8 bewegt, also im Mittel 6 ist. Eine TCP Verbindung benötigt daher eine minimale Senderate von 6 Paketen pro Signalumlauf, wodurch die Anzahl der Verbindungen in einem Link begrenzt wird. Geht man von einer Signalumlaufszeit von mindestens 20 msec z.B. von dem IP-Gate in den USA zu einem deutschen Knoten aus, so ist pro TCP Verbindung für diesen Fall höchstens eine Rate von

$$R = \frac{6 \cdot MTU}{0.02\,sec} \tag{2.2}$$

zu veranschlagen, was bei einer $MTU = 1500$ Bytes auf eine Rate $R = 450$ kbit/s führt. Bei einer Link-Kapazität vom IP-Gate nach Köln von 167 Mbit/s und der gewählten Quellenverteilung (109 Quellen auf dem Link IP-Gate/Köln) ergibt sich dann immer noch, dass jede Quelle $167/109 = 1.53$ Mbit/s zur Verfügung hat. Dieser Sachverhalt ist auch auf den anderen Links gegeben. D.h. dass eine Rufannahme bzw. Ablehnung bei dieser Quellenverteilung nicht notwendig ist. Andererseits sind 1000 Quellen ein Limit, dass

nicht für die zahlreichen Simulationen überschritten werden konnte, da mehr als 1000 Quellen zu einer zu langen Simulationsdauer geführt hätten. Bei einzelnen Simulationen wäre es jedoch bei geeigneter Hardware-Ausstattung (1 GByte RAM, schnelle CPU) ohne weiteres möglich, eine Quellenanzahl von 6000 zu wählen. In diesem Fall greift dann die Rufannahme nach dem oben beschriebenen Prinzip ein.

2.2 Modell eines IP-Routers

2.2.1 Notwendige Kommunikationsschichten

Die für die Simulation von Internetverbindungen notwendigen Kommunikationsprotokolle sind in Bild 2.4 dargestellt. Davon sind die Anwendungsschicht und das “Transfer Control Protocol” (TCP) nur im Terminal Equipment, das heißt im Modell SOURCE zu realisieren. Für den IP-Router sind das Internet Protocol (IP) der ATM Adaptation Layer (AAL) und die eigentliche ATM-Schicht von Bedeutung. Das in Abschnitt 2.2.3 beschriebene Internet Protocol sorgt dabei für den Ende-zu-Ende-Transport von IP-Datagrammen und bedient sich dabei des in Abschnitt 2.2.2 beschriebenen AAL5 Protokolls, das bis zu 64 kByte große Datenblöcke über ATM-Verbindungen transportieren kann.

2.2.2 ATM Adaptation Layer (AAL)

Durch die Begrenzung der Payload von ATM-Zellen auf 48 Byte ist es Aufgabe der Schichten oberhalb der ATM-Schicht, dort vorhandene, größere Datenpakete oder Datenströme entsprechend zu segmentieren und zu reassemblieren. Für den Fall von Datenpaketen mit bis zu 64 kByte Größe wird dieser Dienst von der direkt über der ATM-Schicht liegenden, in Bild 2.5 dargestellten “ATM Adaptation Layer” mit der Protokollvariante 5 (AAL5) angeboten. Die AAL5-Schicht nimmt also, bezogen auf die Darstellung in Bild 2.4, von oben so genannte AAL5 Service Data Units (SDU) entgegen und verpackt die darin enthaltenen Daten in eine entsprechende Anzahl von ATM-Zellen. Diese werden am Empfänger ebenfalls vom AAL5-Protokoll wieder zu einer AAL5 SDU zusammen-

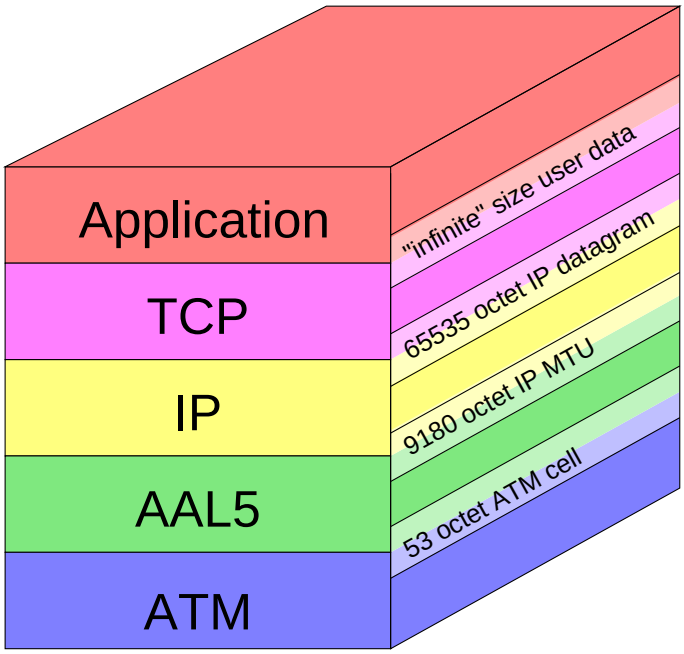


Abbildung 2.4: Simulierte Kommunikationsschichten für Internetverkehr mit maximalen Datagramm-Größen.

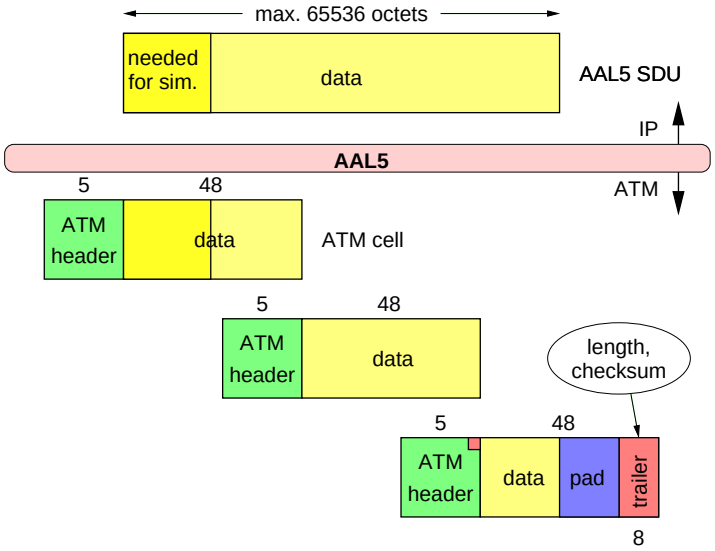


Abbildung 2.5: Aufteilung eines AAL5 Rahmens auf mehrere ATM-Zellen.

gefügt und an die höheren Schichten weitergeleitet, falls keine Übertragungsfehler aufgetreten sind. Sind durch die CRC-Prüfung aufgrund der in der letzten Zelle einer AAL5 SDU mitgeschickten CRC-Prüfsumme oder aufgrund der ebenfalls dort enthaltenen Längenangabe Fehler festgestellt worden, wird die gesamte SDU verworfen.

Das AAL5-Protokoll ist im Simulator für die Übertragung von IP-Paketen notwendig und

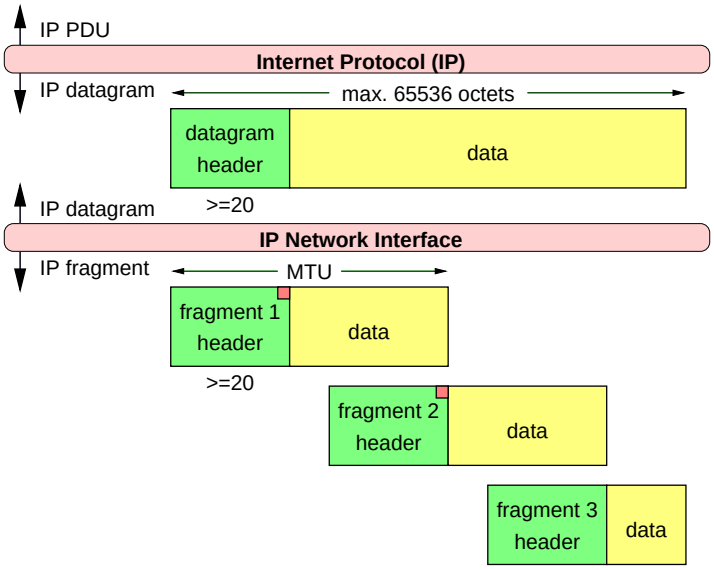


Abbildung 2.6: Aufteilung eines IP-Datagramms auf mehrere IP-Fragmente.

muss sowohl im TE-Modell als auch in den Switches implementiert sein, denn letztere greifen im Zusammenhang mit den für UBR notwendigen “Early Packet Discard” Algorithmen auf Teile der Funktionalität des AAL5 zurück.

2.2.3 Internet Protocol (IP)

Die in Bild 2.4 dargestellte IP-Schicht transportiert von höheren Ebenen (zum Beispiel TCP) bereitgestellte IP-SDUs von Endgerät zu Endgerät. Das Hinzufügen von Headern und die Segmentierung und Reassemblierung innerhalb der IP-Schicht soll hier betrachtet werden, wie in Bild 2.6 gezeigt. So wird den SDUs ein IP-Kopffeld vorangestellt und das dadurch entstehende IP-Datagramm[1] in mehrere Fragmente aufgeteilt, falls auf dem Weg liegende physikalische Medien nur kleinere Pakete am Stück transportieren können. Dazu teilen die Endgeräte oder die dazwischen liegenden Router dem Sender ihre so genannte “Maximum Transfer Unit” (MTU) mit, die dann als maximale Größe[2] für die Fragmente benutzt wird. Der Fragmentheader unterscheidet sich dabei nur um zwei Bits von dem ursprünglichen IP-Header, von denen eines im letzten Fragment zurückgesetzt wird, und so das Ende der PDU anzeigt.

[1] Ein IP-Datagramm kann bis zu $65536 = 2^{16}$ Oktets lang sein.
[2] 1500 Oktets bei Ethernet, 4200 Oktets bei FDDI.

Bei dem Vergleich der maximalen Größen von IP-Datagramm und AAL5-SDU fällt auf, dass sie identisch sind und so eine Segmentierung und Reassemblierung im IP-Layer innerhalb eines reinen ATM-Netzes nicht notwendig ist. Sobald jedoch an das ATM-Netzwerk ein anderes physikalischen Netz angeschlossen wird, muss entweder ein dazwischen liegender Router oder der eigentliche Sender beides beherrschen. Um dem Einfluss dieser Fragmentierung auf zu untersuchende “Early Packet Discard” Algorithmen untersuchen zu können, ist sie im Simulator vorgesehen, kann aber durch eine entsprechende Wahl der MTU gegebenenfalls abgeschaltet werden.

2.2.4 Gegenwärtige Benutzung von Routing im B-WiN

Derzeit wird Video- und Telefonverkehr als CBR-Dienst im B-WiN integriert. Dem IP-Verkehr werden ebenfalls feste Bandbreiten zugewiesen, beispielsweise 20 MBit/s von Hamburg nach Hannover. Alle TCP/IP-Verbindungen zwischen diesen beiden Knoten müssen sich die spezifizierte Bandbreite teilen.

Die IP-Router der einzelnen Knoten des B-WiN sind untereinander nicht voll vermascht, sondern es erfolgt ein Hop-by-Hop-Routing[3]. Entsprechend werden die IP-Pakete über das ATM-Netz von IP-Router zu IP-Router geschickt, da sich nur benachbarte IP-Router kennen. Es werden, außer in Ausnahmefällen dauerhaft hoher Last, keine direkten ATM-Verbindungen von z.B. Hamburg nach München geschaltet.

Das Border Gateway Protokoll 4 (BGP-4) wird zum Austausch von Routing-Informationen zwischen benachbarten Routern benutzt[4] . Da in diesem Projekt Ausfälle von Routern nicht berücksichtigt werden und der Einfluss der zum Routing versandten Daten aufgrund des geringen Datenvolumens vernachlässigt werden kann[5], wird das BGP-4 Protokoll im Simulator nicht implementiert. Es ist zur Simulation des gegenwärtig benutzten statischen Routings ausreichend, bei Beginn einer Simulation die Routing-Tabellen mit entsprechenden Adressen zu laden. Dementsprechend hat der IP-Router hier nur die Aufgabe, den entsprechenden Ausgangs-Port für das Paket mit einer bestimmten Ziel-

[3] ATM ermöglicht im Prinzip eine virtuelle Vollvermaschung aller Router im B-WiN.
[4] siehe auch RFC 1654, 1771, 1773 und 1774
[5] vergleiche RFC 1774

Adresse zu finden, und das Paket dann dort auszugeben. Am Ausgang eines jeden Ports befindet sich dann wieder eine Warteschlange. Aus dem Gesagten ergibt sich, dass das Modell für den IP Router auf dem Abstraktionsniveau von Bild 2.1 mit dem dort gezeigten Modell eines ATM Knotens identisch ist, dass aber die Verfeinerung in Anlehnung an Bild 2.2 einfach wegfällt, da die unterschiedlichen Prioritätsniveaus auf IP Ebene nicht benötigt werden.

2.3 Realisierung der Regelung für ABR

Die “Available Bit Rate Service Category” (ABR) wurde entwickelt, um die von “normalen” ATM-Diensten (CBR, VBR) momentan auf einzelnen Links nicht genutzte Bandbreite verwendbar zu machen. Dafür ist es von Seiten des Netzes notwendig, den über ein bestimmtes Link sendenden ABR-Quellen die momentan für sie noch verfügbare Datenrate direkt oder indirekt mitzuteilen, wie es in Abschnitt 2.3.1 erläutert wird. Ein wesentliches Problem bereiten dabei die durch Leitungen, Puffer und Bearbeitung in den Netzknoten hervorgerufenen Verzögerungen der Steuerinformationen, wodurch zwischenzeitlich auftretende Überlasten mit entsprechend großen Puffern abgefangen werden müssen. Um diese Puffer möglichst klein halten zu können und die Linkauslastungen zu maximieren, werden in den Netzknoten mehr oder weniger komplexe Algorithmen implementiert, die letztendlich die Quellen steuern. Dazu sind einige grundsätzliche Überlegungen und eine Auswahl der derzeit im Simulator implementierten Algorithmen in Abschnitt 2.3.2 dargestellt.

2.3.1 Rückkopplung durch RM-Zellen

Grundsätzlich sind ABR-Verbindungen, wie alle anderen ATM-Verbindungen auch, bidirektional. Der Übersicht wegen, soll hier allerdings nur der Datenfluss in eine Richtung betrachtet werden, wie er in Bild 2.7 von A nach B dargestellt ist. Die Steuerinformationen werden von den zwischen A und B liegenden ATM-Switches in so genannten RM-

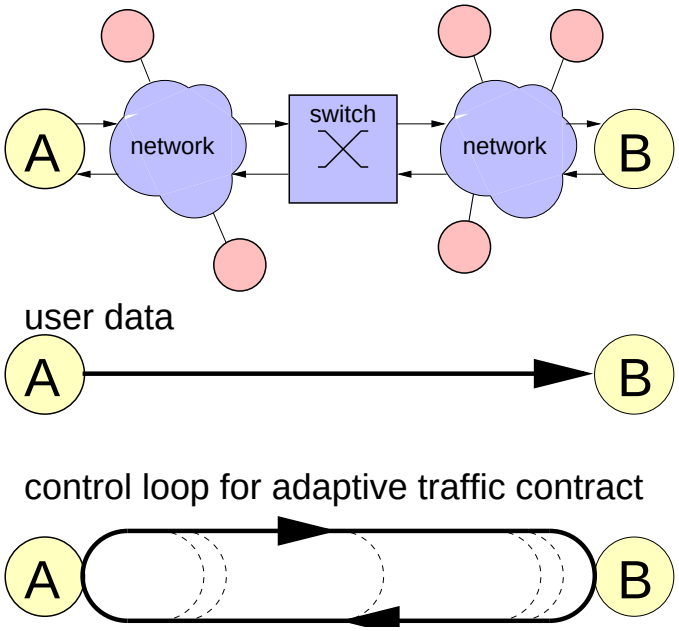


Abbildung 2.7: ABR-Regelschleife für eine Verbindung von A nach B.

Zellen (Ressource Management) eingetragen, die von A ausgehend über B wieder nach A umlaufen.

Wird jetzt auf einem der zwischen A und B liegenden Links die für ABR zur Verfügung stehende Datenrate – zum Beispiel durch eine zusätzlich angenommene CBR-Verbindung – kleiner, wird dies der Quelle A mitgeteilt, die darauf unverzüglich reagieren muss. Trotzdem dauert es eine Weile, bis das Resultat der vom Switch eingetragenen Steuerinformation, nämlich die Reduktion der von A verwendeten Datenrate, bei A ankommt. Die in der Zwischenzeit im Switch eintreffenden ABR-Zellen dürfen jedoch nicht verworfen werden (ABR erlaubt zwar Verzögerungen aber keine Verluste), sondern müssen in entsprechend großen Puffern zwischengespeichert werden.

Die RM-Zellen werden von der Quelle nach festgelegten Regeln in den Datenstrom eingefügt und heißen FRM-Zellen (Forward) auf dem Weg von A nach B. In der anderen Richtung werden sie BRM-Zellen (Backward) genannt und durch einen entsprechenden Eintrag in der Payload auch als solche gekennzeichnet. Dies ist notwendig, weil die ABR-Verbindung eigentlich bidirektional ist, und daher auch B an A Daten und dazugehörige RM-Zellen schickt. Um die BRM-Zellen von den in derselben Richtung fließenden ABR-Daten, beziehungsweise deren Verzögerungen durch ABR-Puffer unabhängig zu machen,

sind für die BRM-Zellen separate Puffer in den Vermittlungen notwendig. Ohne diese wäre es “unfreundlichen” TEs möglich, die gemeinsamen Puffer zum Überlaufen zu bringen.

Vom ATM-Forum sind die drei folgenden Rückkopplungsstrategien für den Transport von Steuerinformation vom Netz zum TE festgelegt worden:

1. Jeder Zellkopf einer ATM-Zelle enthält ein EFCI-Feld (**Explicit Forward Congestion Indication**). Die Vermittlungen setzen dieses Bit in den Datenzellen, wenn bei ihnen ein Datenstau vorliegt oder erwartet wird. Der Empfänger B liest dieses Bit und sendet ein Feedback in Form einer RM-Zelle an die Quelle A zurück.

2. Bei der zweiten Möglichkeit werden kontinuierlich RM-Zellen von der Quelle A verschickt. Die RM-Zellen besitzen zwei Bits in ihrer Payload, nämlich das CI-Bit (Congestion Indikation) und das NI-Bit (No Increase), mit denen die Vermittlungen auf einen Stau beziehungsweise Engpass reagieren können. Dies wird auch als **Relative Rate Marking** bezeichnet.

3. Bei der dritten Möglichkeit wird das ER-Feld (Explizite Rate) aus der Payload der RM-Zellen benutzt, die auch hier kontinuierlich von der Quelle A gesendet werden. Die Vermittlung hat nun allerdings die Möglichkeit, über dieses Feld der Quelle genau mitzuteilen, welche Datenrate sie verarbeiten kann. Deshalb wird diese Art der Steuerung auch als **Explicit Rate Marking** bezeichnet.

2.3.2 Available Bit Rate (ABR) Algorithmen

Einen wesentlichen Einfluss auf die notwendige ABR-Puffergröße haben die in den Switches verwendeten Algorithmen zur Überlasterkennung und Quellensteuerung. Von den beiden im ATM-Simulator implementierten Algorithmen soll nur der leistungsfähigste, der “Explicit Rate Indication for Congestion Avoidance” Algorithmus (ERICA) kurz dargestellt werden. Der ERICA-Algorithmus zählt zu den Verfahren mit Feedback in Form einer explizit vom Netzwerk angegeben Rate, deren wesentliche Vorteile die folgenden sind:

1. Sie konvergieren schneller gegen ihren Arbeitspunkt.

2. Sie sind robuster gegenüber RM-Zellverlusten.

3. Das Schwingen der Raten wird weitgehend verhindert.

Der ERICA-Algorithmus ist in [Kal97, AX96, KK96] näher beschrieben worden und soll hier nur kurz skizziert werden. Er ist ein rein ratenbasiertes Verfahren, bei dem die momentan für ABR verfügbare Rate gemessen und daraus explizite Raten für die einzelnen ABR-Verbindungen berechnet werden.

Zunächst wird der so genannte Lastfaktor aus der momentanen ABR-Eingangsrate, der Zielauslastung[6] U und der momentan für ABR zur Verfügung stehenden Link-Kapazität berechnet:

$$z = \frac{ABR_{Eingangsrate}}{ABR_{Zielrate}} = \frac{ABR_{Eingangsrate}}{U \cdot ABR_{Capacity}}. \tag{2.3}$$

Der Lastfaktor kann dabei als ein Maß für die Last in der augenblicklichen Situation angesehen werden. Werte oberhalb von eins bedeuten Überlast und Werte unterhalb von eins Unterlast. Die optimale Situation liegt bei einem Lastfaktor von eins.

Die ABR-Eingangsrate kann dabei von der Vermittlung entweder aus dem Wert für die "Current Cell Rate" in der RM-Zelle oder durch eine direkte Messung bestimmt werden. Auch die für ABR zur Verfügung stehende Bandbreite $ABR_{Capacity}$ wird durch Messung bestimmt. Um diese Datenraten zu messen, können unterschiedliche Wege beschritten werden.

Messung der Datenrate Eine Möglichkeit ist, die Datenraten über ein festes Mittellungsintervall zu messen und im Anschluss an die Messung den Lastfaktor zu bestimmen. Hierbei hat die Länge des Messintervalls folgenden Einfluss auf die Eigenschaften der Regelung:

- Lange Intervalle verbessern die Genauigkeit der Messung, dafür wird jedoch die Anpassungsgeschwindigkeit verlangsamt .

[6] Typische Werte für die Zielauslastung liegen zwischen 90% und 95%.

- Kurze Intervalle verbessern die Anpassungsgeschwindigkeit, dafür werden jedoch die gemessenen Werte ungenauer.

Eine andere Möglichkeit ist, für diese Messung eine laufende Summe der folgenden Form zu verwenden.

$$Last_{neu} = Last_{alt}(1-\alpha) + \alpha Mess_{Var} \tag{2.4}$$

Hierbei ist $Mess_{Var}$ gleich eins, wenn eine Zelle gesendet wurde, ansonsten gleich Null. Der Faktor α gibt die Gewichtung des neuen Messwertes an. Ein kleiner Wert von α ergibt sehr langsame Anpassung, während ein großer Wert eine schnellere Anpassung, aber einen ungenaueren Messwert ergibt. Die jeweilige Datenrate ergibt sich dann aus:

$$Rate = Linkbitrate \cdot Last. \tag{2.5}$$

Der Vorteil dieser Art der Messung ist, dass eine Veränderung der Datenrate unabhängig von einem Fenster sofort wirksam wird.

Aufteilung der Bandbreite Zusätzlich zum Lastfaktor z wird für jedes VC ein *FairShare* berechnet. Dies ergibt sich mit der Anzahl der aktiven Quellen N wie folgt:

$$FairShare = \frac{ABR_{Zielrate}}{N} \tag{2.6}$$

Die Vermittlung erlaubt den Quellen, deren Rate kleiner als die berechnete faire Aufteilung ist, die Rate zu erhöhen. Falls eine Quelle ihren fairen Anteil nicht gänzlich nutzt, wird der Rest von der Vermittlung den Quellen zugeteilt, die diesen nutzen können. Hierfür wird von dieser das so genannte *VCShare* berechnet. Dieses ergibt sich aus dem Quotienten der aktuellen Rate der Quelle *CCR* und dem Lastfaktor z:

$$VCShare = \frac{CCR}{z}. \tag{2.7}$$

Die Kombination aus beiden berechneten Werten wird zur Bestimmung der einzutragenden expliziten Rate (ER) benutzt. Die berechnete explizite Rate ergibt sich dabei zu:

$$ER_{berechnet} = \max(FairShare, VCShare). \tag{2.8}$$

Dabei versucht das *VCShare*, das System in einen effizienten Zustand zu bringen, der nicht notwendigerweise fair sein muss, während das *FairShare* versucht, die Fairness zu sichern.

Eine generelle Konvergenz im Sinne einer Max-Min-Fairness ist so allerdings noch nicht gegeben. Nach Shivkumar Kalayanaraman [Kal97] konvergiert der Algorithmus in folgenden Fällen nicht gegen das Max-Min-Kriterium:

1. Falls der Lastfaktor eins wird,
2. wenn einige Quellen in vorangehenden Vermittlungen begrenzt werden und
3. das CCR für alle übrigen Quellen größer als die faire Aufteilung ist.

Um auch in diesen Fällen Max-Min-Fairness zu erreichen, wurde der ERICA-Algorithmus etwas erweitert. Die letzte Bandbreitenzuteilung, die für eine Quelle gemacht wurde, wird dabei gespeichert. Die maximale Zuteilung für eine Quelle (MaxAllocPrevious) wird gespeichert. Falls der Lastfaktor z kleiner als 1 ist, wird im Gegensatz zu Gleichung (2.8), die somit nur für $z > 1 + \delta$ gilt, die maximale Zuteilung aus dem letzten Intervall (Z_{max}) mit in die Formel aufgenommen. Auf diese Art und Weise ergibt sich für $z < 1$ folgende Formel.

$$ER_{berechnet} = \max(FairShare, VCShare, Z_{max}) \quad ,\forall z \leq 1 + \delta. \tag{2.9}$$

Das δ ist dabei ein kleiner Wert. Er bewirkt, dass kleine Schwankungen nicht sofort Auswirkungen auf die Raten haben.

Da diese berechnete explizite Rate die ABR-Zielrate nicht überschreiten darf, wird folgendes Minimum gebildet:

$$ER_{berechnet} = \min(ER_{berechnet}, ABR_{Zielrate}). \tag{2.10}$$

Zur Vermeidung von unnötigen Schwankungen in den Raten der Quellen sollte im Anschluss an die Gleichung (2.10) die **Fairshare First** Option durchgeführt werden [Kal97]. Hierdurch wird ein Überschwingen der Raten vermieden. Dieses Überschwingen kann z. B. passieren, wenn verschiedene RM-Zellen asynchron bei einer Vermittlung ankommen. Nachdem das ER berechnet wurde, wird folgender Schritt durchgeführt: Wenn ($CCR < FairShare$) und ($ER_{berechnet} \geq FairShare$) gilt wird $ER_{berechnet} = FairShare$ gesetzt.

In das ER-Feld der Forward RM-Zelle wird schließlich folgendes eingetragen:

$$ER_{in,RM-Zelle} = \min(ER_{berechnet}, ER_{in,RM-Zelle}). \tag{2.11}$$

Die Implementierung des ERICA-Algorithmus im ATM-Simulator entspricht der vorangegangenen Beschreibung.

2.3.3 Verifikation und Veranschaulichungen

Bereits während der Implementierung der ABR-Algorithmen wurden mit einem sehr einfachen Szenario erste Simulationen durchgeführt, um die korrekte Funktion des Simulators zu verifizieren. Dazu wurden die mit Hilfe der Simulation gewonnen Messwerte mit denen aus bekannten Veröffentlichungen auf Übereinstimmung geprüft. Der ERICA-Algorithmus wurde anhand einiger Untersuchungen von Shivkumar Kalayanaraman [Kal97] verifiziert und brachte insgesamt eine wesentlich bessere Performanz als EFCI. Betrachtet wurden dabei zwei konkurrierende ABR-Quellen einmal ohne und einmal mit einer On/Off-Quelle als Hintergrundlast. Ein Schwingen wie bei den EFCI-Verfahren wurde dabei in beiden Fällen nicht beobachtet.

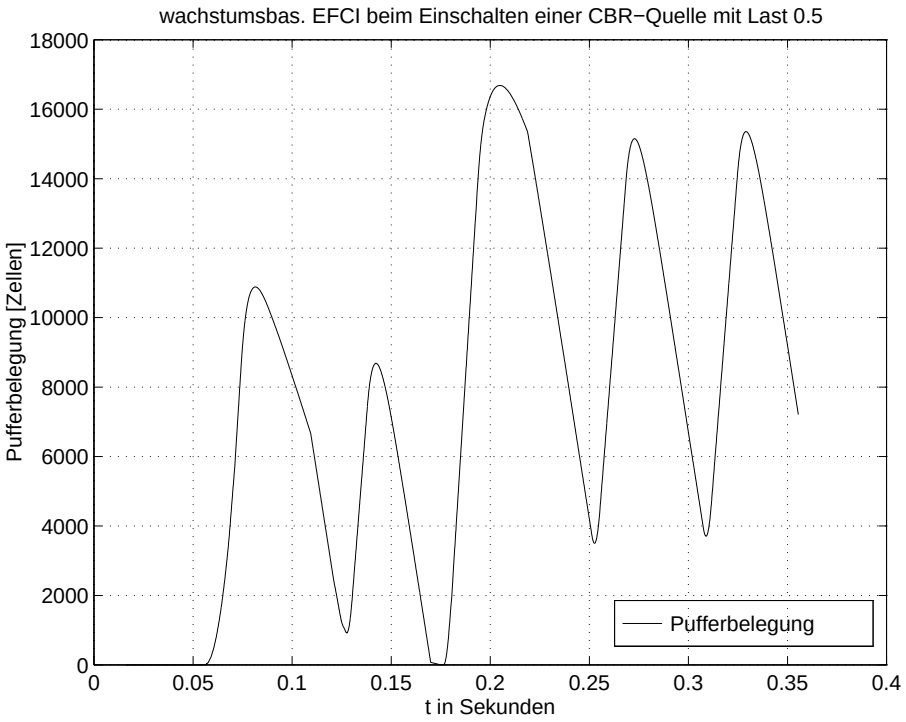


Abbildung 2.8: EFCI: Pufferbelegung beim Einschalten einer CBR-Quelle.

Schwach dynamisches Netzwerk mit seltenem Verbindungsauf- und Abbau

Um einen ersten Eindruck von EFCI und ERICA bezüglich ihres Verhaltens bei Änderungen der verfügbaren Datenrate im Netzwerk zu bekommen, wurden beide mit einer kurzzeitig aktiven CBR-Quelle (77.5 MBit/s, halbe Linkrate) als “Störung” simuliert. Jeweils drei ABR-Quellen teilen sich dabei die von der CBR-Verbindung nicht genutzte Bandbreite auf der 1600 km (3000 Zellen) langen Leitung zwischen den NTs. Bei allen dreien wurde die “Minimum Cell Rate” (*MCR*) auf 64 kBit/s und die “Peak Cell Rate” (*PCR*) auf 155 Mbit/s festgelegt. Die Ziellinkauslastung ist für den ERICA-Algorithmus auf 90% und die Schwellwerte für das wachstumsbasierte EFCI sind $Q_H = 100$ und $Q_L = 50$.

Bei der Betrachtung der in den Bildern 2.8 und 2.9 dargestellten Simulationsergebnisse lässt sich ein großer Unterschied bei der maximal benötigten Anzahl von Pufferplätzen erkennen. Die binäre EFCI-Regelung benötigt wesentlich mehr Zeit bzw. Umläufe von RM-Zellen, da die Regelinformation nur bitweise und nicht innerhalb einer RM-Zelle als ganze Zahl übertragen werden kann.

Bei der hier nicht dargestellten Linkauslastung treten sowohl bei ERICA in dem Moment

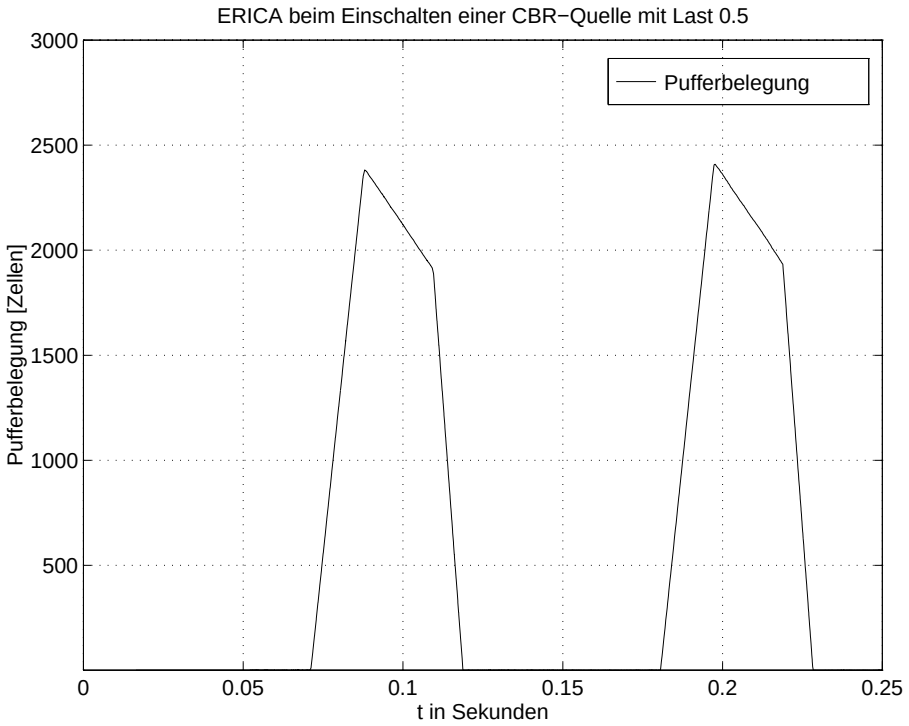


Abbildung 2.9: ERICA: Pufferbelegung beim Einschalten einer CBR-Quelle.

Einbrüche auf, wo die CBR-Verbindung angemeldet wird, beziehungsweise zu senden beginnt. Die Einbrüche werden durch die Verzögerung in der Regelschleife hervorgerufen, die bei EFCI durch die in diesem Beispiel gut gefüllten Puffer ausgeglichen werden. Über die Zeit gemittelt wird hier mit dem ERICA-Algorithmus eine Linkauslastung von 93% und mit dem wachstumsbasierten EFCI sogar eine Linkauslastung von 97% erreicht. Diese Werte sind jedoch nur als beispielhaft anzusehen, da sie stark von dem betrachteten Szenario und der Parametrisierung von Quellen und Regelung abhängen.

Kapitel 3

Quellenmodelle

In diesem Kapitel wird die Modellierung von Quellen behandelt, die zur Generierung von Verkehr für die Simulation des B-WiNs benutzt werden können. Selbstähnliche Quellen ohne Flusskontrolle (Abschnitt 3.1) und TCP-Quellen, im besonderen eine TCP On-Off Quelle (Abschnitt 3.2). Das Ziel der Modellierung ist es, die Messungen aus [GGS97] durch die Simulation so weit nachzubilden, dass es möglich ist, Untersuchungen der QoS mit ABR, UBR bzw. CBR und dynamischem Routing durchzuführen. Um dies sinnvoll zu ermöglichen, ist es notwendig, die Quellen auf Benutzer-Ebene zu modellieren.

3.1 Quellen ohne Flusskontrolle

Quellen zur Generierung von selbstähnlichem Verkehr ohne Flusskontrolle sind dazu geeignet, aggregierten Hintergrundverkehr auf ATM-Ebene zu modellieren. Dabei kann ein Simulationsmodul einer Quelle so parametrisiert werden, dass es N Quellen repräsentiert. Der Vorteil hierbei ist, dass die statistischen Parameter eines von vielen Quellen stammenden, aggregierten Verkehrs in einem Modul von vorneherein eingestellt werden können.

3.1.1 N-Burst

Diese Quelle wurde schon in [GGS97] sehr ausführlich beschrieben, daher hier nur das Wesentliche: Die N-Burst Quelle wurde in [Sch97] und [Gre97b] als ein Modell für

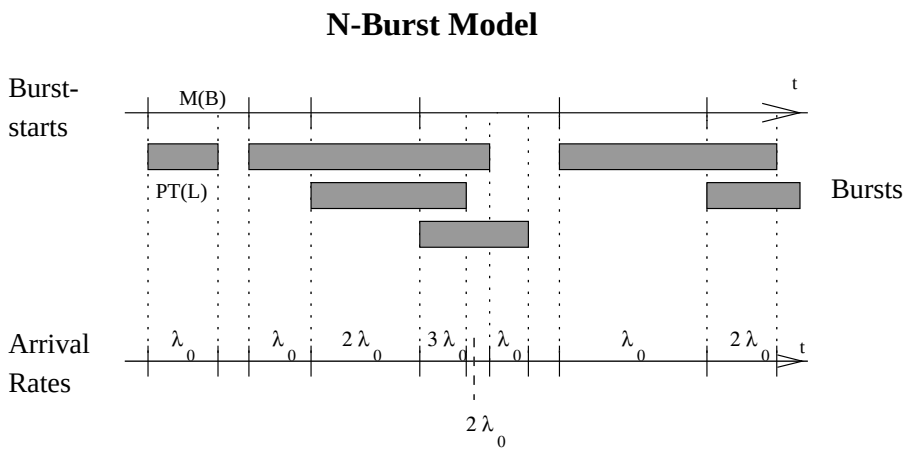


Abbildung 3.1: Überlagerunsgsschema der N-Burst Quelle.

ATM Netzwerk-Verkehr eingeführt. Die Quelle basiert auf dem Konzept der Superposition von On/Off-Quellen mit power-tail verteilter Burst-Länge (vergl. Gl. A.12), wie in Bild 3.1 dargestellt. Die maximale Anzahl der simultan aktiven Quellen ist auf N begrenzt und die Zwischenankunftszeiten sind negativ exponentiell verteilt. Die einzelnen Quellen haben alle die gleiche mittlere Rate λ_0 bzw mittlere Zwischenankunftszeit $1/\lambda_0$. Die Burst-Länge wird mit der “Truncated Power-Tail” (TPT) Verteilung modelliert, die die power-tail Eigenschaft, durch eine Summe von negativ exponentiellen Funktionen annähert. Für den Fall der Überlagerung von unendlich vielen negativ exponentiellen Funktionen (“truncation level” ist unendlich) wird die power-tail Eigenschaft exakt erfüllt (siehe Bild 3.2). Es wurde hier die “truncated power-tail” Verteilung, und nicht eine exakte power-tail Verteilung gewählt, da durch das Abschneiden (engl. truncation) eine sinnvolle obere Grenze für die Länge der Bursts eingestellt werden kann, wie sie auch in real existierenden Netzen existiert.

3.1.2 SupFRP

Der Name der Quelle kommt aus dem engl. “Superposition of Fractal Renewal Processes”. Bei dieser Quelle werden N fraktale Erneuerungsprozesse überlagert [RN97]. Dabei sind die Zwischenankunftszeiten der einzelnen Erneuerungsprozesse power-tail verteilt (siehe Gl. A.12). Für die Einzelquellen einer SupFRP sind die Zeiten t_i zwischen dem Versenden von Paketen voneinander unabhängig und werden durch eine einfache Transfor-

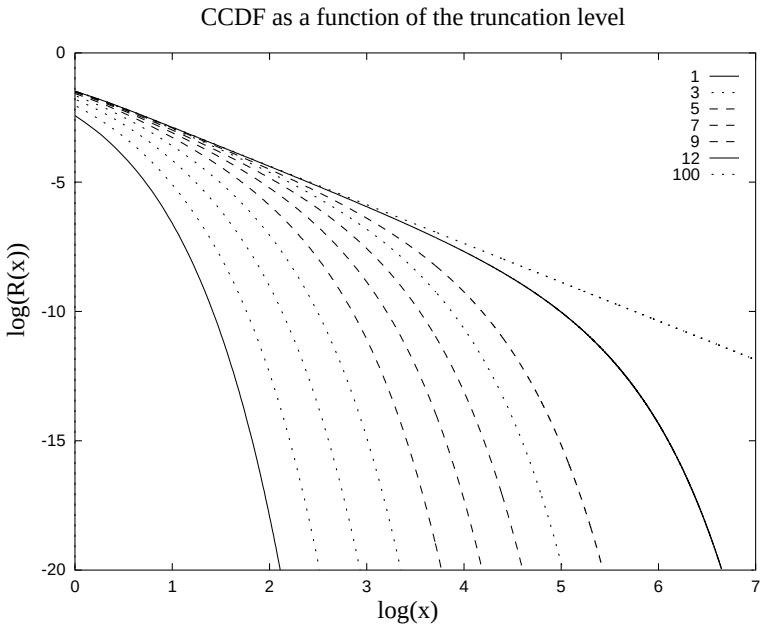


Abbildung 3.2: Komplementäre Verteilungsfunktion der TPT-Verteilung.

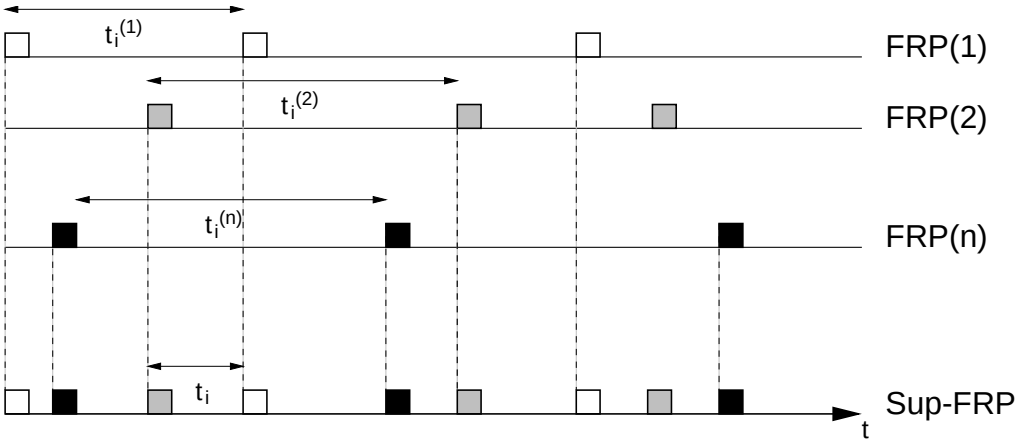


Abbildung 3.3: Zwischenankunftszeiten der SupFRP.

mation einer gleichverteilten Zufallsvariablen $0 < U < 1$ gewonnen. Auf On/Off-Zustände wird hier verzichtet, da die power-tail Verteilung der Zwischenankunftszeiten schon mit nicht zu vernachlässigender Wahrscheinlichkeit sehr große Zwischenankunftszeiten hervorruft.

3.2 TCP On/Off Quelle

Diese Quelle modelliert das Benutzerverhalten von einzelnen Internet Teilnehmern bzw. LAN/WAN Benutzern bei der Interaktion mit z.B. dem Web-Browser oder dem FTP-Client: Der Benutzer fordert eine Datei an; daraufhin wird eine Verbindung aufgebaut und die Datei übertragen. Nach der vollständigen Übertragung wird das Dokument bearbeitet (evtl. ausgepackt, dargestellt, gelesen und evtl. eine andere Tätigkeit begonnen); während der Bearbeitung tritt keine Netzaktivität auf. Dieses Verhalten wird durch die On- und Off-Zustände der TCP-Quelle modelliert.

In [CB97] wurde festgestellt, dass die Dateilängenverteilung der übertragenen TCP-Ströme und die Off-Zeit die Power-Tail Eigenschaft (Gl. A.12) erfüllt. Es wurde jedoch ebenfalls ermittelt, dass die Dateilängenverteilung einen wesentlich kleineren Tail-Index α als die Off-Zeit besitzt ($\alpha_{File} \approx 1.1$ bzw. $\alpha_{Off} \approx 1.5$, [CB97]). Das bedeutet, dass die Dateilängenverteilung die Eigenschaft der Selbstähnlichkeit zweiter Ordnung dominiert. Daher wird, um einen freien Parameter zu eliminieren, der die Messungen nicht wesentlich beeinflusst, im Weiteren für die Off-Zeit eine negativ exponentielle Verteilung benutzt. Als Verteilung der Dateilänge wird die TPT-Verteilung aus [GGS97] benutzt (siehe auch Abschnitt 3.1.1). Der “truncation level” wird auf 12 gesetzt, so dass die power-tail Eigenschaft für etwa fünf Größenordnungen gilt. Für den On- und Off-Status der Quelle sind jedoch prinzipiell die Verteilungen “deterministisch”, “geometrisch” und “negativ exponentiell” ebenfalls einsetzbar, da die Umschaltung der Verteilung nur das Verändern eines Parameters der Quelle erfordert.

Da TCP im Prinzip fast beliebig große Raten der einzelnen Quellen zulassen würde, wurde die minimale Zwischenankunftszeit der TCP-Segmente der Quellen begrenzt. Das entspricht etwa einer Modellierung einer begrenzten Netzwerkschnittstelle, wie z.B. eine Netzwerkkarte mit 10 MBit/s, die auch nicht in der Lage ist, die 155 MBit/s eines ATM-Links zu füllen. Ohne diese Limitierung würde außerdem die Selbstähnlichkeit des Gesamtverkehrs abnehmen, da die Übertragung langer Dateien dann nicht zu langen Übertragungszeiten, sondern ähnlich schnell wie kurze Dateien erledigt werden könnte [PKC96].

Zur Übersicht hier noch einmal die charakteristischen Eigenschaften der TCP On-Off

Quelle, so wie sie im folgenden eingesetzt wird:

- Off-Zeit: exponentiell verteilt
- On-Zeit: Dateilänge ist Truncated Power-Tail (TPT) verteilt [GGS97]
- Die Quelle ist “greedy” bis zu einer minimalen Zwischenankunftszeit für TCP-Segmente
- Das weitere zeitliche Verhalten wird vom TCP-Protokoll bestimmt:
  - “slow-start”
  - “congestion avoidance”
  - “sliding window”
  - “fast retransmit”
  - “fast recovery”
  - Schätzung der “Round Trip Time” (RTT) mit dem Karn-Algorithmus

Diese Implementierung entspricht dem TCP Reno, dessen Überlastkontrolle im Anhang B noch einmal kurz erläutert wird.

In Bild 3.4 ist die schematische Darstellung der TCP On/Off-Quelle dargestellt. Auf der linken Seite ist die TCP-Quelle und auf der rechten Seite die TCP-Senke zu sehen. In der Mitte befindet sich die Warteschlange und der Server, an denen die statistischen Messungen mit Hilfe der Blöcke “MyStatistics” und “IAT_Stat” durchgeführt werden (Zählprozeß- bzw. Zwischenankunftszeit-Messung). Dabei entspricht der Server einem “Flaschenhals”, der die Raten der N TCP Quellen begrenzt.

Anders als bei der N-Burst und der SupFRP Quelle wird hier jede einzelne Quelle modelliert, die individuell durch TCP geregelt wird. Erst nach der Regelung durch TCP wird durch Superposition ein aggregierter Verkehr aus N TCP On/Off-Quellen gebildet. Die Parameter des Moduls “MapGr” bestimmen hier, wieviele Replikationen der TCP-Quelle bzw. TCP-Senke für die Superposition generiert werden, um nicht nur eine Quelle, wie in Bild 3.4 sichtbar, sondern N Quellen in gleicher Form zu verschalten. Durch diese sog.

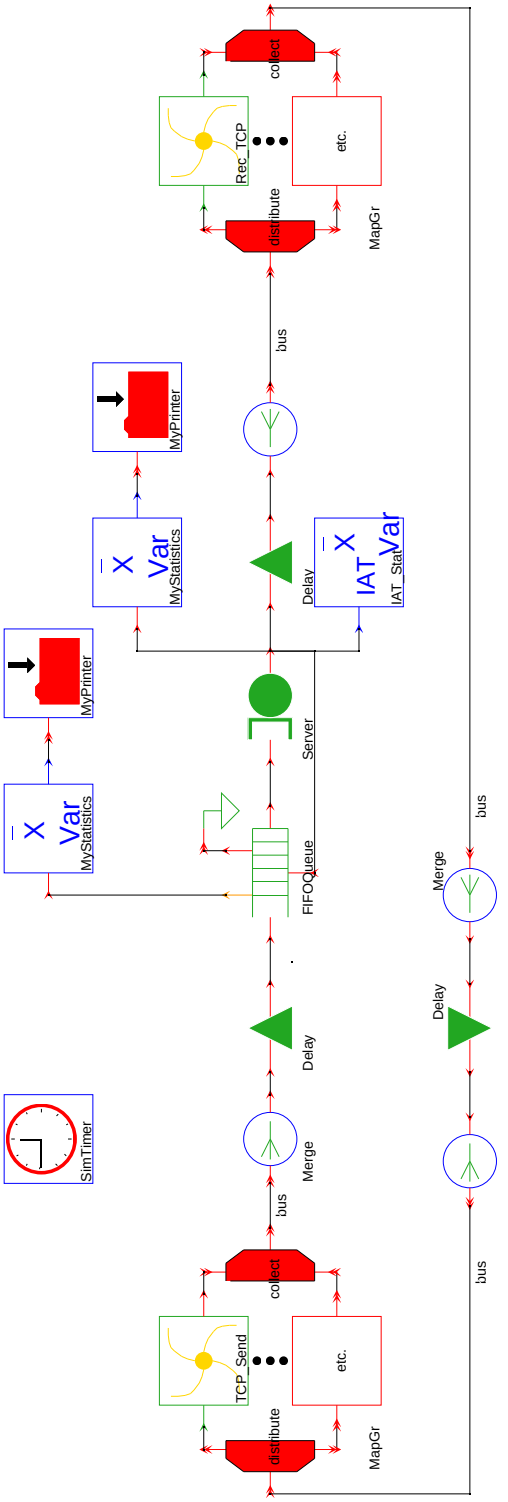


Abbildung 3.4: Schema der Aggregierung von TCP On/Off-Quellen.

"Higher Order Functions (HOF)" ist es möglich, die Überlagerung von N TCP On/Off-Quellen mit dem einfach zu überblickenden Szenario aus Bild 3.4 darzustellen, wobei

die Komplexität der Verschaltung im Hintergrund gehalten wird und mit Parametern zu steuern ist.

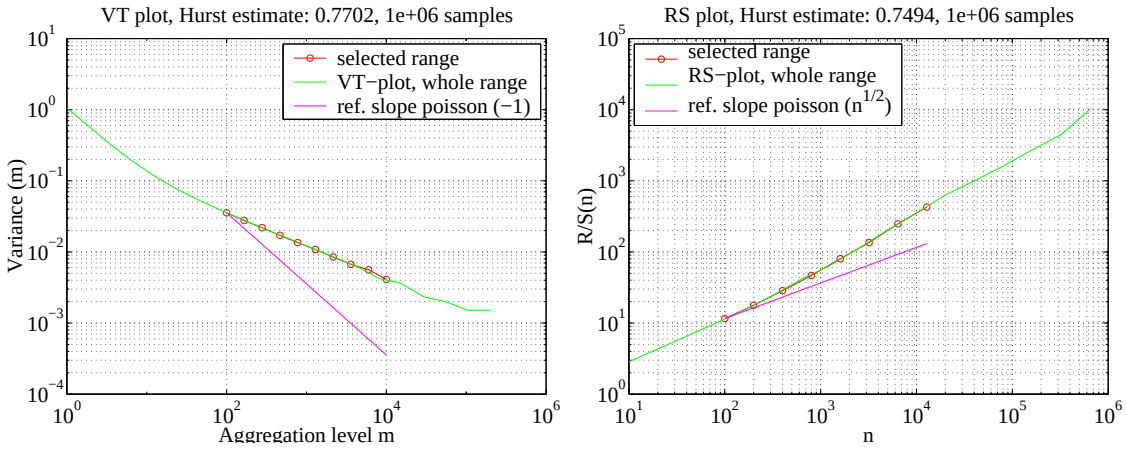
3.3 Validierung der Quellenmodelle

In diesem Abschnitt soll zum Einen gezeigt werden, dass die in Abschnitt 3 vorgestellten Quellen implementiert wurden und so funktionieren, wie es dem jeweiligen Modell entspricht (Abschnitt 3.3.1 und 3.3.2). Außerdem wird in Abschnitt 3.4 die Parametrisierung der selbstähnlichen TCP On/Off-Quelle auf die Messungen aus [GGS97] mit einem vereinfachten Modell vorgestellt. Wir verzichten hier auf eine triviale Validierung der konventionellen Quellen ohne Flusskontrolle, die in Abschnitt 3.1 eingeführt wurden.

3.3.1 Quellen ohne Flusskontrolle

In diesem Abschnitt wird gezeigt, dass mit der SupFRP-Quelle und der N-Burst Quelle Verkehr erzeugt werden kann, der die Eigenschaft der Selbstähnlichkeit zweiter Ordnung (siehe Anhang A.3) besitzt. Der “Variance-Time Plot” (VT-Plot, siehe Anhang A.6.1) und der “ReScaled adjusted range Plot” (R/S-Plot, siehe Anhang A.6.2) werden hier benutzt, um den Hurst Parameter H, den Grad der Selbstähnlichkeit, zu schätzen. Bei der Schätzung werden zwei verschiedene Zufallsvariablen zugrundegelegt: der Zählprozeß und die Zwischenankunftszeiten. Die Einstellungen für die SupFRP und die N-Burst Quelle wurden wie folgt gewählt:

- Einstellungen:
  - Anzahl der zu überlagernden Quellen: $N = 20$ (20-FRP bzw. 20-Burst)
  - vorgegebener Hurst Parameter: $H = 0.8$
  - mittlere Zwischenankunftszeit: $\tau = 1.0$
  - Fenstergröße des Zählprozesses: $w = 10^3$
  - gemessen: 10^6 Zwischenankunftszeiten und ca. 10^5 Fenster des Zählprozesses.



(a) VT Plot (b) R/S Plot

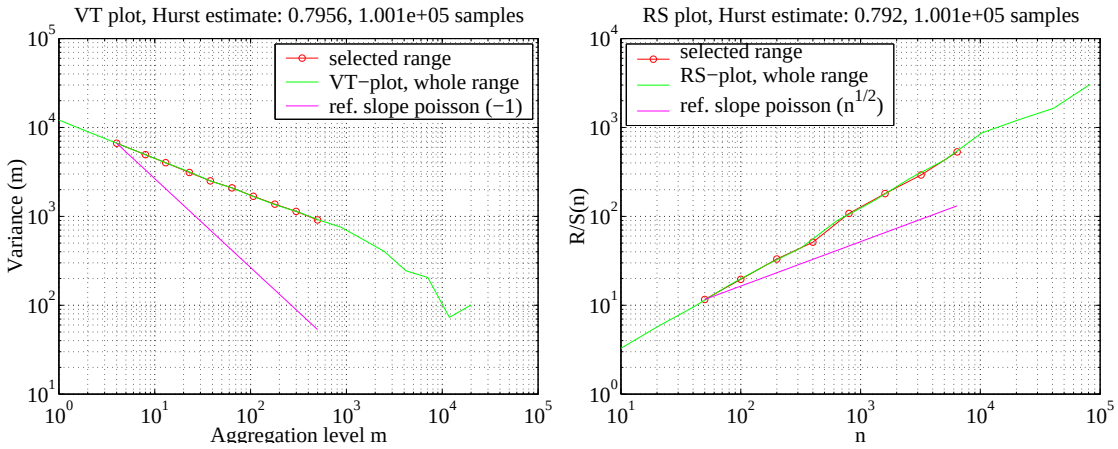
Abbildung 3.5: Schätzung der Selbstähnlichkeit zweiter Ordnung der 20-FRP Zwischenankunftszeiten.

Die Wahl der Fenstergröße des Zählprozesses ist hierbei kritisch, da bei einem zu klein gewählten Fenster kaum mehr Varianz auftreten kann. Wenn z.B. die Fenstergröße nur maximal eine Zelle zulässt, so ist die Varianz wesentlich kleiner, als wenn z.B. maximal 10^3 Zellen in einem Fenster gezählt werden können. Wird die Fenstergröße zu groß gewählt, so dauert die Simulation jedoch zu lange, und es werden nicht genug Messwerte erzielt, um mit hinreichender Genauigkeit den Hurst Parameter schätzen zu können. Als Kompromiss wurde deshalb hier ein Fenster der Größe $w = 10^3$ gewählt, d.h. bei einer mittleren Zwischenankunftszeit von $\tau = 1$ treten in einem Fenster im Mittel 10^3 Zellen auf, es können jedoch kurzzeitig auch mehr sein.

In den Bildern 3.5–3.8 sind die Hurst Parameter Schätzungen für die SupFRP und die N-Burst Quelle abgebildet. Es ist ersichtlich, dass sowohl der Zählprozeß, als auch die Zwischenankunftszeiten die Eigenschaft der Selbstähnlichkeit zweiter Ordnung (siehe Gl. A.8) aufweisen. Die Schätzwerte liegen in dem Bereich der Messgenauigkeit von ca. ± 0.05.

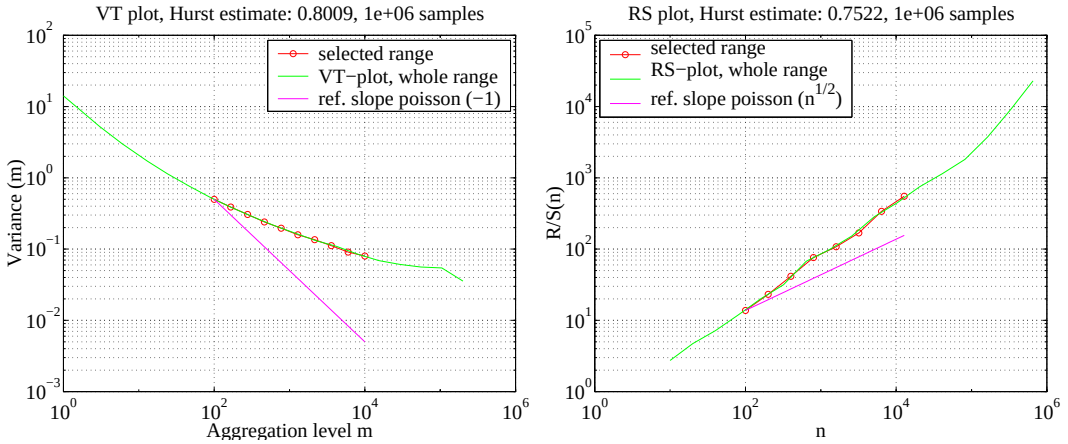
3.3.2 TCP On-Off Quelle

Analog zum vorigen Abschnitt wird hier die Untersuchung des von 20 TCP On/Off-Quellen erzeugten Verkehrs auf Selbstähnlichkeit zweiter Ordnung präsentiert.



(a) VT Plot (b) R/S Plot

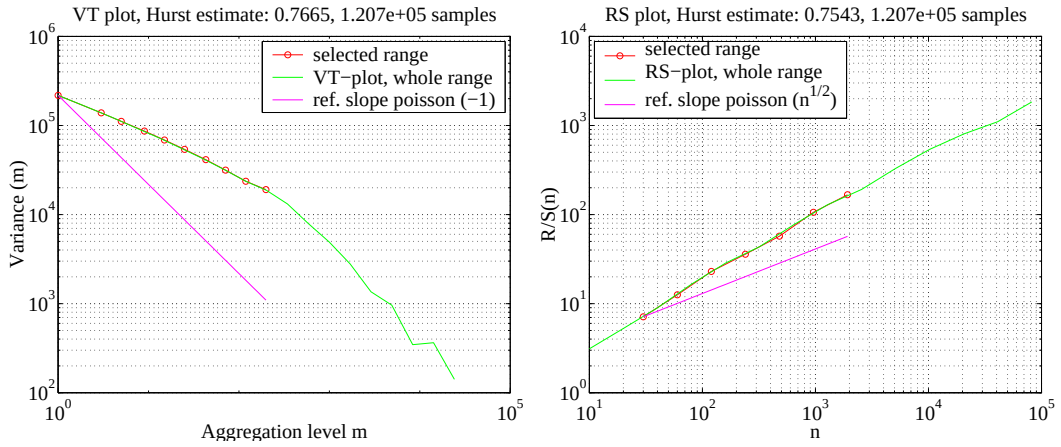
Abbildung 3.6: Schätzung der Selbstähnlichkeit zweiter Ordnung des 20-FRP Zählprozesses.



(a) VT Plot (b) R/S Plot

Abbildung 3.7: Schätzung der Selbstähnlichkeit zweiter Ordnung der 20-Burst Zwischenankunftszeiten.

- Einstellungen:
  - Anzahl der Quellen: $N = 20$
  - On-Zeiten: power-tail verteilte Dateilänge
    - $H = 0.8$
    - $\mu_{ON} = 7.9$ KByte, entspricht 15 TCP-Segmenten der Größe $MSS = 536$ Bytes
  - Off-Zeiten: negativ exponentiell verteilt mit Mittelwert $\mu_{OFF} = 20$
  - Server: konstante Bedienzeit: $\tau = 0.01$



(a) VT Plot (b) R/S Plot

Abbildung 3.8: Schätzung der Selbstähnlichkeit zweiter Ordnung des 20-Burst Zählprozesses.

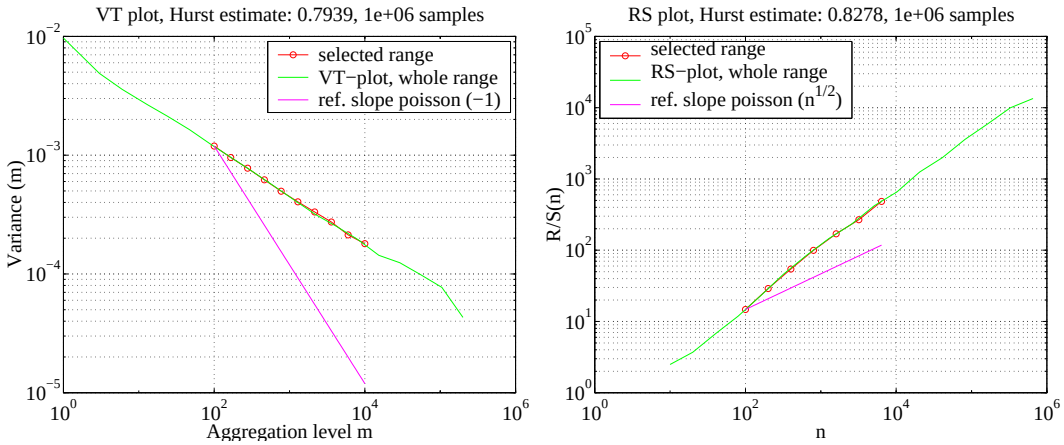
- Fenstergröße des Zählprozesses: $w = 10$ (maximal $\frac{w}{\tau} = \frac{0.01}{10} = 10^3$ Ankünfte)
- "propagation delay": $RTT_{min} = 0.04$ (plus evtl. Wartezeit in dem Puffer des Servers)
- gemessen: 10^6 Zwischenankünfte, 10^6 Zählfenster
- mittlere Last während der gesamten Simulationsdauer: $\rho \approx 13\,\%$

In den Bildern 3.9–3.10 sind die VT- bzw. RS-Plots der Zwischenankunftszeiten und des Zählprozesses dargestellt. Es ist zu erkennen, dass die Schätzungen in etwa mit dem vorgegebenen Hurst Parameter übereinstimmen.

3.4 Parametrisierung der TCP On/Off-Quelle

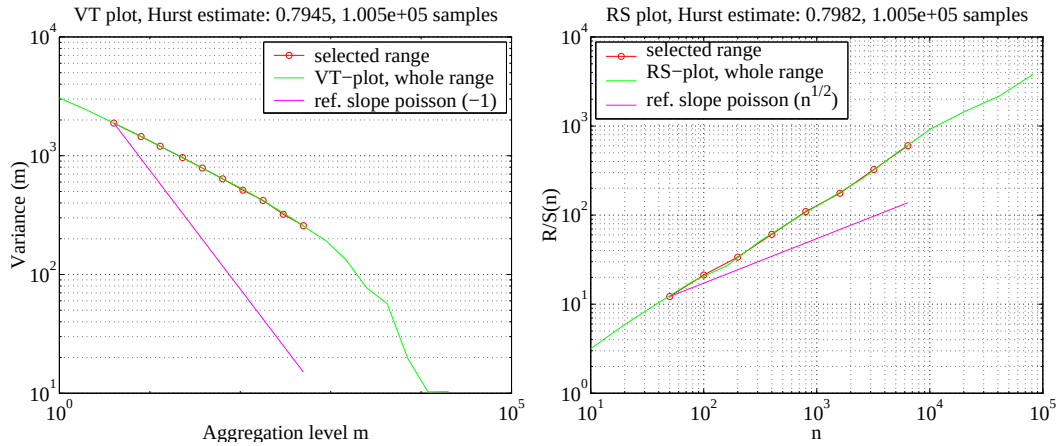
Die für diese Untersuchung relevanten Zielvorgaben stammen aus [GGS97, Seite 20]:

- Hurst Parameter der Zwischenankunftszeiten: $H = 0.7 \ldots 0.8$
- mittlere Netzwerk-Last: $\rho = 4 \ldots 20\,\%$
- quadrierter Variations-Koeffizient: $C(X)^2 = \frac{\sigma^2}{\mu^2} = 13 \ldots 22$
- mittleres Volumen einer TCP Datei-Übertragung: $\mu_{TCP} = 7.9$ KByte



(a) VT Plot (b) R/S Plot

Abbildung 3.9: Schätzung der Selbstähnlichkeit zweiter Ordnung der 20-TCP Zwischenankunftszeiten.



(a) VT Plot (b) R/S Plot

Abbildung 3.10: Schätzung der Selbstähnlichkeit zweiter Ordnung des 20-TCP Zählprozesses.

Bei allen Simulationen wurden folgende Parameter konstant eingestellt:

- On-Zeiten: power-tail verteilte Dateilänge
  - $\mu_{ON} = 7.9$ KByte, entspricht 15 TCP-Segmenten der Größe $MSS = 536$ Bytes
- Off-Zeiten: negativ exponentiell verteilt mit Mittelwert μ_{OFF}
- Server: konstante Bedienzeit: $\tau = 0.01$ sec
- Fenstergröße des Zählprozesses: $w = 10$ sec (maximal $\frac{w}{\tau} = \frac{0.01}{10} = 10^3$ Ankünfte)

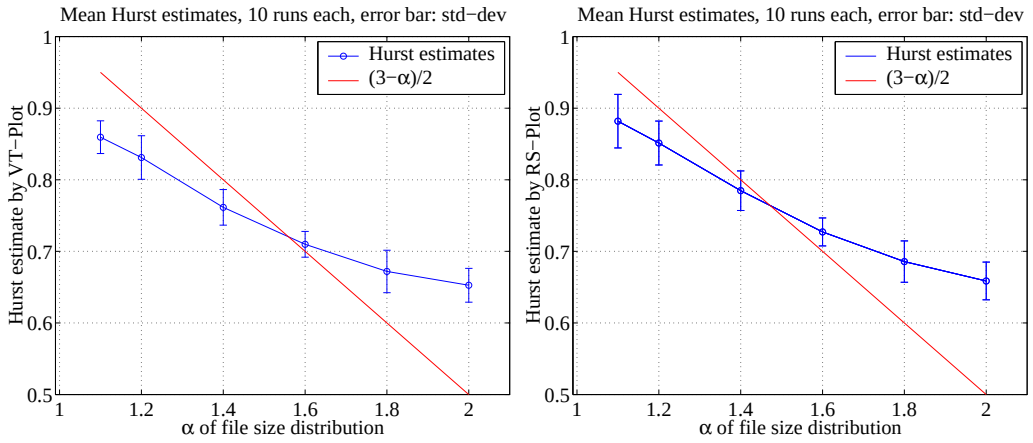
- "propagation delay": $RTT_{min} = 0.04$ sec (plus evtl. Wartezeit in dem Puffer des Servers)
- gemessen: 10^6 Zwischenankünfte

In dieser Untersuchung wird versucht, die oben genannten Zielvorgaben mit einem vereinfachten Netz wie in Bild 3.4 dargestellt zu erreichen. Zunächst wird in Abschnitt 3.4.1 untersucht, welchen Einfluss der power-tail Parameter α der Dateilängenverteilung auf den Hurst Parameter des Gesamtverkehrs hat. In Abschnitt 3.4.2 wird auf die Abhängigkeit der Selbstähnlichkeit zweiter Ordnung des Gesamtverkehrs von der Anzahl der TCP-Quellen bei näherungsweise konstanter mittlerer Last eingegangen. Die angegebenen Hurst Parameter sind Mittelwerte aus mehreren Simulationsläufen mit jeweils 10^6 gemessenen Zwischenankunftszeiten. Die Zwischenankunftszeiten dienen hier wie in [GGS97] als Zufallsvariable für die Schätzung der Selbstähnlichkeit zweiter Ordnung.

3.4.1 Variation des power-tail Index α der Dateilängenverteilung

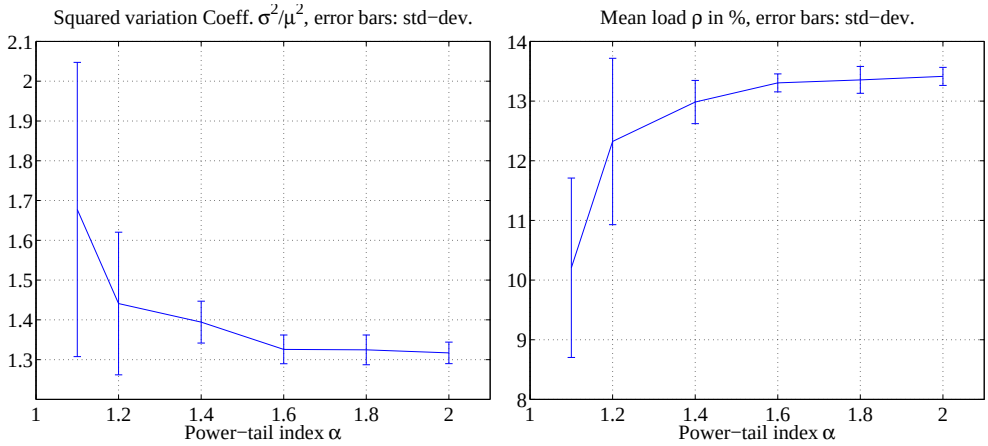
In Bild 3.11 ist die starke Abhängigkeit des Hurst Parameters H des Gesamtverkehrs von dem power-tail Index α der Dateilängenverteilung zu sehen, wie aus [PKC96] zu erwarten war.

Es ist abzulesen, dass ein power-tail Index im Bereich von $\alpha = 1.3 - 1.6$ für die Dateilängenverteilung eingestellt werden muss, um für den Gesamtverkehr einen Hurst Parameter von $H = 0.7 - 0.8$ zu erzielen. Hierzu ist noch anzumerken, dass diese Messung stark vom Mittelwert der Dateilänge abhängt, da es wesentlich ist, ob die Übertragung im Mittel aus z.B. zwei oder im Gegensatz dazu wie hier 15 TCP-Segmenten ("Maximum Segment Size" $MSS = 536$ Bytes, d.h. 7.9 KByte) besteht. Wird der Mittelwert z.B. wesentlich kleiner eingestellt, so ist die Regelung des TCP evtl. nur durch den "Slow-Start" bestimmt, eine Begrenzung aufgrund von Verzögerungen des "Acknowledgements" oder der maximalen Fenstergröße kann hier nicht auftreten. Auf die Auswertung dieses Phänomens wird hier jedoch zunächst verzichtet, da der Mittelwert von 7.9 KByte aus Messungen von [GGS97] bekannt war. Die gleichen Messungen sind unten noch einmal für $N = 10$



(a) VT Schätzungen (b) R/S Schätzungen

Abbildung 3.11: Schätzung des Hurst Parameters bei Variation des Tail-Index α der Dateilängenverteilung, 25 Quellen, Mittelwerte und Standardabweichungen aus jeweils 10 Messungen.

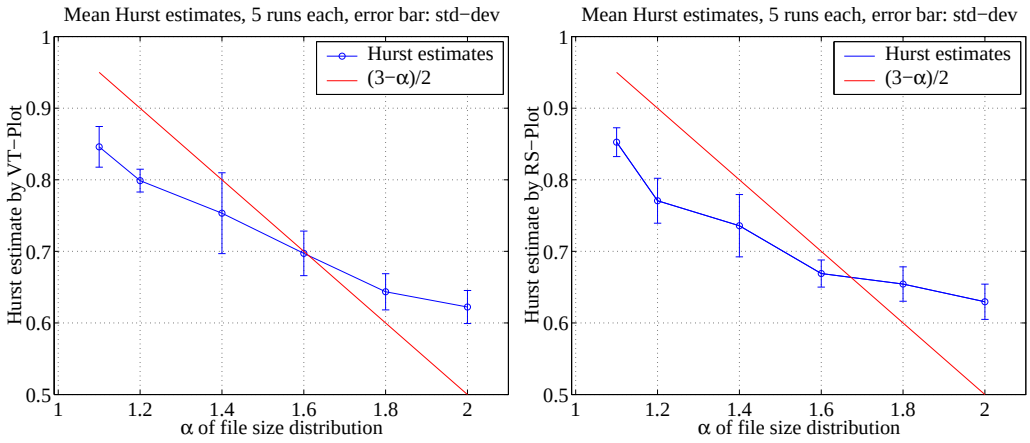


(a) Quadrierter Variationskoeffizient (b) Mittlere Last

Abbildung 3.12: Quadrierter Variations-Koeffizient und mittlere Last ρ als Funktion des tail-Index α der Dateilängenverteilung, 25 Quellen, Mittelwerte und Standardabweichungen aus jeweils 10 Messungen.

Quellen in den Bildern 3.13 und 3.14 dargestellt. Der quadrierte Variations-Koeffizient $C(X)^2 = \sigma^2/\mu^2$ und die mittlere Last während der Simulation ist in Bild 3.12 dargestellt. Es ist ersichtlich, dass die Zielgröße $C(X)^2 = 13 - 22$ nicht zu erfüllen ist. Die Veränderung der Varianz bzw. der Variations-Koeffizienten in Abhängigkeit von der Anzahl der Quellen unter näherungsweise konstanter Last wird daher im folgenden Abschnitt behandelt.

Zu den Bildern 3.11 und 3.12 ist jedoch noch folgender Sachverhalt zu bemerken, der

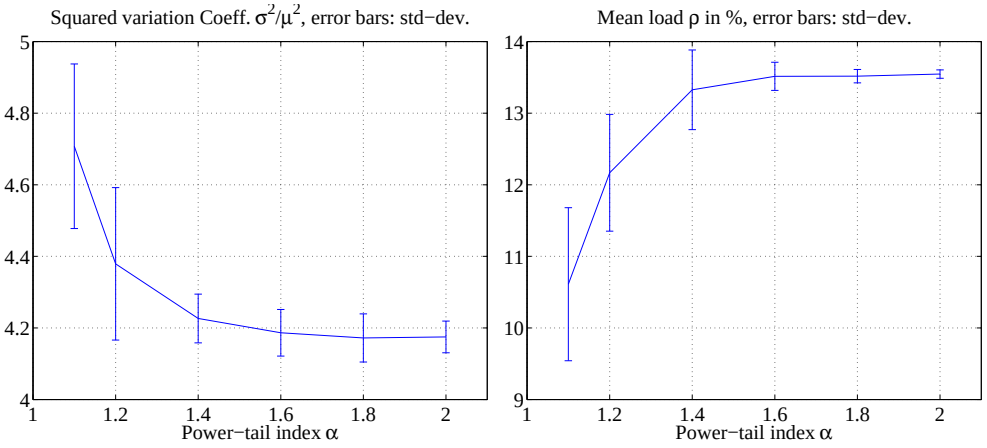


(a) VT Schätzungen (b) R/S Schätzungen

Abbildung 3.13: Schätzung des Hurst Parameters bei Variation des Tail-Index α der Dateilängenverteilung, 10 Quellen, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen.

auch für die folgenden, ähnlichen Messungen bzw. Bildern gilt: Bei der Schätzung von Parametern eines selbstähnlichen Prozesses gilt der zentrale Grenzwertsatz in seiner ursprünglichen Form nicht. D.h., dass die Varianz in dem zu schätzenden Parameter nicht der Anzahl der Messwerte N gemäß $1/\sqrt{N}$ abfällt. Für den hier vorliegenden Fall eines selbstähnlichen Prozesses gilt, dass die Varianz nur langsam, gemäß $1/N^{1-1/\alpha}$ abfällt ($1 \leq \alpha \leq 2$, siehe auch Anhang A und [Gre97a, Ber94]). Daraus lässt sich schlussfolgern, dass die in den Bildern dargestellten Fehlerbalken mit der Breite der Standardabweichung der Messwerte systematisch den wirklichen Fehler unterschätzen. Insbesondere bei Schätzung von Mittelwert μ und Varianz σ^2 ist in diesem Falle besonders problematisch, sowie auch die daraus abgeleitete Größe des quadrierten Variations-Koeffizienten $C(X)^2 = \sigma^2/\mu^2$, da Varianz und Mittelwert bei selbstähnlichen Prozessen nur ganz langsam gegen einen stationären Wert konvergieren.[1] Die Fehlerbalken sind demzufolge nur als Anhaltspunkt zu verstehen und nicht in dem Sinne von Vertrauensintervallen zu interpretieren.

[1] Es ist gut möglich, dass bei den betrachteten Zeitspannen bzw. Simulationszeiten kein stationärer Wert erreicht wird. Demzufolge können die Messwerte von dem Beobachtungszeitraum bzw. von der Anzahl der Messwerte abhängen.



(a) Quadrierter Variationskoeffizient (b) Mittlere Last

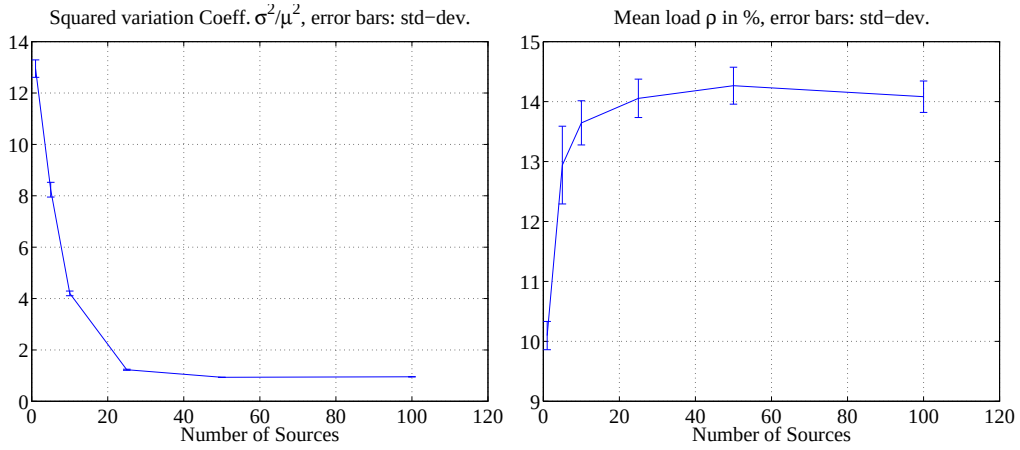
Abbildung 3.14: Quadrierter Variations-Koeffizient und mittlere Last ρ als Funktion des Tail-Index α der Dateilängenverteilung, 10 Quellen, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen.

3.4.2 Variation der Quellenzahl bei näherungsweise konstanter Last

In diesem Experiment wurde die Anzahl der Quellen in folgenden Schritten erhöht: $N = 1, 5, 10, 25, 50, 100$. Dabei wurde versucht, die mittlere Last des Netzwerks näherungsweise konstant zu halten, wobei folgende Einstellungen vorgenommen wurden:

- die Off-Zeit steigt linear mit der Anzahl der Quellen N an
- die minimale Zwischenankunftszeit steigt linear mit N an, jede Quelle kann maximal mit $N \cdot \tau$ senden ($N = 1$: eine Quelle kann den ganzen Link füllen).

Wie in Bild 3.15 (b) zu erkennen ist, ist die Last in dem Bereich von $10 - 14\,\%$ geblieben, was für diese Messung ausreichend konstant ist. In Bild 3.15 (a) ist zu sehen, dass der Variations-Koeffizient mit steigender Quellen-Anzahl abfällt, was zunächst merkwürdig erscheinen mag. Stellt man sich jedoch den Extremfall vor, so ist dieser Verlauf verständlich: Bei dem Vergleich von einer mit 100 On/Off-Quellen ist klar, dass die Varianz größer ist, wenn eine einzelne Quelle benutzt wird, die die gesamte Kapazität des Netzes kurzfristig auffüllen kann, als wenn 100 Quellen mit jeweils einem hundertstel der Link-Kapazität ein- und ausgeschaltet werden. Der gewünschte Bereich $C(X)^2 = 13 - 22$



(a) Quadrierter Variationskoeffizient (b) Mittlere Last

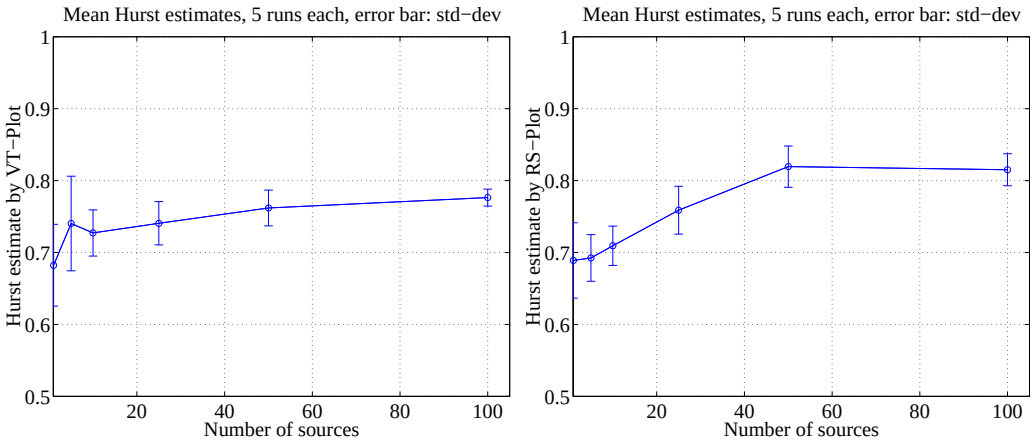
Abbildung 3.15: Quadrierter Variations-Koeffizient und mittlere Last ρ als Funktion der Quellen-Anzahl, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen.

wird nur bei $N = 1$ erreicht. Es ist jedoch anzumerken, dass das Ergebnis nicht bedeutet, dass in der Messung in [GGS97] nur eine Quelle aktiv war. Es bedeutet vielmehr, dass sich mit dem in Bild 3.4 gezeigten, vereinfachten Modell noch kein Verkehr mit gleicher Charakteristik wie in [GGS97] erzeugen lässt, offensichtlich ist die Variabilität (der Zähler des Variations-Koeffizienten, vgl. Abschnitt 3.4) in dem einfachen Modell noch zu klein. In Abschnitt 3.6 werden diese Untersuchungen durch eine Simulation mit Messung der Zwischenankunfts-Zeiten am B-WiN Knoten in München erweitert, wobei sich dort für viele Quellen aufgrund der höheren Komplexität des Netzwerkes auch ein höherer Variations-Koeffizient einstellt.

In Bild 3.16 ist zu erkennen, dass die Standardabweichung der Schätzung des Hurst Parameters mit zunehmender Quellen-Anzahl abnimmt, und dass der Hurst Parameter leicht ansteigt und ab $N = 50$ in Sättigung geht.

3.5 Erweiterung der TCP On/Off-Quelle um ein HTTP-Modell

Im Folgenden wird eine leicht modifizierte Variante der TCP On/Off-Quelle benutzt: die Quelle wird um ein darüber liegendes HTTP-Modell erweitert, d. h. es wird zuerst die



(a) VT Schätzungen (b) R/S Schätzungen

Abbildung 3.16: Schätzung des Hurst Parameters bei Variation der Quellen-Anzahl unter näherungsweise konstanter Last, Mittelwerte und Standardabweichungen aus jeweils 5 Messungen.

Anzahl der für eine HTML-Seite zu übertragenden Dateien aus einer geometrischen Verteilung gezogen, dann werden die einzelnen Dateien (HTML-Includes) einer TCP Verbindung übertragen, und nach vollendeter Übertragung wird die TCP-Verbindung geschlossen und die Off-Phase der Quelle beginnt. Dies ist eine vereinfachte Modellierung von HTTP 1.1 mit “persistent-connections”. Mit der Erweiterung um das HTTP-Modell wird das zeitliche Verhalten von “echten” WWW-Klienten noch genauer nachgebildet.

3.6 Parametrisierung der Quelle im B-WiN Modell

In Ergänzung zu der Untersuchung in Abschnitt 3.4 wird hier eine Charakterisierung des sich einstellenden Verkehrs für das komplette B-WiN gezeigt, wobei die Zwischenankunfts-Zeiten an drei ausgewählten Ausgängen des ATM-Switches in München gemessen wurden. Die Auswahl von drei von fünf Ausgängen wurde aufgrund der Last-Situation auf den Leitungen getroffen, da ein Vergleich mit den Werten aus [GGS97] bei ähnlich niedriger Last sinnvoll ist. Die Ergebnisse sind in Tabelle 3.1 für niedrige Last (Anzahl der HTTP-Includes $I = 1$) dargestellt.

Es ist deutlich zu sehen, dass der quadrierte Variationskoeffizient $C^2(X)$ auch bei den im B-WiN insgesamt verwendeten 1000 Quellen deutlich höhere Werte einnimmt. Der Wer-

Tabelle 3.1: Messwerte am Knoten München (Port 1, 3 und 5), Anzahl der Quellen N (die Verkehr über diesen Port schicken) bei $I = 1$ HTTP-Includes: Hurst Parameter H_{VT} (VT-Plot), Länge der Folge L, Mittelwert und Standardabweichung der Zwischenankunftszeiten ($\mu\sigma$), quadrierter Variationskoeffizient $C^2(X) = \sigma^2/\mu^2$ und Linkauslastung ρ.

Port / N	α	H_{VT}	H_{Wav}	L	μ	σ	$C(X)^2$	ρ [%]
1 / 43	1.3	0.77	0.90	362385	1.66e-5	2.56e-5	2.39	18.23
	1.5	0.79	0.65	418356	1.43e-5	2.11e-5	2.16	21.10
	1.7	0.64	0.71	363378	1.65e-5	2.55e-5	2.39	18.33
	1.9	0.61	0.83	353435	1.70e-5	2.69e-5	2.52	17.82
3 / 45	1.3	0.80	0.74	201530	2.98e-5	1.02e-4	11.83	14.23
	1.5	0.73	0.78	220124	2.73e-5	9.32e-5	11.69	15.55
	1.7	0.78	0.82	274480	2.19e-5	6.83e-5	9.76	19.39
	1.9	0.79	0.73	262986	2.28e-5	7.59e-5	11.08	18.58
5 / 36	1.3	0.79	0.66	174340	3.44e-5	8.04e-5	5.46	7.58
	1.5	0.47	0.85	165405	3.63e-5	8.95e-5	6.08	7.19
	1.7	0.71	0.71	204521	2.93e-5	6.79e-5	5.36	8.89
	1.9	0.75	0.57	189128	3.17e-5	7.80e-5	6.04	8.22

Tabelle 3.2: Messwerte am Knoten München (Port 1, 3 und 5), Anzahl der Quellen N (die Verkehr über diesen Port schicken) bei $I = 2$ HTTP-Includes: Hurst Parameter H_{VT} (VT-Plot), Länge der Folge L, Mittelwert und Standardabweichung der Zwischenankunftszeiten ($\mu\sigma$), quadrierter Variationskoeffizient $C^2(X) = \sigma^2/\mu^2$ und Linkauslastung ρ.

Port / N	α	H_{VT}	H_{Wav}	L	μ	σ	$C(X)^2$	ρ [%]
1 / 43	1.3	0.79	0.98	652759	9.19e-06	1.10e-05	1.43	32.92
	1.5	0.77	0.75	672475	8.92e-6	1.13e-5	1.60	33.92
	1.7	0.70	0.71	668355	8.98e-6	1.06e-5	1.38	33.71
	1.9	0.73	0.97	664406	9.03e-6	1.11e-5	1.52	33.51
3 / 45	1.3	0.80	0.78	449372	1.34e-5	3.92e-5	8.63	31.73
	1.5	0.69	0.67	418966	1.43e-5	4,45e-5	9.66	29.59
	1.7	0.77	0.97	453724	1.32e-5	4.07e-5	9.46	32.05
	1.9	0.75	0.75	409641	1.46e-5	4.36e-5	8.86	28.93
5 / 36	1.3	0.81	0.57	310518	1.93e-5	4.00e-5	4.28	13.50
	1.5	0.73	1.04	346164	1.73e-5	3.64e-5	4.42	15.05
	1.7	0.71	0.73	326715	1.84e-5	3.72e-5	4.11	14.20
	1.9	0.67	0.56	363176	1.65e-5	3.29e-5	3.97	15.78

tebereich $C(X)^2 = 13 - 22$ aus [GGS97] nicht ganz erreicht wird, so sind die Messwerte z.B. bei Port drei doch recht knapp darunter.

Zur Demonstration der Sensibilität dieser Messgrößen sind die gleichen Messungen nocheinmal für etwas höhere Last (Anzahl der HTTP-Includes $I = 2$) in Tabelle 3.2 dargestellt. Es ist deutlich, dass der Variationskoeffizient deutlich mit steigender Last abfällt.

3.7 Zusammenfassung

In diesem Kapitel wurden verschiedene Konzepte zur Modellierung von Netzwerk-Verkehr vorgestellt. Dabei war das Ziel, den Verkehr aus den Messungen an einem ATM Knoten im B-WiN [GGS97] zu modellieren. Diese Messung hatte ergeben, dass der Verkehr die Eigenschaft der Selbstähnlichkeit zweiter Ordnung besitzt (siehe auch Anhang A.3). Da die Selbstähnlichkeit sehr starke Auswirkungen auf die Leistungsfähigkeit von Netzen hat, ist sie ein wesentlicher Bestandteil der Modellierung.

Es wurden zwei Quellen-Typen vorgestellt: selbstähnliche Quellen ohne Flusskontrolle (N-Burst und SupFRP) sowie die Überlagerung von TCP On/Off-Quellen. In Abschnitt 3.3 wurde mit beispielhaften Simulationsergebnissen gezeigt, dass die implementierten Quellen im Sinne der zugrunde liegenden Modelle funktionieren.

Die Parametrisierung des aggregierten, selbstähnlichen Verkehrs von den TCP On/Off-Quellen in Abschnitt 3.4 hat schließlich gezeigt, dass dieser Ansatz geeignet ist, den in [GGS97] gemessenen Verkehr zu modellieren. Die parametrischen Simulationen haben jedoch auch gezeigt, dass das einfache Modell des Netzes, bestehend aus zwei Multiplexern, nicht ausreicht, um die Komplexität und hier speziell die Variabilität eines ganzen Netzes nachzubilden. Dies ist jedoch auf die Einfachheit des Simulationsmodells (“single bottleneck”) zurückzuführen.

In den Messungen mit dem B-WiN Modell in Abschnitt 3.6 ist die Variabilität jedoch aufgrund der hohen Komplexität des Netzes wesentlich höher, hier werden die Zielgrößen nahezu erreicht. Es wurde jedoch auch deutlich, wie sensibel diese Messgrößen auf eine Veränderung der Last reagieren. Mit dem in Abschnitt 3.6 und in den weiteren Untersuchungen benutzten HTTP-Modell in Abschnitt 3.5 wurde noch ein Schritt weiter in Richtung einer möglichst genauen Modellierung des Quellen-Verhaltens vollzogen.

Kapitel 4

Netzmodell

Ein wesentlicher Aspekt der im Rahmen dieses Projektes durchgeführten Untersuchungen ist die Simulation des gesamten B-WiN einschließlich der Wechselwirkungen zwischen den darin vorkommenden Verkehrsströmen. Das Besondere hierbei ist, dass nicht einzelne Teil-Abschnitte zur Simulation herausgegriffen werden und die zu- und abgehenden Datenströme zu anderen Teil-Abschnitten als unabhängige Quellen simuliert werden, sondern dass sämtliche Wechselwirkungen innerhalb des Netzes nachgebildet werden. Dabei ist es möglich, an jedem "Punkt" im Netz Parameter wie zum Beispiel Puffer-Belegungen, Link-Auslastungen, Verlustraten oder Durchsätze zu protokollieren.

Die verschiedenen in diesen Untersuchungen betrachteten Netzmodelle sind:

- IP über ATM, wobei die IP Router über CBR Verbindungen miteinander virtuell vermascht sind (siehe Kapitel 5 und Kapitel 6), modelliert mit einem reinen IP Netzmodell ohne ATM (siehe Abschnitt 4.3).
- IP über UBR mit und ohne Early Packet Discard (direkt geschaltete ATM-Verbindungen, keine IP-Router, siehe Kapitel 7).
- IP über ABR und IP über UBR mit EPD mit hop-by-hop Routing auf IP Ebene (siehe Kapitel 8): Jede Verbindung zwischen Eingangs- und Ausgangs-Routern wird durch eine ABR Verbindung bzw. UBR realisiert, direkte Verbindungen auf ATM Ebene existieren nur zwischen benachbarten Knoten.

- IP über ATM, wobei jede TCP-Session zu einem Verbindungsaufbau auf ATM Ebene führt, bei der ein dynamisches Routing eingesetzt wird (direkte ATM-Verbindungen von Quelle zu Senke, keine IP-Router, siehe Kapitel 9).

Die letzten drei Netz-Szenarien sind ohne die exakte Modellierung der ATM Ebene nicht denkbar. Da andererseits die Simulationen unter Einschluss der ATM Ebene äußerst zeitaufwendig sind, wurde zunächst das erste Modell vereinfacht als IP, aufgesetzt auf dem “Physical Layer”, implementiert und dann für Parameter-Studien sowie für die Analyse des durch Router begrenzten Leistungsverhaltens genutzt. Überdies stellt dieses Modell auch ein realistisches Modell für das Netz-Konzept des G-WiN dar. Wir werden dieses Modell im folgenden als "IP Netzmodell" bezeichnen. Im folgenden wird zuerst das Netzmodell mit ATM beschrieben. Die Unterschiede zwischen IP und ATM Netzmodell werden im Abschnitt 4.3 erläutert.

4.1 Geometrie und Parameter des Netzmodells

Neben den für die Quellen- und Knoten-Modelle relevanten Parametern sind netzplanerische Größen festzulegen, die sich zum größten Teil aus dem aktuellen Betrieb des B-WiN ergeben. Zur Erläuterung ist in den Bildern 4.1 und 4.2 die hier betrachtete Netzkonfiguration des B-WiN gezeigt, die Verbindungsleitungen und die ATM Knoten erkennen lässt. Die einzelnen Knoten sind wie in Abschnitt 2.1 beschrieben modelliert: Bilder 4.3–4.5.

Das Quellen-Senken Modul mit dem Protokoll-Stack einer einzelnen Quelle ist in Bild 4.4 gezeigt. Ein Paket-Generator erzeugt die Pakete gemäß einer simulierten HTTP-Verbindung und reicht sie zum TCP Socket weiter, dieser übernimmt zusätzlich zu dem TCP-Protokoll das Einpacken in IP-Pakete und danach folgend auch das Verpacken in AAL5-Pakete. Der unterste Block segmentiert die AAL5-Pakete und packt sie in ATM-Zellen und sorgt für den Verbindungsauf- und -abbau.

In Ergänzung zu der topologischen Information von Bild 4.1 gibt die vom DFN Verein veröffentlichte Verkehrsmatrix in Tabelle 4.1 Auskunft über die Intensität der Verkehre zwischen jedem Knotenpaar. Mit diesen Eingangs-Informationen muss dann das plane-

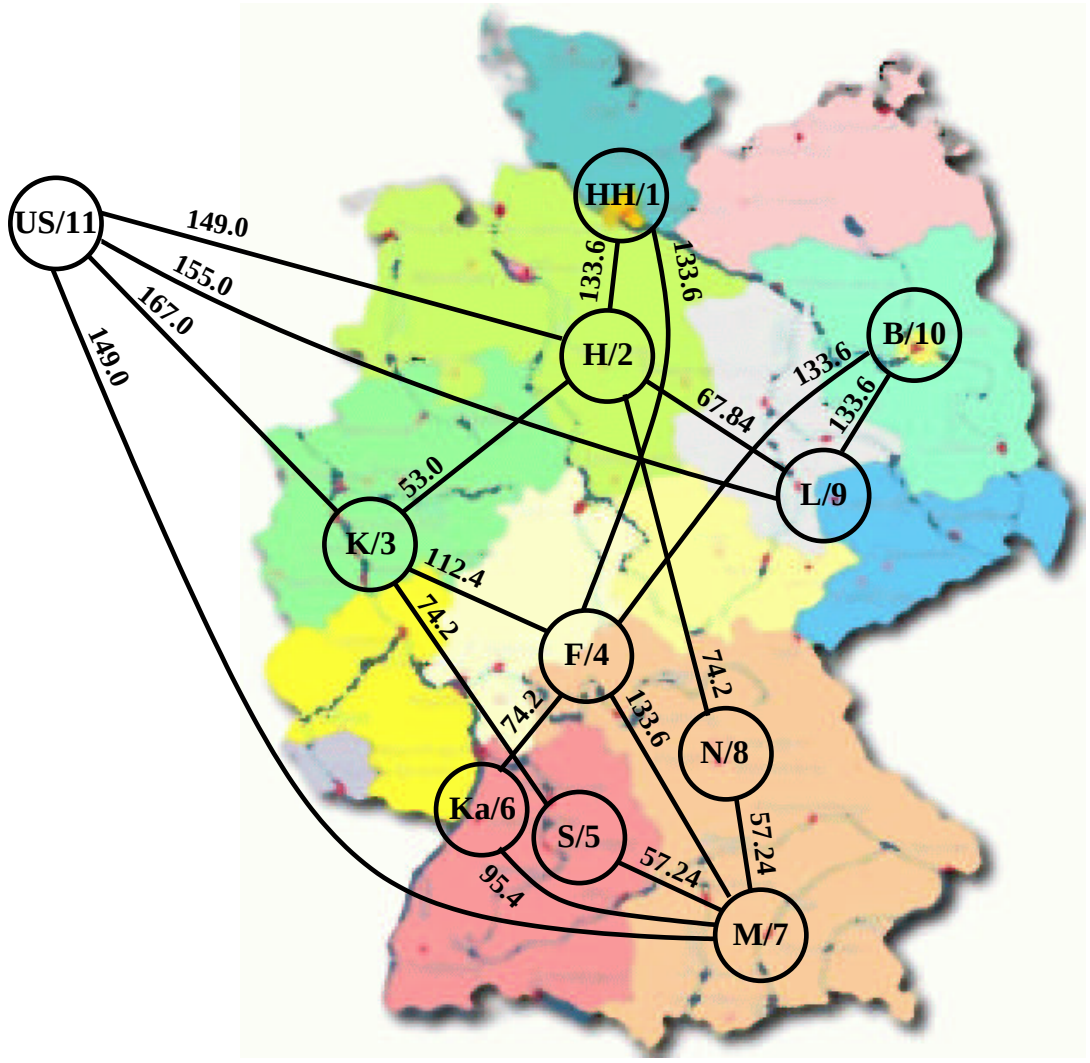


Abbildung 4.1: Linkraten im B-WiN.

rische Problem gelöst werden, über welchen Weg jeder einzelne IP-flow geführt werden soll, um die Kosten der auf den einzelnen Links angemieteten Kapazitäten zu minimieren. Als Ergebnis dieser Optimierungsaufgabe ergeben sich zum einen die Raten für die einzelnen Links und zum anderen die Gewichte der einzelnen Links, nach denen die Router die Wegelenkung vornehmen. Aus den vom DFN veröffentlichten Linkgewichten ergibt sich dann die in Tabelle 4.2 gezeigte Routing Tabelle.

Benutzt man nun diese Routing Tabelle, um dem in der Verkehrs-Matrix angegebenen Verkehrsaufkommen im Netz geeignete Wege zuzuweisen, so ergibt sich eine Verteilung auf die einzelnen Links, die nicht einer gleichmäßigen, sondern doch einer sehr stark schwankenden Auslastung entspricht (Bild 4.6).

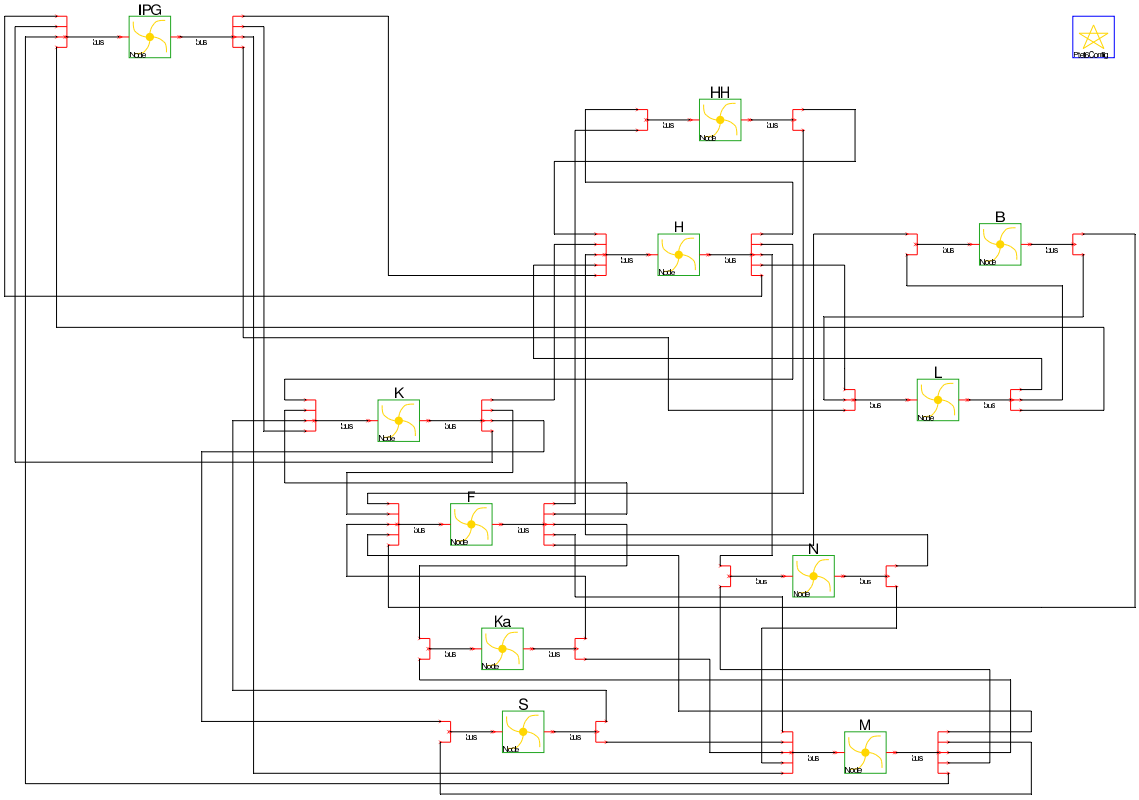


Abbildung 4.2: Das B-WiN Netzmodell.

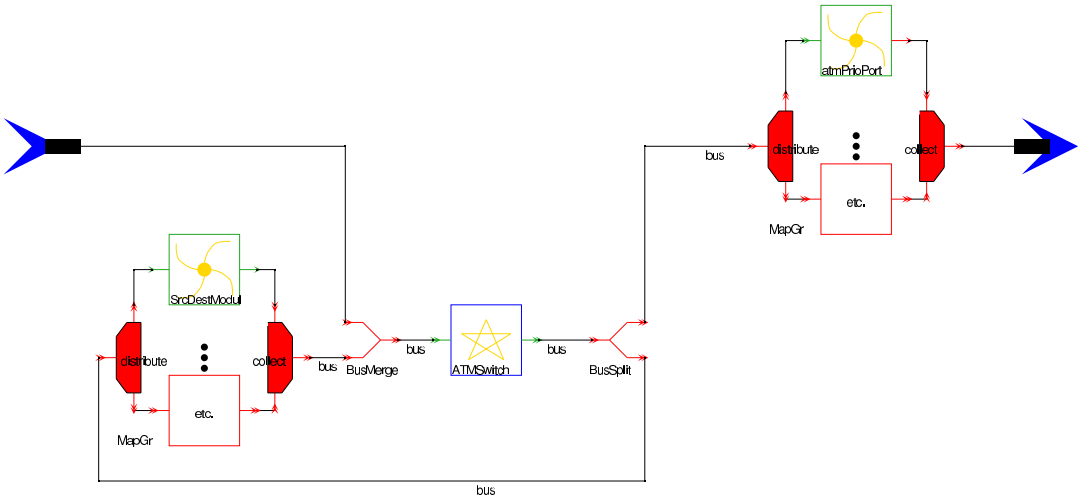


Abbildung 4.3: B-WiN Knoten.

Ohne den Ergebnissen zu weit vorgreifen zu wollen, sei hier festgehalten, dass die relativ ungleichmäßigen Auslastungen der Verkehrs-Matrix in Tabelle 4.1 nicht den Ergebnissen der Simulation entsprechen. In den Simulationen stellt sich eine andere, z.T gleichmäßigere Auslastung ein. Hierfür wird u.a. die Dynamik der TCP Überlast-Abwehr verantwortlich gemacht, die bei der Planungsaufgabe naturgemäß nicht berücksichtigt wird.

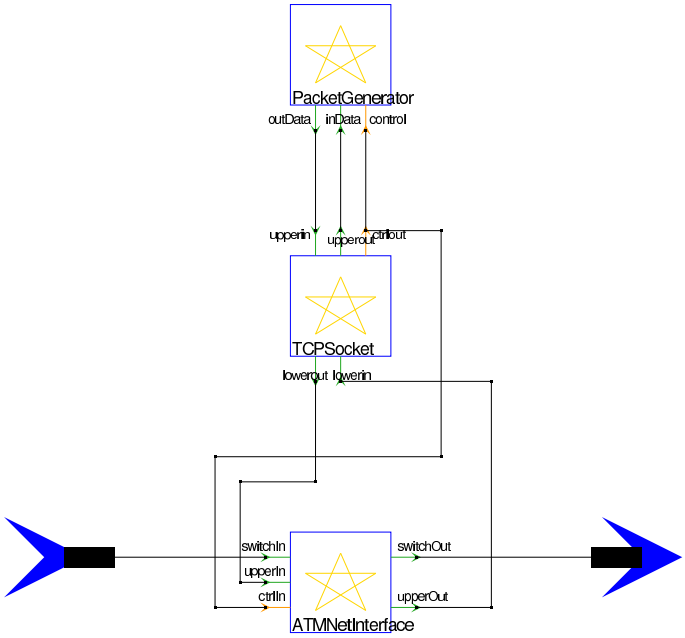


Abbildung 4.4: B-WiN Quellen-Senken Modul.

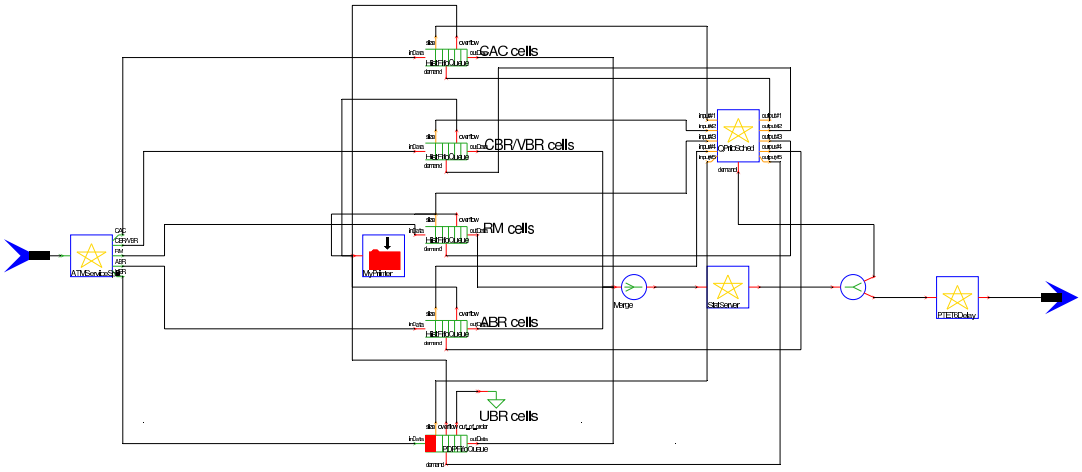


Abbildung 4.5: ATM Prioritäts-Warteschlangen Modul.

4.2 Verteilung der Quellen auf Virtuellen Pfade

Zur Verteilung der 1000 Quellen auf die virtuellen Pfade wurde folgendes Vorgehen gewählt: Die Anzahl der Quellen auf dem VP von A nach B ist proportional zu dem gemessenen Durchsatz (vgl. Tabelle 4.1). Bei 1000 Quellen führte das zu einer Rate von ca. 1.5 Mbit/s pro Quelle. Die resultierende Verteilung Quellen[1] auf die virtuellen Pfade ist

[1]Das Ziel bei der Quellenverteilung war 1000 Quellen, das Resultat des Algorithmus war jedoch 998 Quellen. Weil die Verteilung der Quellen proportional zu der gemessenen Verkehrs-Matrix ist, ist es nicht immer möglich, die gewünschte Anzahl an Quellen exakt zu erreichen.

Tabelle 4.1: Gemessene Verkehrs-Matrix auf IP-Ebene, Raten in Mbit/sec.

	1-HH	2-H	3-K	4-F	5-S	6-Ka	7-M	8-N	9-L	10-B	11-IPG
1-HH	0.00	7.18	5.50	14.81	1.34	2.91	3.02	0.82	6.35	8.26	17.61
2-H	5.77	0.00	11.05	10.80	1.23	3.49	3.83	2.95	6.76	4.36	15.47
3-K	4.62	15.84	0.00	23.81	2.84	4.42	3.56	1.69	7.07	2.66	39.49
4-F	37.92	32.50	62.93	0.00	24.01	34.62	20.97	17.04	43.59	24.47	19.84
5-S	1.55	1.19	3.33	9.99	0.00	0.94	3.22	1.31	2.03	0.59	11.13
6-Ka	2.12	4.20	4.93	14.33	0.93	0.00	1.57	1.08	3.50	1.01	21.91
7-M	4.12	7.40	7.26	7.61	2.42	3.62	0.00	4.13	5.60	4.67	12.06
8-N	0.80	23.27	1.22	5.27	2.36	1.36	6.84	0.00	3.32	0.44	8.85
9-L	7.82	11.51	6.32	17.82	1.97	3.28	2.47	1.91	0.00	4.03	30.92
10-B	5.99	11.43	3.11	10.17	0.95	1.29	5.86	0.71	4.34	0.00	18.65
11-IPG	54.11	56.24	126.78	28.53	37.54	60.32	37.60	33.71	71.11	41.55	0.00

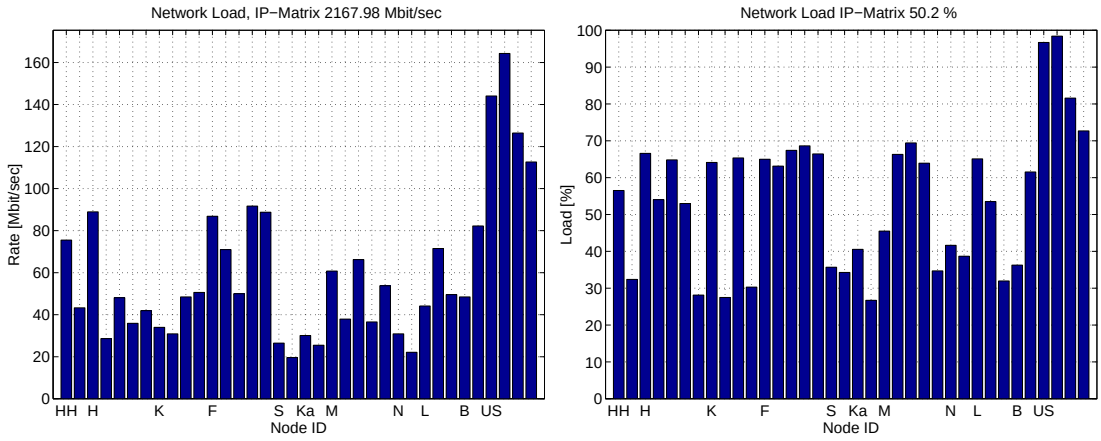
Tabelle 4.2: Wegelenkung auf der Basis von Knotennummern.

	1-HH	2-H	3-K	4-F	5-S	6-Ka	7-M	8-N	9-L	10-B	11-IPG
1-Hamburg	-	2	2	4	2	4	4	2	2	4	2
2-Hannover	1	-	3	1	3	1	8	8	9	9	11
3-Köln	2	2	-	4	5	4	5	2	2	4	11
4-Frankfurt	1	1	3	-	7	6	7	7	10	10	7
5-Stuttgart	3	3	3	7	-	7	7	7	3	7	3
6-Karlsruhe	4	4	4	4	7	-	7	7	4	4	7
7-München	4	8	5	4	5	6	-	8	8	4	11
8-Nürnberg	2	2	2	7	7	7	7	-	2	2	2
9-Leipzig	2	2	2	10	2	10	2	2	-	10	11
10-Berlin	4	9	4	4	4	4	4	9	9	-	9
11-IP-Gate (USA)	2	2	3	7	3	7	7	2	9	9	-

in Tabelle 4.3 zu sehen. Zu den aufgeführten (aktiven) Quellen gehört jeweils noch eine (passive) Senke hinzu. Die Verteilung ergibt sich durch Spiegelung an der Diagonalen. Die Zeilensummen ergeben die gesamte Anzahl der Quellen an dem jeweiligen Knoten.

4.3 IP Netzmodell

Dieses Modell betrachtet ein gemäß Bild 4.2 vereinfachtes Modell, indem die IP Router direkt über Links mit der in Abschnitt 4.1 erläuterten Dimensionierung verbunden sind. Es werden also im Wesentlichen die ATM-Switches durch IP-Router ersetzt (s. Bild 4.7) und das ATM-Netz-Interface weggelassen (s. Bild 4.8). Zusätzlich sind die Ausgangs-Warteschlangen des IP-Routers einfacher aufgebaut, da sie keine priorisierten Verkehre unterstützen (s. Bild 4.9).



(a) Netz-Last in Mbit/s. (b) Netz-Last in %.

Abbildung 4.6: Berechnete Auslastung der Links in (a) Mbit/s bzw. (b) %, als Resultat von Verkehrs-Matrix, Link-Dimensionierung und Routing Tabelle.

Tabelle 4.3: Quellen-Verteilung bei insgesamt 1000 Quellen.

	1-HH	2-H	3-K	4-F	5-S	6-Ka	7-M	8-N	9-L	10-B	11-IPG
1-Hamburg	-	5	4	10	1	2	2	1	5	6	12
2-Hannover	4	-	8	8	1	3	3	2	5	3	11
3-Köln	4	11	-	16	2	3	3	2	5	2	26
4-Frankfurt	25	22	42	-	16	23	14	12	29	17	14
5-Stuttgart	2	1	3	7	-	1	3	1	2	1	8
6-Karlsruhe	2	3	4	10	1	-	2	1	3	1	15
7-München	3	5	5	6	2	3	-	3	4	4	8
8-Nürnberg	1	3	1	4	2	1	5	-	3	1	6
9-Leipzig	6	8	5	12	2	3	2	2	-	3	21
10-Berlin	4	8	3	7	1	1	4	1	3	-	13
11-IP-Gate (USA)	36	37	84	19	25	40	25	23	47	28	-

4.4 Parameter der B-WiN Simulationen

Um die Simulationszeiten in vernünftigen Grenzen zu halten, wurde von einer Population von 1000 aktiven TCP-Quellen ausgegangen. Da die Quellen als TCP On-Off Quellen modelliert werden, kann die Off-Zeit dazu benutzt werden, verschiedene Verkehrs-Intensitäten einzustellen. Es hat sich jedoch gezeigt, dass dies nicht zu einem voll ausgelastetem Netz führt. Demgegenüber lässt sich die Netz-Last sehr gut mit der Anzahl der “Includes” der HTTP-Modellierung einstellen: Sie gibt an, wieviele Dateien innerhalb eines HTTP-Anforderung übertragen werden[2]. Hingegen wird der Hurst-Parameter, also der Grad der Selbst-Ähnlichkeit zweiter Ordnung des Verkehrs, durch den Parame-

[2]Da die Modellierung HTTP 1.1 mit “persistent connections” entspricht werden alle “Includes” innerhalb einer TCP-Verbindung übertragen.

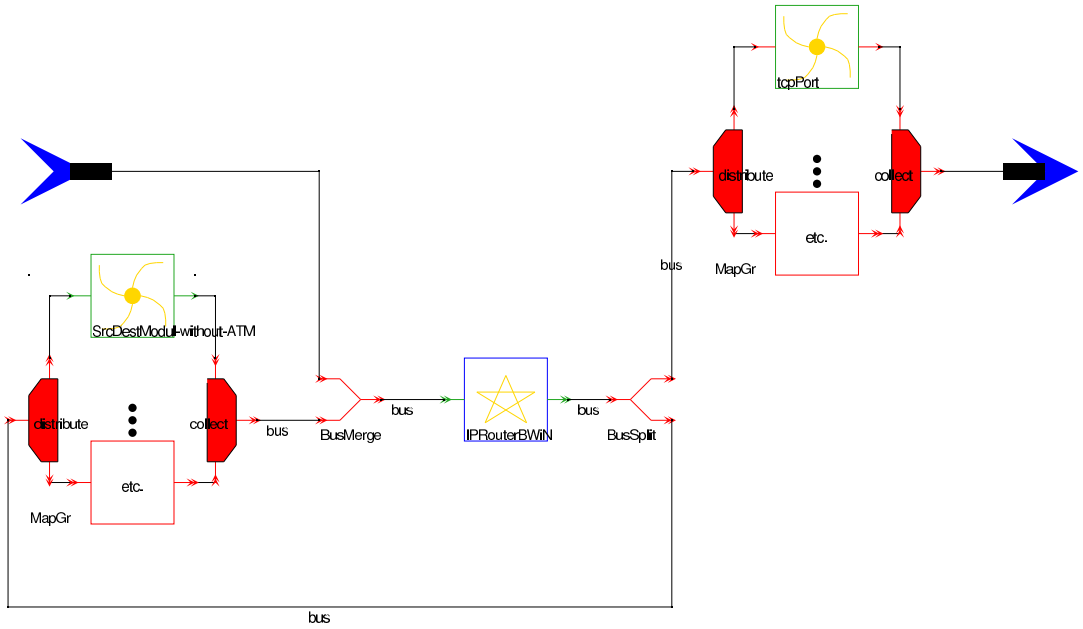


Abbildung 4.7: B-WiN Knoten IP Netzmodell.

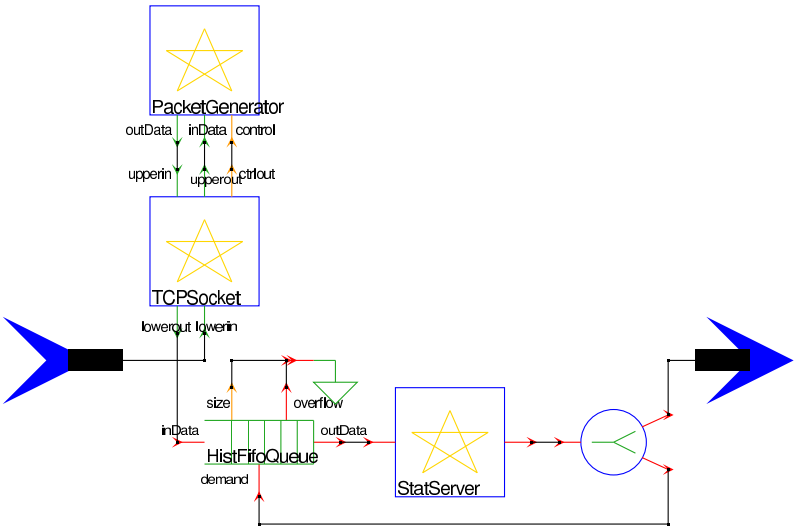


Abbildung 4.8: B-WiN Quellen-Senken Modul IP Netzmodell.

ter α der TPT Verteilung festgelegt. Die gemeinsamen Parameter von Quellen sowie die Puffergrößen der Knoten sind in Tabelle 4.4 zusammengefasst.

Die wesentlichen variablen Parameter des Netzmodells sind die mittlere Anzahl der HTML-Includes *I* und der α -Parameter der TPT-Verteilung. Mit *I* wird die Last im Netz eingestellt und mit Hilfe von α wird der Grad der Selbstähnlichkeit zweiter Ordnung des zu generierenden Verkehrs eingestellt. Die TPT-Verteilung wurde hart auf die maximale Dateilänge von einem MByte begrenzt, da ansonsten zu große Dateien für die hier ge-

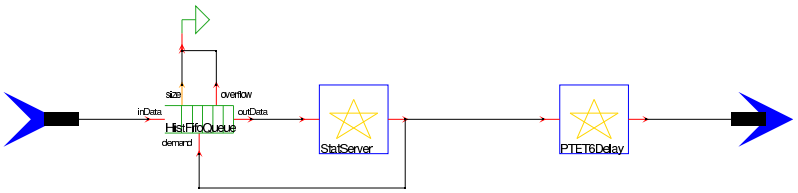


Abbildung 4.9: Warteschlangen Modul IP Netzmodell.

Tabelle 4.4: Gemeinsame Parameter aller Simulationen.

Quellen	On-TPT: mittlere Dateilänge	10 kByte
	On-TPT: truncation level für TPT	12
	mittlere Off-Zeit	100 msec
	maxmale Rate einer einzelnen Quelle	10 Mbits/sec
	maxmale Dateilänge	1 MByte
TCP/IP	TCP Version	Reno
	MSS	1460 Bytes
	MTU	1500 Bytes $\approx$ 32 ATM-Zellen
Puffer [Paketen]	CAC	50
	CBR/VBR	100
	RM	1000
	ABR	10000
	UBR	20000
	“Early Packet Discard“ Schwelle	18000
	IP-Router (UBR-Puffer/32)	$20000/32 = 625$
Mess-Parameter	Fenstergröße des Zählprozesses	1 msec

wählte Simulations-Dauer gezogen werden könnten, die ganz wesentlich die statistischen Eigenschaften der Ergebnisse dominieren könnten.

Kapitel 5

Warteschlangen-Dimensionierung bei selbstähnlichem Verkehr

In diesem Kapitel wird die Frage untersucht, wie die Dimensionierung von Warteschlangen bei selbstähnlichem Verkehr gegenüber einer Dimensionierung bei herkömmlichen Verkehrsmodellen vorzunehmen ist. Hierzu werden die Warteschlangen der IP-Router der Verbindungen Köln/München, Frankfurt/München und USA/Köln betrachtet, da über diese Verbindungen sehr viel Verkehr fließt. Es wird jedoch das gesamte B-WiN simuliert, d.h. auch die Wechselwirkungen der einzelnen Verkehrsströme, die mit dem hier untersuchten Verkehrs-Strom über andere Knoten geroutet werden, sind in dem Modell berücksichtigt. Da der untersuchte Verkehr sich sowohl aus direkt am Knoten angeschlossenen Quellen und Senken als auch aus weiter entfernten Quellen und Senken zusammensetzt, wird eine realitätsnahe Situation nachgebildet. Für die Simulationen in diesem Kapitel wurde das in Abschnitt 4.3 beschriebene IP-Netzmodell (ohne ATM Layer) verwendet.

Die Wahl des Qualitäts-Parameters für den Vergleich der Leistungsfähigkeit wird in Abschnitt 5.1 diskutiert. Die Formeln für den Goodput des Poisson Referenz Modells werden in Abschnitt 5.2 eingeführt. Die Abhängigkeit des Goodputs von anderen Parametern im Falle des mit Hilfe von TPT On/Off TCP Quellen erzeugten selbstähnlichen Verkehrs wird in Abschnitt 5.3 ebenso wie das weitere Vorgehen bei der Dimensionierung der Warteräume beschrieben. Die Ergebnisse werden in Abschnitt 5.4 beschrieben. In Ab-

schnitt 5.4.2 werden die Ergebnisse mit einem weitere Referenz System dargestellt, bei dem gegenüber dem TPT On/Off TCP die TPT Verteilung der Dateilängen durch eine negativ exponentielle Verteilung ersetzt wurde. In Abschnitt 5.4.3 werden die Ergebnisse zusammengefasst.

5.1 Qualitäts-Parameter für den Leistungs-Vergleich

Bei der Warteraum-Dimensionierung ist es das Ziel, eine Funktion $Z = f(\rho, \alpha)$ aufzustellen, die das Verhältnis der benötigten Puffergrößen bei selbstähnlichem Verkehr im Vergleich zu einem Referenz-Verkehr bei einem konstantem Qualitäts-Parameter darstellt. Aus dieser Funktion kann dann z.B. abgeleitet werden, dass für einen bestimmten Parametersatz der selbstähnliche Verkehr eine um den Faktor Z größere Warteschlange als der Referenz-Verkehr benötigt, um eine ähnliche Leistung zu erreichen.

Es ist zunächst wichtig, die Entscheidung zu treffen, welcher Qualitäts-Parameter gewählt werden soll, und von welchen anderen Parametern dieser, und damit die gesuchte Funktion, abhängt. Es stehen hier z.B. die Zell-Verlustrate CLR, die Zell-Verzögerung, der Datendurchsatz sowie der effektive Durchsatz “Goodput”, (Gl. 5.1) als charakteristische Qualitätsparameter zur Auswahl. Da es sich bei TCP/IP-Verkehr um eine gesicherte Übertragung handelt, ist die CLR nicht das entscheidende Kriterium für die Qualität einer Verbindung. Bei vielen TCP/IP-Anwendungen (z.B. FTP-Anwendungen, WWW-Browser) ist die Zell-Verzögerung kein kritischer Qualitäts-Parameter. Der Datendurchsatz ist ebenfalls kein sinnvolles Kriterium, da häufige Retransmissionen zu einem hohen Datendurchsatz bei kleinem Goodput führen können. Demzufolge wird für diese Studie der relative Goodput G als Qualitätsparameter für den Vergleich ausgewählt, wobei die Volumina in Bytes auf der IP Schicht gemessen werden:

$$G = 1 - \frac{VerlustVolumen + RetransmissionsVolumen}{GesamtVolumen}. \tag{5.1}$$

5.2 Poisson Referenz Verkehr

Obwohl klassische Warteschlangenmodelle dem reaktiven Verhalten des TCP Protokolls nicht gerecht werden können, wird als eine Referenz ein reines Poisson Quellenmodell eingeführt, um einen Eindruck zu vermitteln, wie weit man sich im TCP/IP Umfeld von den Dimensionierungsvorschriften der klassischen Verkehrstheorie entfernt. Die analytische Lösung für die Verlustrate einer Warteschlange der Länge **B** bei der Verkehrslast ρ eines M/D/1/B Systems ist gegeben durch [RMV96], S. 391:

$$P_{loss,M/D/1/B} = 1 - (1-\rho)\sum_{i=0}^{B}\frac{(\rho(i-B))^i}{i!}e^{-\rho(i-B)} \tag{5.2}$$

Diese Formel lässt sich leider nicht nach B auflösen und ist wegen numerischer Instabilitäten nicht direkt nutzbar[1]. Eine attraktive Näherungslösung, die sich auch nach B auflösen lässt, ist in [RMV96], S. 392, zu finden :

$$P_{loss,M/D/1/B} \approx e^{-2B(1-\rho)} \tag{5.3}$$

Durch Auflösen nach B ergibt sich mit $G_{M/D/1/B} = 1 - P_{loss,M/D/1/B}$ der gesuchte Zusammenhang[2] zwischen dem Goodput G, der Last ρ und der Warteschlangenlänge B:

$$B \approx \frac{ln(P_{loss,M/D/1/B})}{-2(1-\rho)} = \frac{ln(1-G_{M/D/1/B})}{-2(1-\rho)} \tag{5.4}$$

Für die praktische Benutzung wird hierbei noch der Verschnitt durch TCP- und IP-Header (40 Bytes Header-Informationen insgesamt, eine MTU von 1500 Bytes) durch einen Faktor $f = 1500/1460$ berücksichtigt.

$$B \approx \frac{ln(1-f\cdot G_{M/D/1/B})}{-2(1-\rho)} \tag{5.5}$$

[1] Es werden im Vorzeichen alternierende Terme mit großen Exponenten und sehr kleinen Differenzen addiert.

[2] Da hier kein Protokoll die Übertragung sichert, und somit auch keine Retransmissionen stattfinden, entartet der Goodput für diesen einfachen Fall somit zum Komplement der Verlustrate.

5.3 TPT On/Off TCP Modell

In dem hier betrachteten TPT On/Off TCP Quellenmodell hängt der Goodput von folgenden Parametern ab, die wiederum teilweise gegenseitig voneinander abhängig sind:

- Verkehrs-Last ρ
- Anzahl der Quellen N
- Hurst-Parameter H
- Warteschlangen-Länge B
- mittlere Round-Trip Time RTT.

Durch das TCP-Verhalten kann die Verkehrslast jedoch nur indirekt über die Anzahl der Quellen N, deren Off-Zeit, und den Mittelwert der TPT-Verteilung (Mittelwert des zu übertragenden Volumens im On-Zustand) beeinflusst werden. Der Hurst-Parameter kann ebenfalls nicht direkt eingestellt werden, hängt jedoch sehr stark mit dem gewählten power-tail Parameter α zusammen (siehe auch Abschnitt 3.4). Andererseits hat jedoch auch der power-tail Parameter α wieder einen Einfluss auf den Goodput, da hiermit die Variabilität des Verkehrs eingestellt wird. Die Round-Trip Time hängt wiederum von der Warteschlangen-Länge B ab. Bei TCP ist außerdem zu erwarten, dass ab einer gewissen Last der Goodput wieder abnimmt, da mit steigender Last auch die Anzahl der Verluste bzw. Retransmissionen steigt. Aufgrund der Komplexität der Abhängigkeiten wird versucht, diese Abhängigkeiten durch Parameterstudien in Simulationsexperimenten zu erfassen.

Die Last ρ wird im folgenden mit der mittleren Anzahl der HTTP “Includes” eingestellt. Es sei aber nochmals ausdrücklich darauf hingewiesen, dass ρ nicht als unabhängige Variable aufgefasst werden kann: In dem TCP Regelprozess stellt sich eine mittlere Last ρ ein, deren Wert in einem so komplexen Netz wie dem B-WiN nicht quantitativ vorhergesagt werden kann. Der Hurst-Parameter H wird mit dem power-tail Index α abgestimmt. Da der Hurst-Parameter H jedoch auch von der sich einstellenden Last, also von der Anzahl der HTTP-Includes, abhängt ist, wird die Funktion $Z = f(\rho, \alpha)$ gesucht, und nicht die

Funktion $Z = f(\rho, H)$. Um einen möglichst großen Bereich mit dieser Dimensionierung abzudecken, wird der für einen Netzbetreiber praktisch relevante Bereich für eine Dimensionierung von Warteschlangen von $\rho \in [0.5, 1]$ und $H \in [0.55, 0.85]$ daher mit mehreren Simulationsläufen untersucht.

Die vergleichende Darstellung des Goodputs in Abhängigkeit von der Warteschlangen-Länge B, von dem Hurst-Parameter H und von der Last ρ des selbstähnlichen Verkehrs mit dem Goodput des Referenzverkehrs bei gleichem B und ρ erlauben dann eine näherungsweise Ermittlung der gesuchten Funktion $Z = f(\rho, H)$ (bzw. $Z = f(\rho, \alpha)$). Für ein gewähltes ρ und H gibt die Funktion den Faktor wieder, um den die Warteschlange bei selbstähnlichem Verkehr größer sein muss als bei dem Referenz Verkehr, um den gleichen Goodput zu erreichen.

5.4 Ergebnisse

5.4.1 M/D/1/B Referenz System

In Bild 5.1 ist der Verlauf von $Z = f(G, \alpha)$ dargestellt, der aus den Messungen von Goodput und Durchsatz (Bild 5.1, unten) im Zusammenhang mit Gl. (5.5) resultiert. Solange die Last unter ca. 55 % liegt, kommt das M/D/1/B System mit einem Pufferplatz aus und der den Pufferbedarf charakterisierende Faktor für einen IP Knoten deckt sich mit der absoluten Größe dieses Puffers. Mit steigendem Verkehr wächst der Pufferbedarf des M/D/1/B Systems an, so dass die Kurven durch das Absinken ein Angleichen des Pufferbedarfs der beiden Systeme signalisieren: Der Faktor Z geht bei höchster Last für $B \geq 200$ auf Werte im Bereich $30 - 190$ zurück (siehe Bild 5.1, oben rechts). Dies bedeutet, dass die Systeme bis zu einer Auslastung von 95 % sehr unterschiedlichen Pufferbedarf haben, so dass das M/D/1/B System kaum als sinnvolle Referenz betrachtet werden kann. Da sich dieses Bild in allen Messungen mit unterschiedlichen Werten des power-tail Index α und an anderen Ports qualitativ wiederholt, wird dieser Aspekt nicht weiter verfolgt. Im folgenden Abschnitt wird daher $Z = f(\rho, \alpha)$ bzgl. des Poisson On/Off TCP Referenz Systems untersucht.

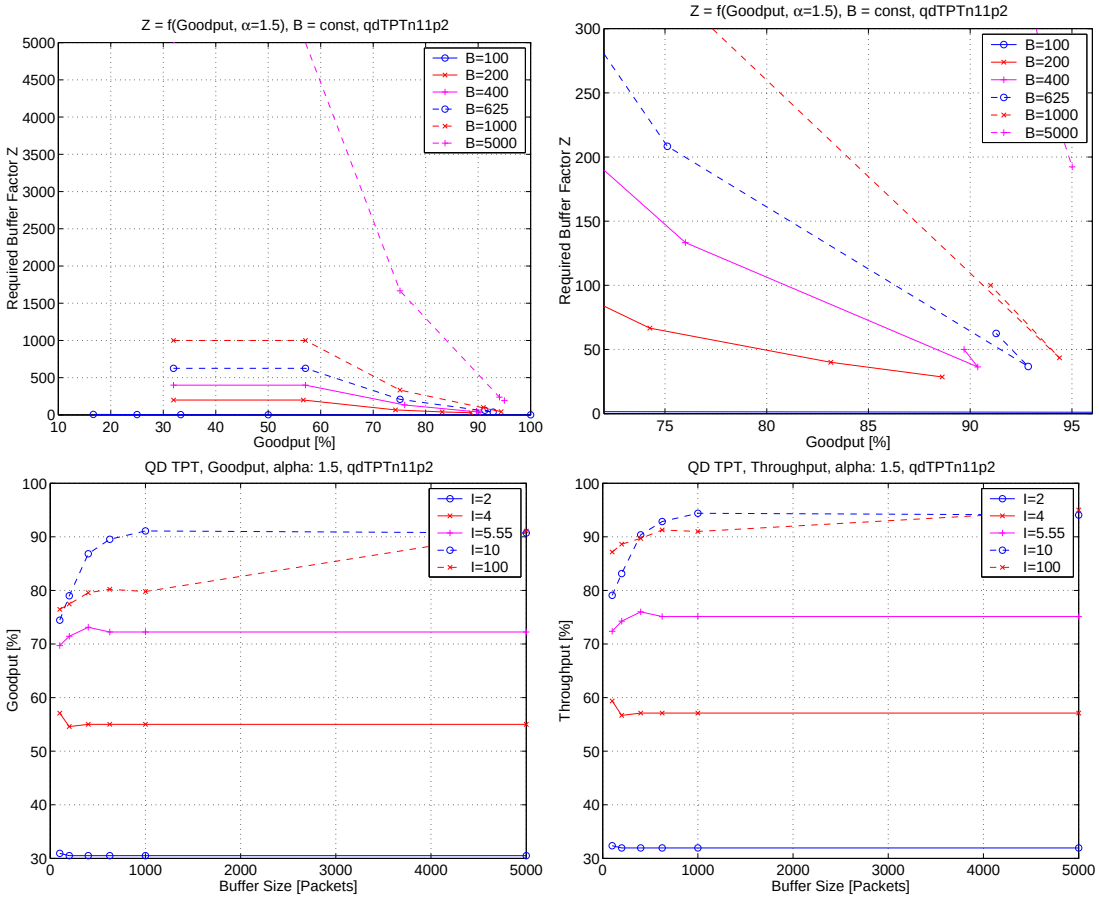


Abbildung 5.1: $Z = f(G, \alpha)$ USA $\rightarrow$ Köln, TPT On/Off TCP mit M/D/1/B als Referenzsystem (oben links bzw. Ausschnitt oben rechts), Goodput (unten links) und Durchsatz (unten rechts).

5.4.2 Poisson On/Off TCP Referenz

Als Näherung an das in Abschnitt 5.3 vorgestellte Modell dient das Poisson On/Off Modell, das wie folgt charakterisiert wird:

- negativ exponentiell verteilte Dateilängen und Off-Zeiten
- TCP-Regelung zur Dateiübertragung
- die verbleibenden Parameter sind identisch mit denen des TPT On/Off Modells

Durch die Verwendung eines Referenz Systems, das selbst nur durch Simulationen ausgewertet werden kann, ergibt sich das Problem, dass sich nicht genau die gleichen Werte für

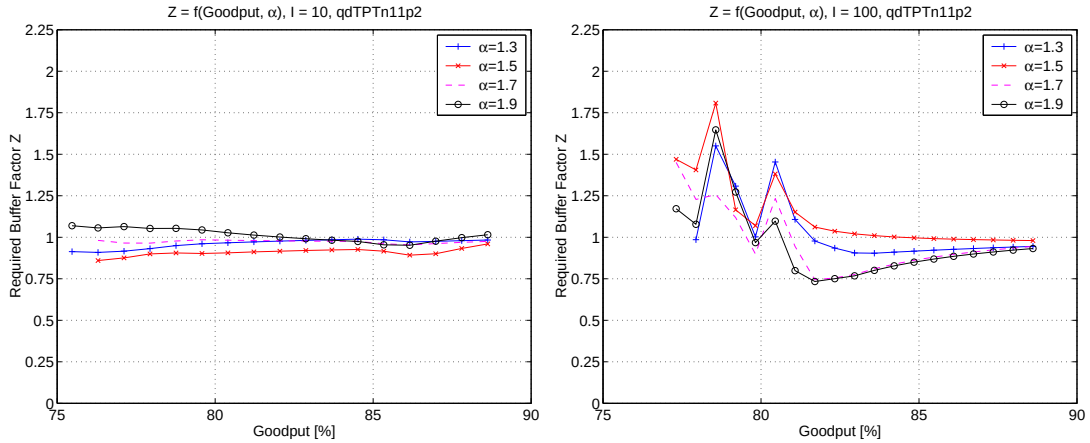


Abbildung 5.2: $Z = f(I, \alpha)$ für $I = 10$ (links) und $I = 100$ (rechts), USA $\rightarrow$ Köln.

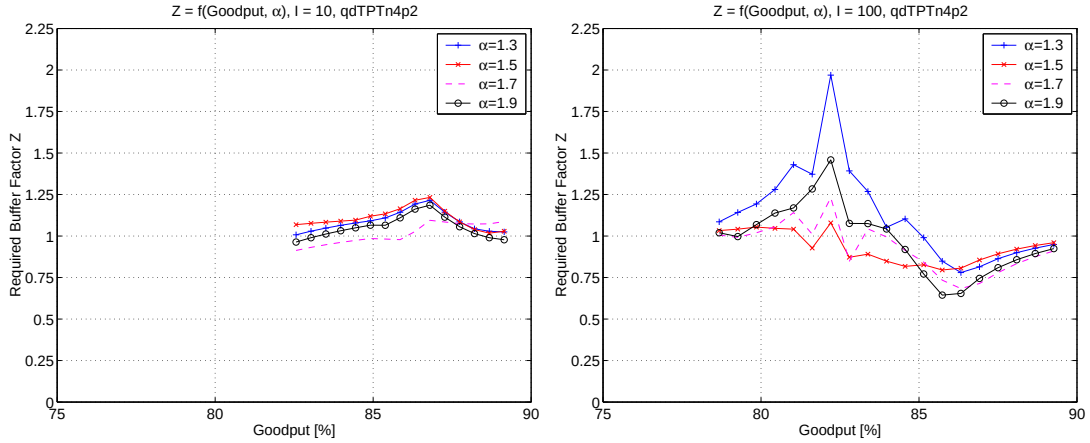


Abbildung 5.3: $Z = f(I, \alpha)$ für $I = 10$ (links) und $I = 100$ (rechts), Frankfurt $\rightarrow$ Köln.

den Goodput bzw. Durchsatz für das Referenz System und die eigentlichen Messungen erzielen lassen. Dies wird jedoch benötigt, da der Pufferbedarf beider Systeme bei gleichem Goodput durcheinander geteilt werden soll. Dieses Problem wird hier durch Interpolation der gemittelten Werte aus jeweils drei Startwerten für den Zufallszahlengenerator gelöst.

Es werden hier die Messungen für die Verkehrsflüsse auf den Links USA $\rightarrow$ Köln (Bild 5.2), Frankfurt $\rightarrow$ München (Bild 5.3) und Frankfurt $\rightarrow$ Köln (Bild 5.4) dargestellt. Im Gegensatz zu den in Abschnitt 5.4.1 gezeigten Ergebnissen ist der Pufferbedarf der beiden Systeme hier sehr ähnlich: die Werte des Faktors Z sind alle im Intervall $[0.6, 2]$.

Die Unterschiede zwischen den Verschiedenen Werten des power-tail Index α sind ebenfalls gering: Die Erwartung, dass mit niedrigerem Wert von α bzw. höherem Grad der Selbstähnlichkeit der Pufferbedarf ansteigt ist nur in einem Teil der Bilder zu erkennen,

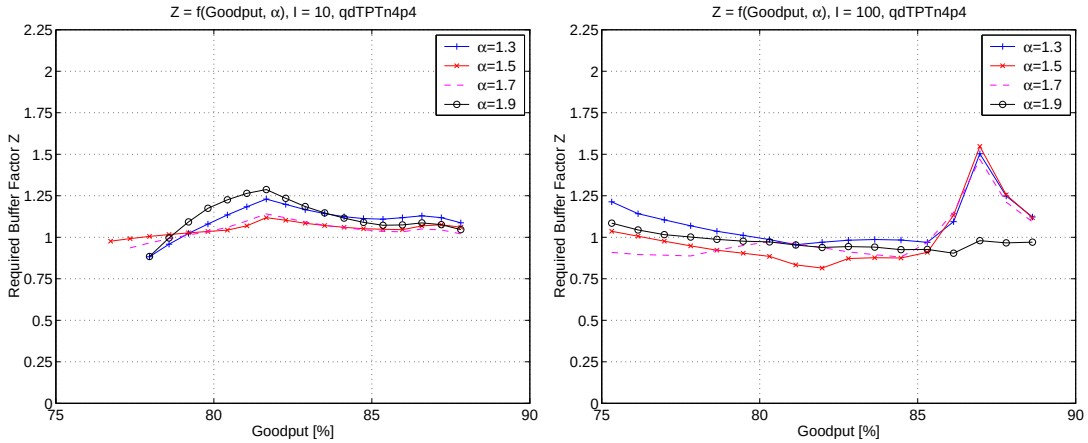


Abbildung 5.4: $Z = f(I, \alpha)$ für $I = 10$ (links) und $I = 100$ (rechts), Frankfurt → München.

eine klare Tendenz lässt sich jedoch nicht generell feststellen. Dies erscheint zunächst sehr verwunderlich, lässt sich jedoch wie folgt erklären: Bei hoher Last sind die Links so stark ausgelastet, dass kaum noch Spielraum für eine Variabilität besteht. Im Extremfall ist, vereinfacht gesagt, jeder Zeitschlitz mit einem Datenpaket benutzt und die Statistik des Verkehrs entspricht einer konstanten Rate. Selbstähnlicher Verkehr jedoch benötigt Variabilität, oder anders herum, nur wo eine entsprechende Variabilität vorhanden ist, kann selbstähnlicher Verkehr entstehen [CCLS01]. Daher hat der power-tail Index α nur geringen Einfluss und daher unterscheidet sich der Pufferbedarf auch nur kaum von dem des Referenzsystems (alle Werte nahe bei eins). Als klarer Trend lässt sich jedoch aus den Bildern ablesen, dass die Z-Werte für steigenden Goodput gegen eins konvergieren. Dieser Fakt stützt die obige Argumentation.

5.4.3 Zusammenfassung

Durch einen Vergleich der Ergebnisse in Abschnitt 5.4.1 und 5.4.2 lässt sich ermitteln, woher der große Puffermehrbedarf des TPT On/Off TCP Systems gegenüber dem M/D/1/B System bei hoher Last herrührt: Da die TPT Verteilung keinen großen Einfluss auf den Pufferbedarf gegenüber einer negativ exponentiellen Verteilung hat (siehe Abschnitt 5.4.2), ist das TCP Protokoll eindeutig als Ursache identifiziert. Dies ist jedoch auch nicht sehr verwunderlich, da TCP ja so entworfen wurde, die Puffer in den Netz-

knoten maximal auszunutzen. Da die Paket-Ankunftszeiten in den Netzwerkknoten bei Verwendung des TCP Protokolls sehr stark korreliert sind, ist auch aus statistischer Sicht dieser Puffermehrbedarf zu verstehen.

Kapitel 6

IP Netzmodell: Ergebnisse

Für jede einzelne Verbindung werden effektiver Datendurchsatz ("Goodput") und Ende zu Ende Verzögerung und deren Standardabweichung innerhalb der virtuellen Pfade zwischen jeweils zwei Knoten gemessen. Diese Werte werden für das gesamte Netz als Durchschnittswerte erhoben. Als Parameter dieser Messungen werden das Verkehrsangebot, parametrisiert durch die Anzahl der “Includes” I, der Hurst-Parameter (bzw. der power-tail Index α) variiert. Die Größe der Ausgangspuffer (pro Link) der Router wurde auf $B = 625$ IP Pakete festgelegt. Für die Simulationen in diesem Kapitel wurde das in Abschnitt 4.3 beschriebene IP-Netzmodell (ohne ATM Layer) verwendet.

6.1 Lasterzeugung

Die Erzeugung einer Lastsituation im gesamten Netz, die so nahe wie möglich an die gemessene Verkehrsmatrix herankommt ist bei der Komplexität des Netzes und dem reaktiven TCP Protokoll keine triviale Aufgabe. Wir haben uns hier darauf beschränkt, bei vorgegebenem Routing und vorgegebenen Linkraten (siehe Abschnitt 4.1) eine Lösung zu finden, die auch unter dem Gesichtspunkt der Simulationsgeschwindigkeit tragbar ist (feste Anzahl von 1000 Quellen). Mit zwei Ansätzen zur Verteilung der Quellen auf die virtuellen Pfade (VPs) wird untersucht, welche Lastsituation sich im Simulationsmodell einstellt.

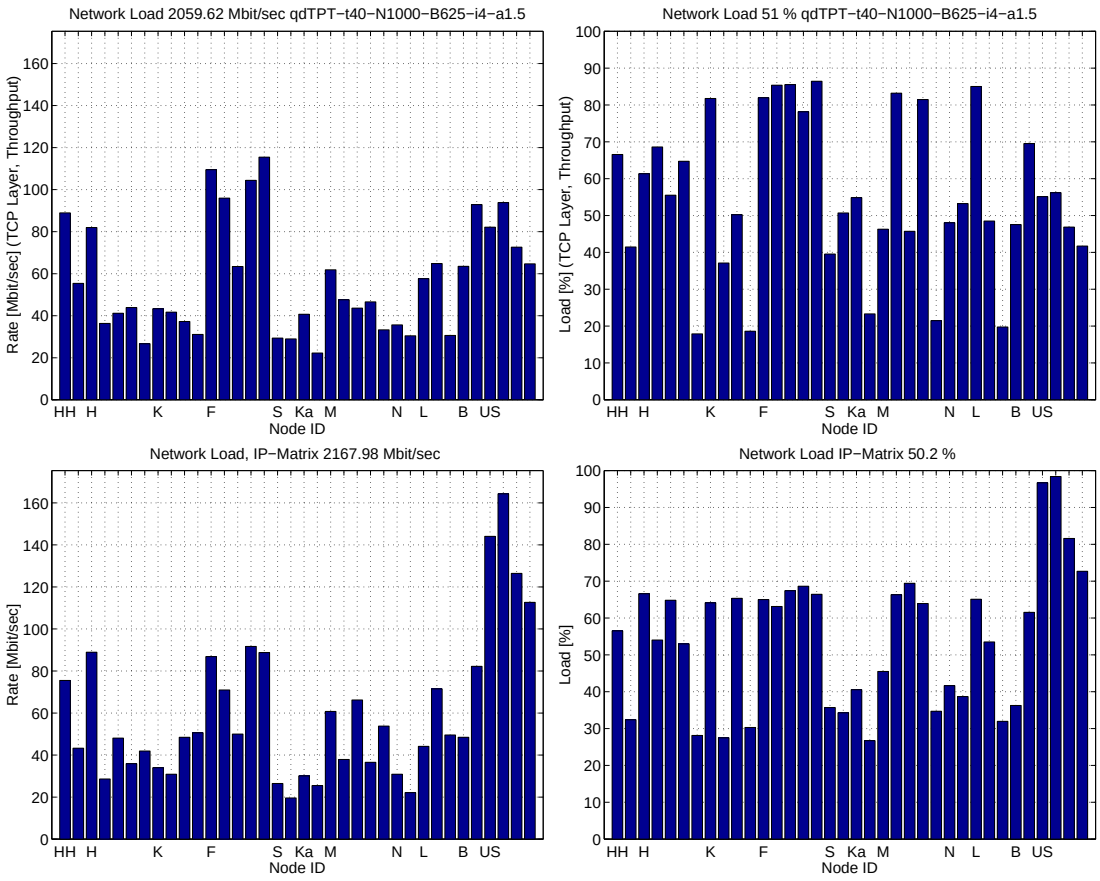


Abbildung 6.1: Auslastung der Ports (oben) im Vergleich zu der ursprünglichen Verkehrsmatrix (unten), Messwerte links in Mbit/s, rechts in Prozent der verfügbaren Bandbreite.

In Bild 6.1 ist die Last des Simulationsmodells (oben) und die gemessene Last im B-WiN (unten, Januar-Februar 2000) dargestellt. Der Vergleich der Bilder zeigt, dass die Lastsituation in dem Simulationsmodell Ähnlichkeit mit den Messwerten aus dem B-WiN besitzt, dass jedoch speziell bei dem Knoten US stärkere Abweichungen zu verzeichnen sind: Im Simulationsmodell wird hier nur eine Last von $42-56\,\%$ erreicht, im B-WiN wurde jedoch eine Last von $72-98\,\%$ gemessen.

Die Ursache der Abweichungen in der Last auf den Links ist ebenfalls bei den Durchsätzen der einzelnen virtuellen Pfade zu sehen (siehe Bild 6.2). Bei allen VPs des Knotens US stellt sich nur ca. 60 % der Rate ein, die in B-WiN gemessen wurde. Die VPs an anderen Knoten, die ebenfalls nur etwa 60 % erreichen, sind die Verbindungen in die USA. Einzelne VPs innerhalb Deutschlands erreichen jedoch z.T. einen deutlich höheren Durchsatz. Diese Abweichungen lassen darauf schließen, dass der Ansatz zur Quellenverteilung

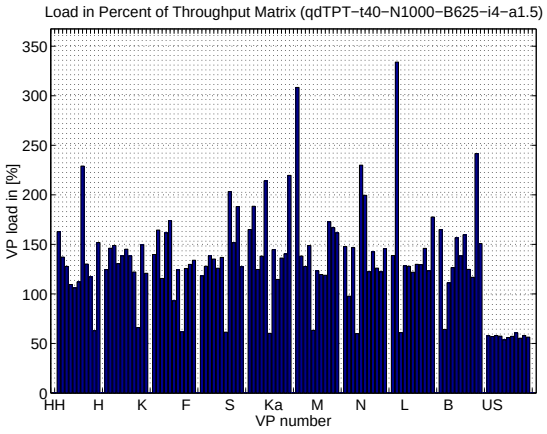


Abbildung 6.2: Durchsatz in Prozent von der gemessenen Verkehrsmatrix, erster Ansatz.

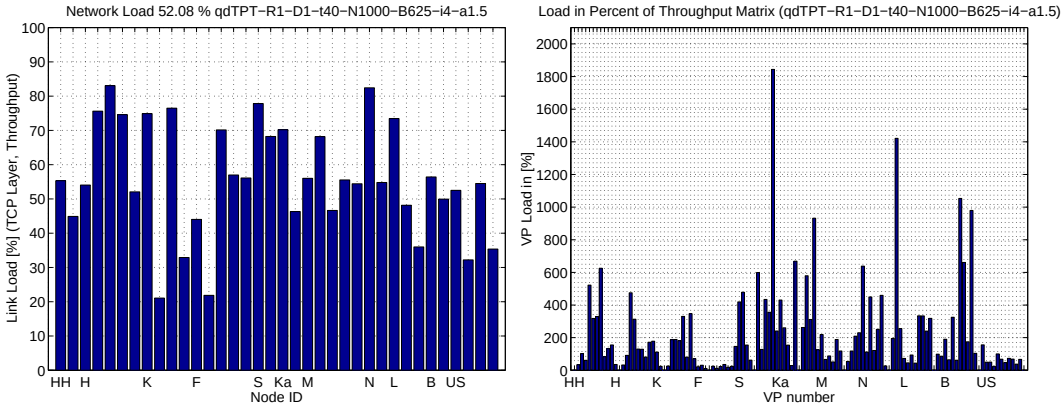


Abbildung 6.3: Last auf den Links (links), Durchsatz in Prozent der gemessenen Verkehrsmatrix (rechts), zweiter Ansatz.

nach Abschnitt 4.2 noch nicht optimal ist. Der Ansatz verwendet die gemessenen Raten, die unterschiedlichen Übertragungszeiten werden jedoch nicht zur Quellenverteilung herangezogen. Das TCP Protokoll reagiert jedoch sehr sensibel auf die Übertragungszeiten bzw. “round trip time”.

Um dieses Problem näher zu untersuchen wird ein zweiter Ansatz zur Bestimmung der Anzahl der Quellen pro VP betrachtet, der eine Linearkombination der Rate und der minimalen Verzögerung in dem jeweiligen virtuellen Pfad nutzt, um die Anzahl der Quellen zu bestimmen (siehe Bild 6.3). Die Abweichungen von gegenüber den Messwerten aus dem B-WiN sind jedoch bei diesem Ansatz noch größer, obwohl ein genaueres Modell zugrunde gelegt wird. Daher wird im Folgenden nur der erste Ansatz verwendet.

6.2 Verschiedene Lastfälle

In diesem Abschnitt wird der Einfluss des Verkehrsangebots (Anzahl der "Includes" I) und des power-tail Index α auf den Goodput und die Ende-zu-Ende Verzögerung aufgezeigt. Bild 6.4 zeigt den Goodput auf den Links (links) sowie die mittlere Ende-zu-Ende Verzögerung pro virtuellem Pfad für die Werte $\alpha = 1.3, 1.5$ und 1.9[1]. Es ist deutlich zu erkennen, dass die Verbindungen von bzw. in die USA deutlich höhere Ende-zu-Ende Verzögerungen haben, es ist sind jedoch kaum Unterschiede bezüglich der verschiedenen α-Werte sichtbar.

Bild 6.5 zeigt die gleichen Graphen für hohe Last ($I = 100$ "Includes"). Auch hier ist kaum ein Unterschied zwischen den verschiedenen Werten von α auszumachen, jedoch sind die Ende-zu-Ende Verzögerungen auch innerhalb Deutschlands stärker gewachsen als die Verzögerungen von bzw. zu Knoten US. Dadurch sind die Verzögerungen jetzt fast in gleicher Größenordnung.

Beim Vergleich der Bilder 6.4 und 6.5 fällt neben der frappierenden Ähnlichkeit der Bilder für unterschiedliche α-Werte auf, dass sich die Struktur für niedrige und hohe Lasten (33% bzw. 77%) stark unterscheidet: Bei niedrigen Lasten schwanken die Auslastungen für die unterschiedlichen IP-Flüsse stark, wohingegen die Verzögerungen relativ gleich ausfallen - mit Ausnahme der Verbindungen, die vom Knoten US ausgehen und eine hohe Laufzeitverzögerung aufweisen. Bei hohen Lasten gleichen sich die Auslastungen an, wohingegen die Schwankungen der Verzögerungen zunehmen. Für ein festes a von $\alpha = 1.5$ ist diese Entwicklung mit den zusätzlichen Werten von $I = 4, 5.55, 10$ zusammen mit Graphen für die Standardabweichung in Bild 6.6 dokumentiert. Die Standardabweichung der Ende-zu-Ende Verzögerung ist in allen Fällen innerhalb Deutschlands wesentlich höher, als bei dem Knoten USA, da dort schon eine hohe Leitungsverzögerung die Variabilität der Werte einschränkt.

Die Mittelwerte der gemessenen Größen für das gesamte Netz sind in Abhängigkeit von α und I in der Tabelle 6.1 zusammengefasst:

[1] Die leeren Balken bei der Ende-zu-Ende Verzögerung stehen für die Verzögerung vom Knoten zu sich selbst.

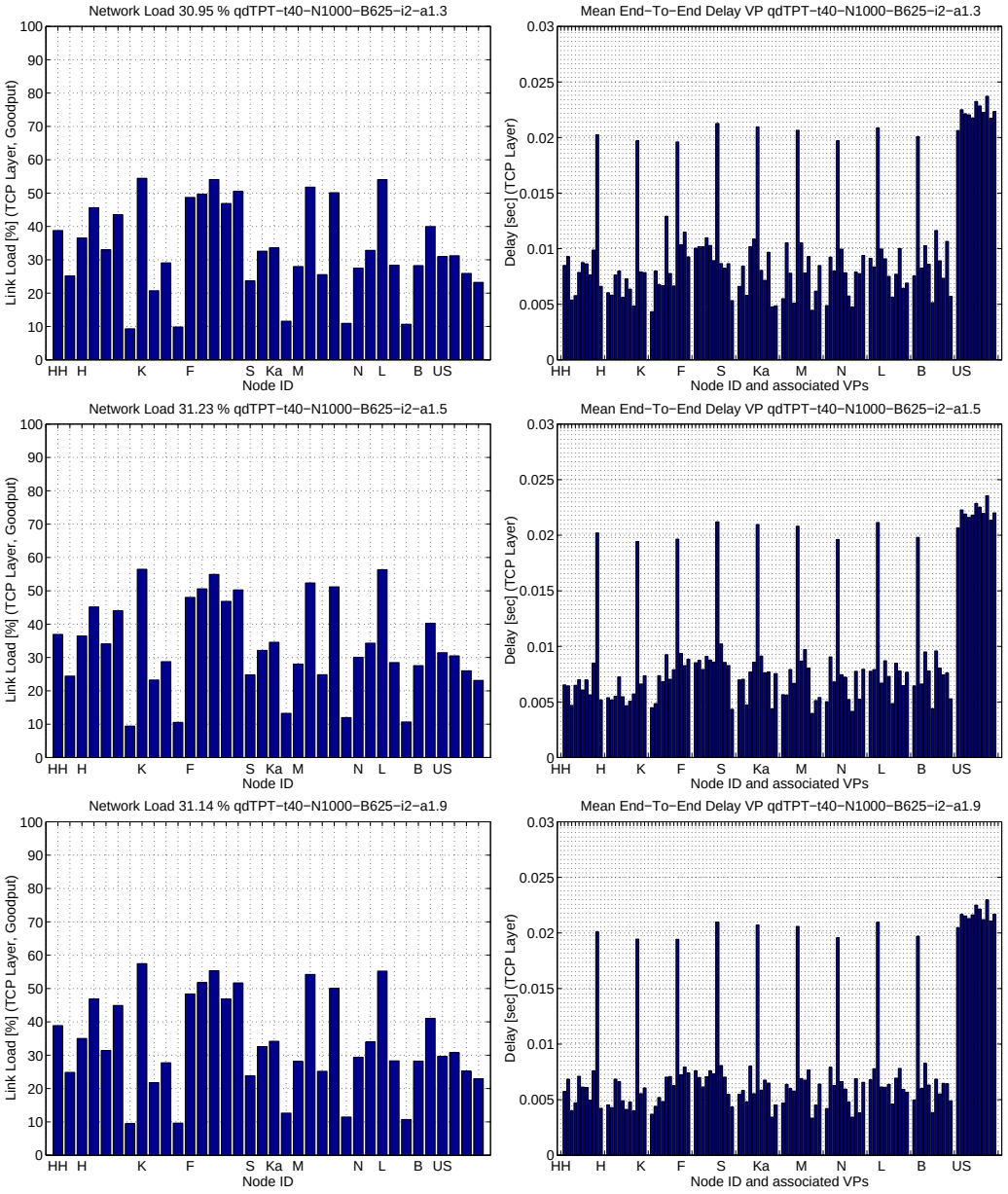


Abbildung 6.4: Goodput und Ende-zu-Ende Verzögerung für $I = 2$, $\alpha = 1.3$ (Oben), $\alpha = 1.5$ (Mitte), $\alpha = 1.9$ (Unten).

6.3 Aufteilung der Raten

In Bild 6.7 ist der Durchsatz der einzelnen TCP Verbindungen auf einem Port für verschiedene Lasten ($I = 2, 4, 10, 100$) dargestellt. Die Bezeichnungen “4/7a” bedeuten hier, dass in diesem VP *aktive* TCP Verbindungen von Knoten 4 (Frankfurt) nach Knoten 7 (München) betrachtet werden. Bei niedrigen Lasten ($I = 2, 4$) sind die Verbindungen im VP in Richtung USA (“4/11a”) deutlich durch die größere Verzögerung benachteiligt,

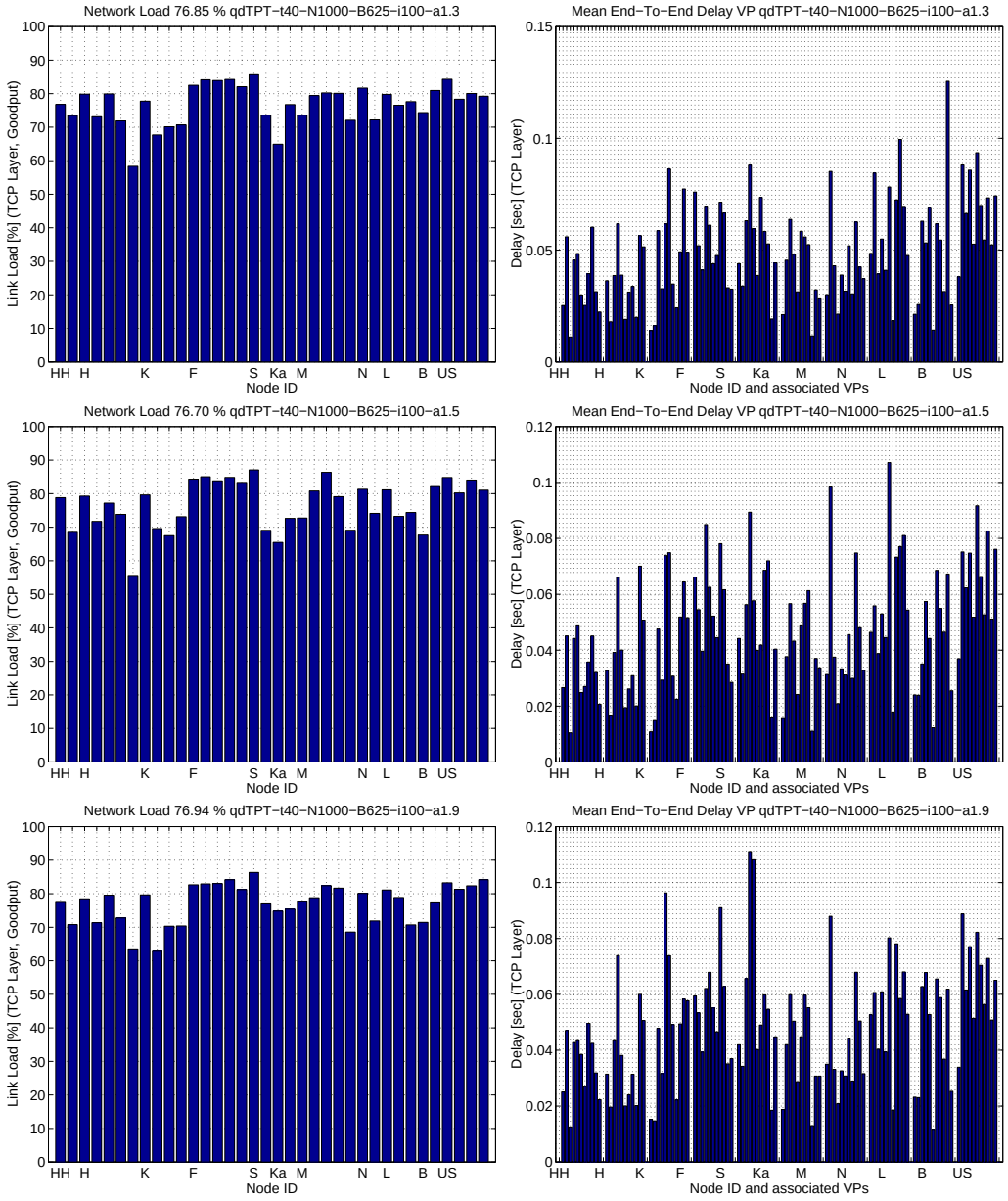


Abbildung 6.5: Goodput und Ende-zu-Ende Verzögerung für $I = 100$, $\alpha = 1.3$ (Oben), $\alpha = 1.5$ (Mitte), $\alpha = 1.9$ (Unten).

der erzielte Durchsatz ist nur etwa halb so hoch, wie der Durchsatz der anderen, innerhalb Deutschlands verlaufenden Verbindungen. Bei hoher Last wandelt sich jedoch wieder das Bild: der VP “4/7a” kann mit deutlich höherem Durchsatz dominieren. Dies ist die einzige Verbindung über diesen Port, die nur über einen Hop geht, und damit auch nur in einer Warteschlange warten muss und Paketverluste erleidet. Innerhalb der VPs erzielen die einzelnen Quellen wie erwartet einen ähnlichen Durchsatz, die Abweichungen sind besonders bei Verbindungen mit kurzen Verzögerungen jedoch recht hoch (siehe

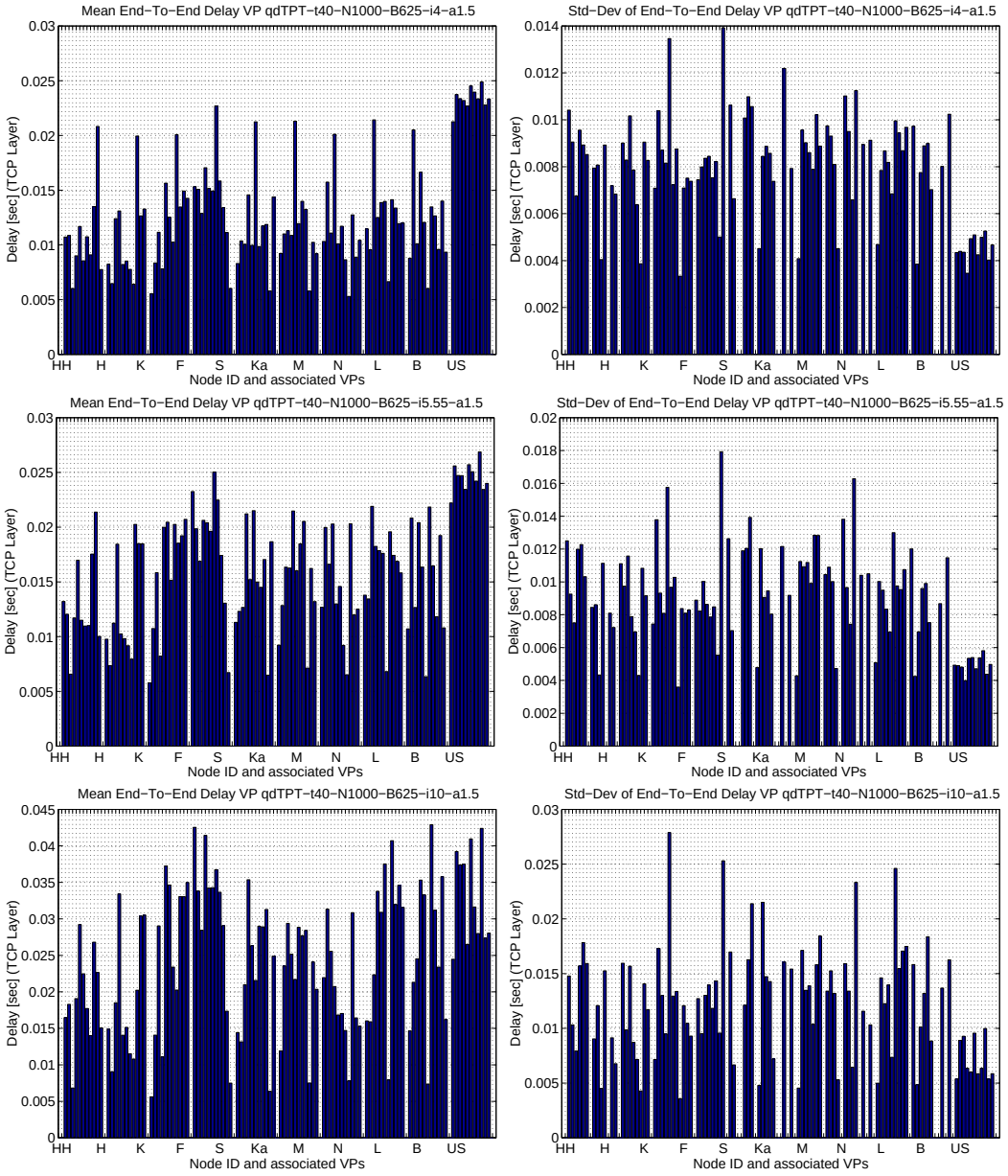


Abbildung 6.6: Mittlere Ende-zu-Ende Verzögerung (links) und deren Standardabweichung (rechts) für $\alpha = 1.5$, $I = 4$ (Oben), $I = 5.55$ (Mitte) bzw. $I = 10$ (Unten).

z.B. VP “4/7a”), was der Modellvorstellung einer fairen Bandbreitenaufteilung zwischen TCP/IP Quellen widerspricht.

Tabelle 6.1: Mittelwerte der Leistungskenngrößen im IP Netzmodell; G: Goodput, T: Throughput, R: Paket-Verlustrate in %, μ_D: mittlere Ende-zu-Ende Verzögerung, σ_D: Standard Abweichung der Ende-zu-Ende Verzögerung bei einer Puffergröße von $B = 625$ IP Paketen.

I	α	G	T	R	μ_D	σ_D
2	1.3	30.95	33.09	0.00	9.4	6.4
	1.5	31.23	33.40	0.00	8.7	5.6
	1.7	31.28	33.45	0.00	8.3	5.1
	1.9	31.14	33.34	0.00	7.9	4.5
	exp.	30.95	33.12	0.00	7.0	2.9
4	1.3	50.59	53.78	0.00	12.4	7.8
	1.5	50.88	54.08	0.00	11.9	7.4
	1.7	51.05	54.27	0.00	11.5	7.0
	1.9	51.07	54.29	0.00	11.3	6.8
	exp.	51.09	54.34	0.00	10.6	6.0
5.55	1.3	57.90	61.45	0.00	15.0	8.6
	1.5	58.45	62.04	0.00	14.7	8.2
	1.7	58.55	62.14	0.00	14.5	8.1
	1.9	58.48	62.08	0.00	14.3	8.0
	exp.	58.53	62.14	0.00	13.8	7.5
10	1.3	66.89	70.94	0.03	22.0	13.7
	1.5	66.75	70.85	0.04	22.4	16.7
	1.7	66.94	70.98	0.04	22.3	15.6
	1.9	66.94	70.93	0.03	22.1	14.6
	exp.	66.88	70.90	0.04	21.9	15.2
100	1.3	76.85	84.15	0.57	44.2	64.7
	1.5	76.70	83.72	0.56	42.6	57.8
	1.7	76.89	83.96	0.56	43.9	61.8
	1.9	76.94	84.18	0.56	43.5	61.6
	exp.	77.20	84.34	0.56	43.1	58.1

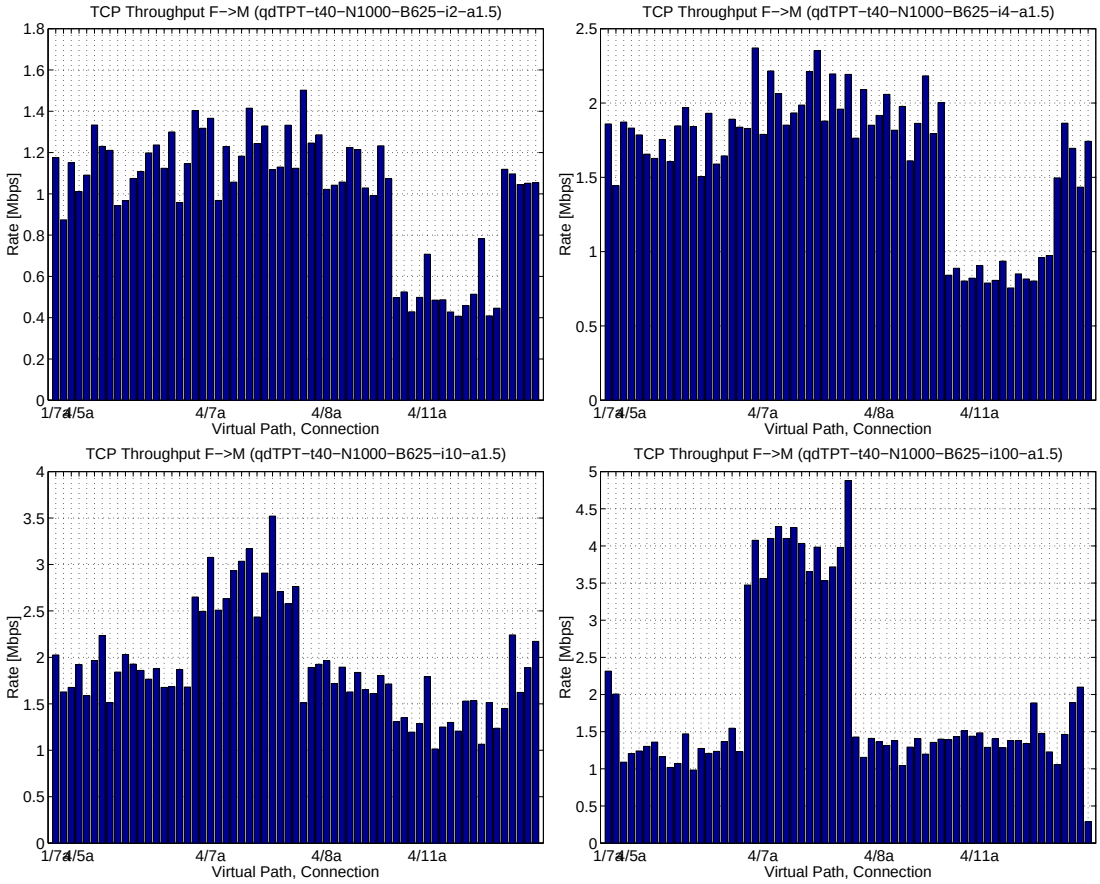


Abbildung 6.7: Durchsatz individueller TCP Verbindungen am Knoten Frankfurt, Port 4 (München) für $I = 2$ (oben links), $I = 4$ (oben rechts), $I = 10$ (unten links) und $I = 100$ (unten rechts).

Kapitel 7

Einfluss von Early Packet Discard (EPD)

Die Spezifikation von UBR sieht die Möglichkeit eines "Early Packet Discard" vor: Damit ist gemeint, dass die Bedienstrategie eines Puffers unterhalb einer Schwelle FIFO ist, während oberhalb der Schwelle der Pufferbelegung wie folgt verfahren wird:

Es wird das PTI-Bit in den ATM Zellen inspiziert, das angibt wann ein AAL5-Rahmen (hier entsprechend einem IP Paket) endet. Wird die letzte Zelle eines IP Paketes gefunden, so werden alle weiteren Zellen mit demselben VCI solange verworfen, bis der Schwellwert wieder unterschritten wird. Damit werden tendenziell immer ganze IP Pakete (und nicht etwa deren Bruchstücke) verworfen. Das Netz wird also von sinnlosem Verkehr freigehalten.

Der Fragestellung, wie groß der Vorteil von EPD ist, wurde durch einen Vergleich nachgegangen, indem dieselbe Verkehrslast einmal für ein Netz ohne EPD und einmal für ein Netz mit EPD erzeugt wurde. Dabei wurde eine virtuelle Vollvermaschung der ATM-Knoten mit Hilfe des UBR Dienstes modelliert. Diese Art der Modellierung erscheint zwingend, wenn man die Leistungsfähigkeit von EPD erfassen will, denn wenn im Gegensatz dazu hop-by-hop Routing eingesetzt werden würde, würde der erste Router nach dem Auftreten eines Zellenverlustes bereits einen Assemblierungsfehler feststellen und das entsprechende IP Paket komplett verwerfen. Insofern liefe die Einführung der EPD-

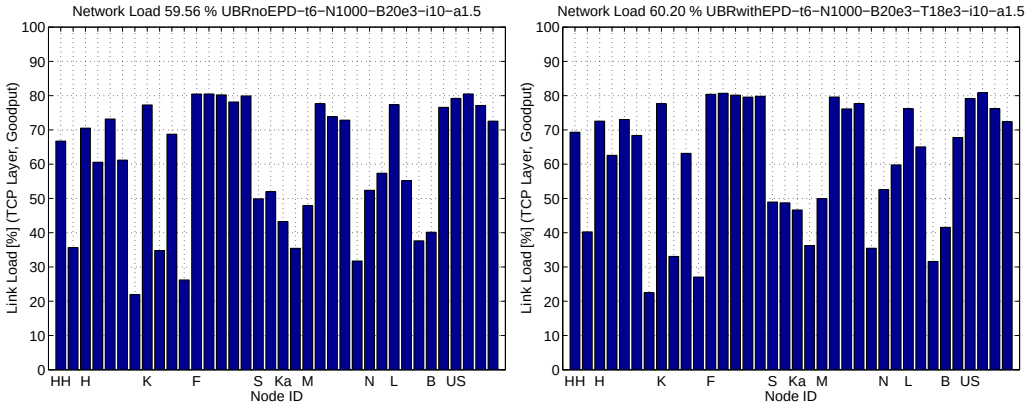


Abbildung 7.1: Goodput, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 10$.

Option nur darauf hinaus, dass der Assemblierungsfehler in einem ATM Switch und nicht in dem unmittelbar angeschlossenen Router erkannt würde.

Für die Simulationen in diesem Kapitel wurde das in Abschnitt 4 beschriebene Netzmodell mit ATM Schicht verwendet. Da direkte ATM-Verbindungen von Quelle zu Empfänger geschaltet wurden, wurde kein IP-Router verwendet. Die Simulationsexperimente wurden für einen festen Parameter $\alpha = 1.5$ durchgeführt. Die Puffer in den ATM Switches waren durchgängig für Ausgangswarteschlangenlängen von 20000 ATM-Zellen dimensioniert, wobei der EPD-Schwellwert auf 18000 ATM-Zellen gesetzt wurde. Auf dieser Basis wurden Experimente durchgeführt, die den effektiven Durchsatz und die mittlere Ende-zu-Ende Verzögerung sowie deren Standard-Abweichung für jeden virtuellen Pfad zwischen jeweils zwei Knoten als Funktion des Angebots bestimmen.

Die Bilder 7.1, 7.2 und 7.3 zeigen den Goodput auf den Links, die Ende-zu-Ende Verzögerung der TCP-Pakete sowie deren Standard-Abweichung für einen Wert von $I = 10$. Links ist jeweils das Ergebnis ohne EPD, rechts das Ergebnis mit EPD zu sehen. In ähnlicher Weise geben die Bilder 7.4, 7.5 und 7.6 die Verhältnisse für $I = 100$ wieder. Für kleine Lasten ($I < 10$) werden keine signifikanten Unterschiede in den Ergebnissen festgestellt, da dort kaum hohe Pufferfüllstände oder Verluste auftreten.

Man erkennt, dass für die Leistungsmaße (Goodput, mittlere Ende-zu-Ende Verzögerung, Standardabweichung der mittleren Ende-zu-Ende Verzögerung) eine leichte Verbesserung durch den Einsatz von EPD erzielt wurde: der Goodput steigt bei Verwendung von EPD leicht an und sowohl die mittlere Ende-zu-Ende Verzögerung als auch die Standardabwei-

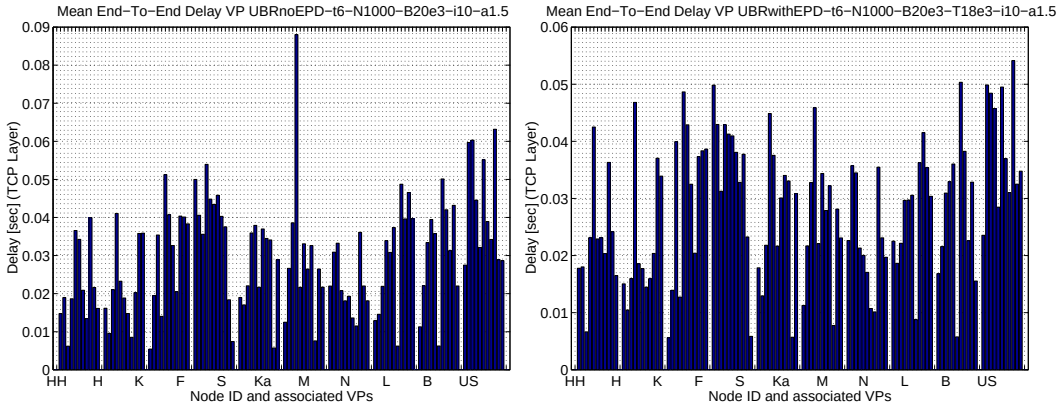


Abbildung 7.2: Mittlere Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5$, $I = 10$ (man beachte die unterschiedlich skalierten Ordinaten).

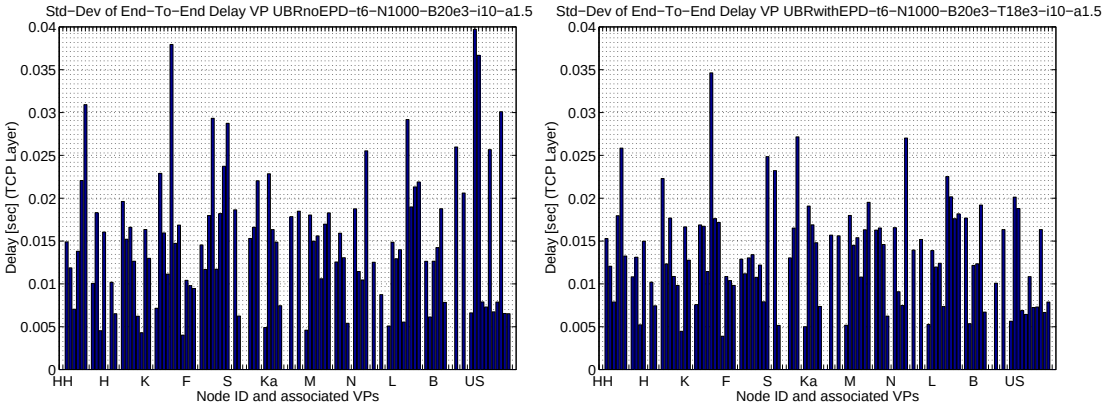


Abbildung 7.3: Standardabweichung der Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5$, $I = 10$.

chung der mittleren Ende-zu-Ende Verzögerung sinken bei Verwendung von EPD leicht ab. Dieser Effekt ist bei hoher Last ($I = 100$) erwartungsgemäß stärker ausgeprägt als bei etwas niedrigerer Last ($I = 10$): Der Goodput steigt bei hoher Last von 63.7 % auf 67.6 %, bei niedrigerer Last jedoch nur unwesentlich von 59.6 % auf 60.2 %.

Diese Ergebnisse legen nahe, bei Verwendung von UBR auch die Strategie EPD zu nutzen, da weniger Fragmente von IP Paketen übertragen werden und damit ein etwas größerer Goodput erzielt werden kann und die Verzögerungen reduziert werden können, da nicht so viele IP Pakete aufgrund der Verluste erneut übertragen werden müssen. Eine Umrüstung bestehender Hardware auf die EPD Option dürfte allerdings in der Regel nicht wirtschaftlich sein.

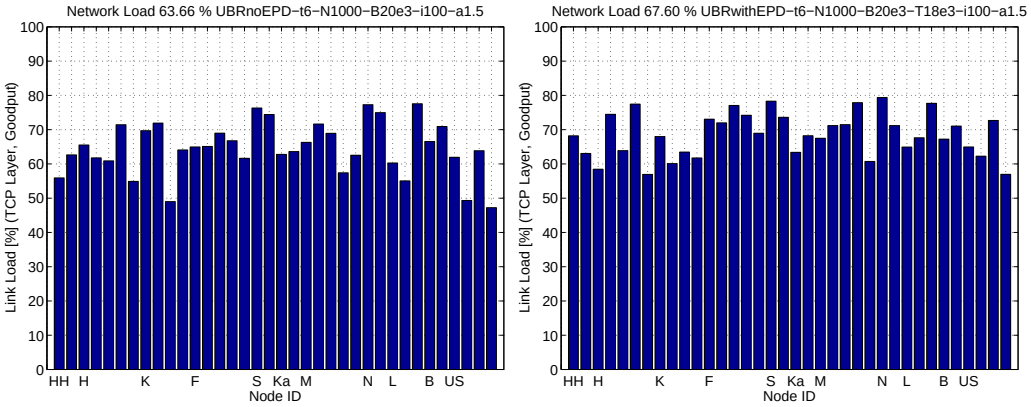


Abbildung 7.4: Goodput, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 100$.

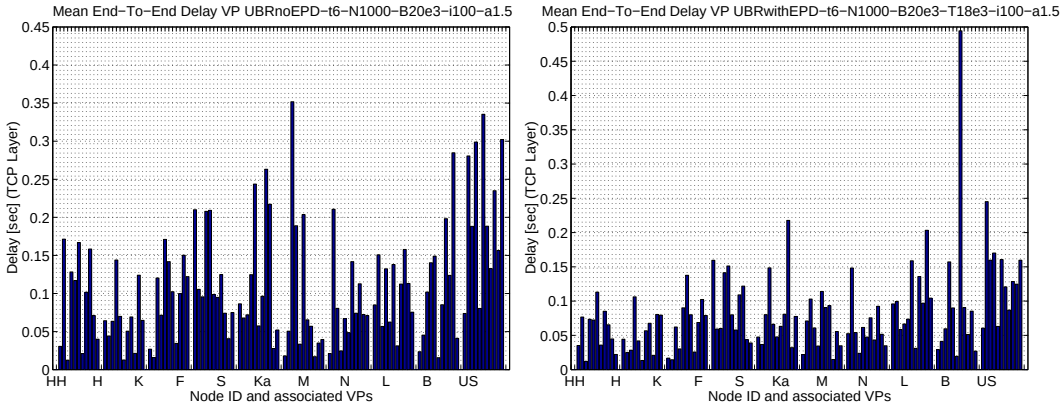


Abbildung 7.5: Mittlere Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 100$.

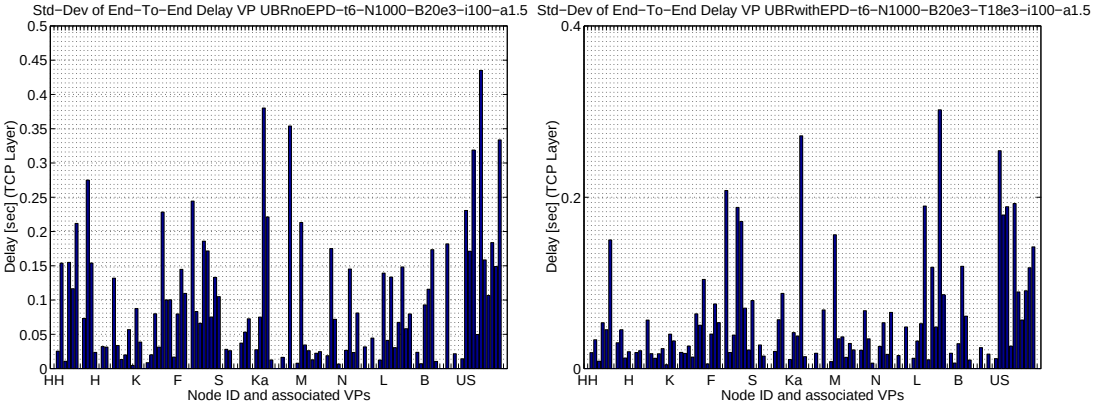


Abbildung 7.6: Standardabweichung der Ende-zu-Ende Verzögerung, ohne EPD (links) und mit EPD (rechts), für $\alpha = 1.5, I = 100$.

Kapitel 8

Netzmodell mit ABR Regelung

8.1 Modellierung

Diesem Szenario liegt das in Abschnitt 4 beschriebene Simulationsmodell zugrunde, das nunmehr um die ABR-Fähigkeit der Knoten ergänzt wurde. Es wurde in diesem Szenario eine Variante mit “hop-by-hop” Routing auf IP Ebene modelliert, wo in jedem Knoten ein IP-Router angeschlossen war. Dies entspricht weitgehend der realen Situation im B-WiN. In Abschnitt 2.3.2 wurde bereits der "Explicit Rate Indication for Congestion Avoidance (ERICA)" Algorithmus vorgestellt. Die jetzt vorliegende Implementierung geht über diesen Sachstand hinaus, indem der ERICA+ Algorithmus implementiert wurde. Der ERICA Algorithmus macht von einer ABR Zielrate Gebrauch, die wie folgt berechnet wird (vgl. Gleichung 2.3):

$$ABR_{Zielrate} = U \cdot ABR_{Kapazitaet} \tag{8.1}$$

Dabei ist der Wert U frei wählbar und wird typischerweise zwischen 0.9 und 0.95 angesetzt. Der ERICA Algorithmus neigt aber dazu, bei stark schwankendem Hintergrundverkehr instabil zu werden. ERICA+ bietet weitgehend Abhilfe und erreicht dies durch einen Nutzungsfaktor U, der von der Pufferbelegung Q, einem Sollwert Q_0 und drei Konstanten a und b und $QDLF$ abhängt. Die verwendete Kennlinie für den Verlauf von $U = f(Q, Q_0)$

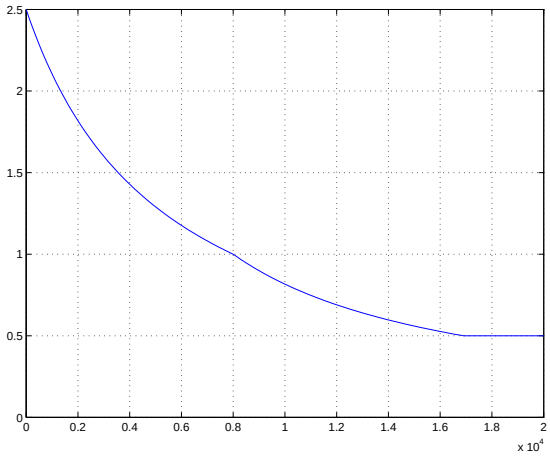


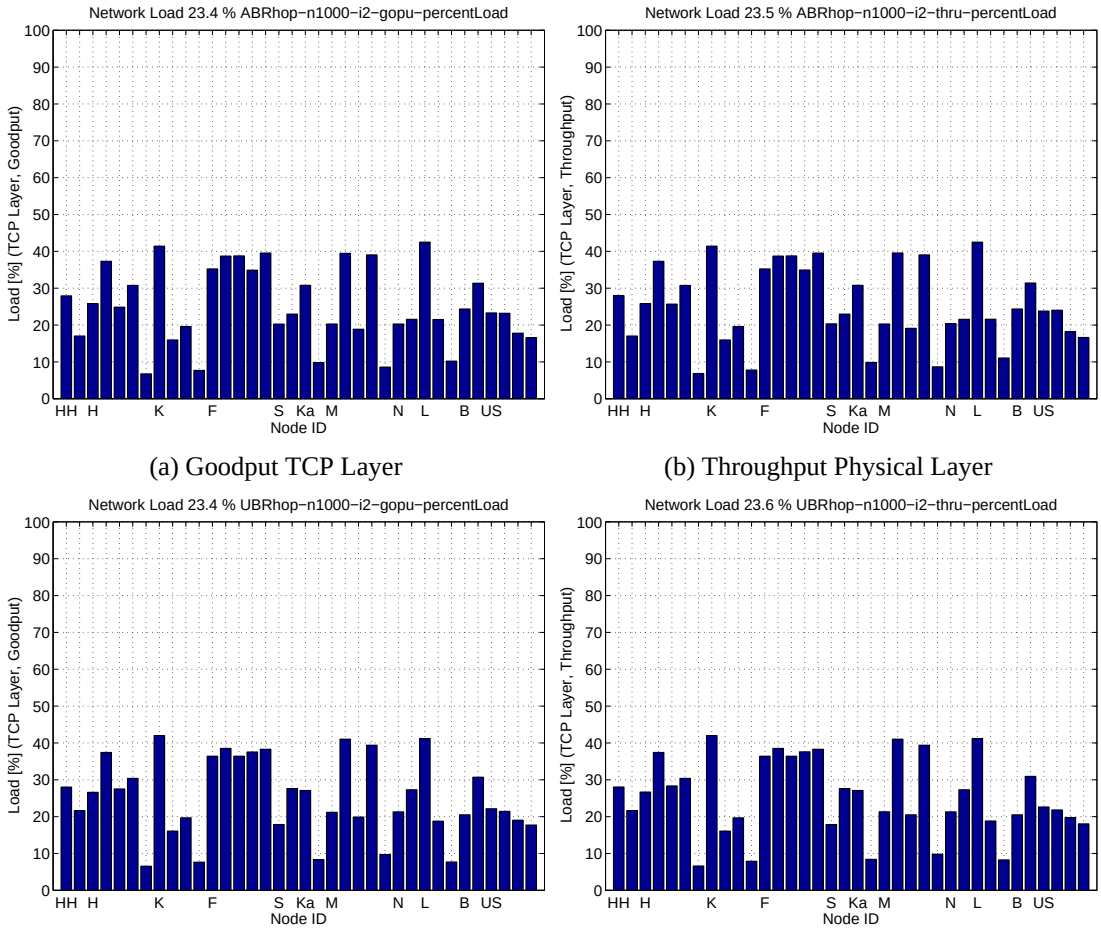
Abbildung 8.1: Verwendete Kennlinie des Nutzungsfaktors $f(Q,Q_0)$ bei ERICA+.

ist in Bild 8.1 zu sehen. U ist eine monoton fallende Funktion zwischen $b = 2.5$ und $QDLF = 0.5$.

Für $Q = Q_0$ wird der Wert 1 erreicht. Der Effekt dieses Verlaufs von U ist, dass der Lastfaktor z ($z \sim 1/U$) für $Q < Q_0$ abgesenkt und für $Q > Q_0$ angehoben wird. Im Vergleich zu ERICA erhöht sich daher die explizite Rate ($ER \sim U$) für $Q < Q_0$, bzw. sie erniedrigt sich entsprechend für $Q > Q_0$. Damit hat ERICA+ die Tendenz einen Sollfüllstand der Puffer von Q_0 zu stabilisieren und damit zu einer konstanten Auslastung der Verbindungsleitungen beizutragen.

8.2 Ergebnisse der Simulation

Die Ergebnisse sind in den Bildern 8.2 bis 8.6 zusammengestellt. Sie zeigen jeweils Goodput und Durchsatz für verschiedene Verkehrslasten, die – wie schon in den vorangegangenen Kapiteln – durch die Anzahl der HTTP “Includes” I eingestellt werden. Für alle Experimente wurde $\alpha = 1.5$ gesetzt. Jedes Bild zeigt im oberen Teil die Ergebnisse für eine auf ABR beruhende Verbindung zwischen den Netzknoten während der untere Teil des Bildes die UBR Ergebnisse widergibt. Alle Bilder zeigen ein sehr ähnliches Verhalten, ein Unterschied zwischen ABR und UBR ist bei diesen Messwerten kaum auszumachen. Der Vollständigkeit halber werden dennoch alle Graphen hier gezeigt.

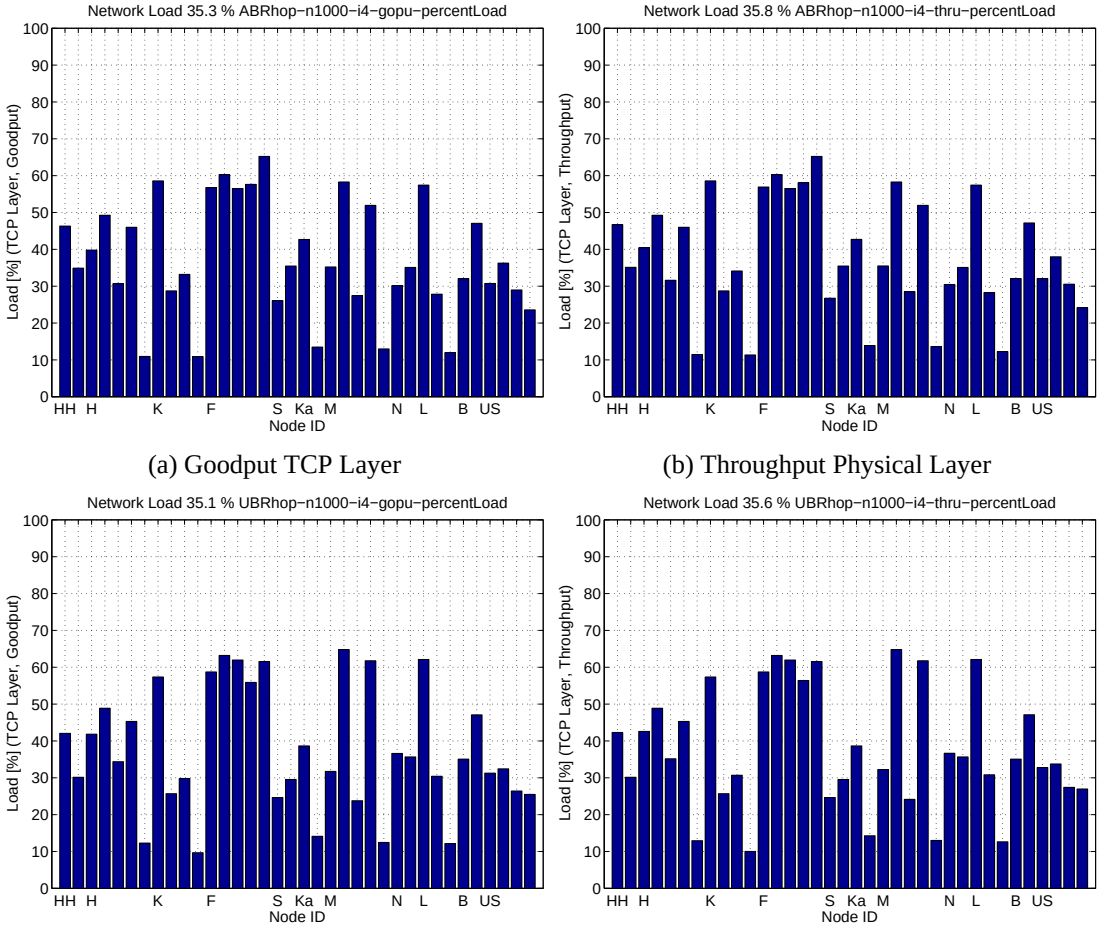


(a) Goodput TCP Layer

(b) Throughput Physical Layer

Abbildung 8.2: Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5$, $I = 2$.

Gemittelt über das gesamte Netz sind die Ergebnisse in Tabelle 8.1 zusammengefasst: Es zeigt sich wider Erwarten, dass ABR kaum dazu beiträgt den Goodput zu erhöhen sondern zum Teil sogar einen niedrigeren Goodput erzielt. Allerdings ist dieser Effekt bei niedrigerer Last weniger ausgeprägt. Der Hintergrund für dieses Verhalten ist darin zu suchen, dass sich hier zwei Regelsysteme gegenseitig beeinflussen: Wenn die ABR-Regelung die Senderate einer Quelle begrenzt, so hat die damit einhergehende verlangsamte Öffnung des “congestion window” einen verstärkenden Effekt (im Vergleich zum UBR-Fall). Dagegen wird bei einer Vergrößerung der verfügbaren Rate durch ABR die Nutzung nur innerhalb der Vorgaben der Signalumlaufzeit innerhalb der “slow start” oder der “congestion avoidance” Phase erfolgen. Insofern ist damit zu rechnen, dass die Nutzung der von ABR zur Verfügung gestellten Restkapazitäten durch den doppelte Regelmechanismus

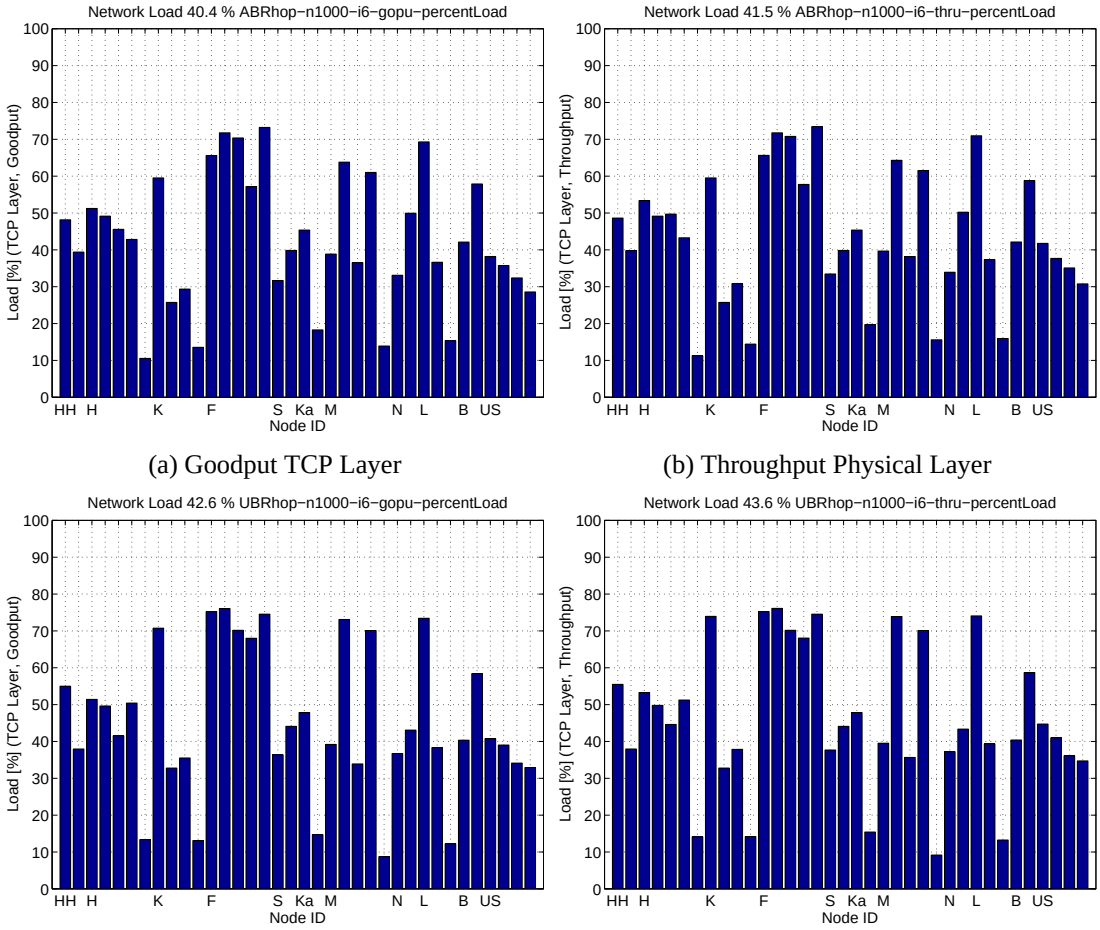


(a) Goodput TCP Layer

(b) Throughput Physical Layer

Abbildung 8.3: Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5$, $I = 4$.

von ABR und TCP eher wieder geschmälert wird. Es ist jedoch bei hoher Last ($I = 100$) an den etwas höheren Werten von G_{ABR}/T_{ABR} gegenüber G_{UBR}/T_{UBR} zu erkennen, dass ABR bei hoher Last weniger Verluste erzeugt und damit effektiver arbeitet. Kritisch sei angemerkt, dass bei noch höheren als den hier erzielten Durchsätzen aufgrund einer ansteigenden Effektivität ein besserer Goodput für ABR als für UBR nicht ausgeschlossen werden kann. Die vernachlässigbaren Unterschiede der Leistungsmaße bei niedrigen Lasten erklären sich einfach daraus, dass hier der ABR Mechanismus nur zum Teil eingreift.



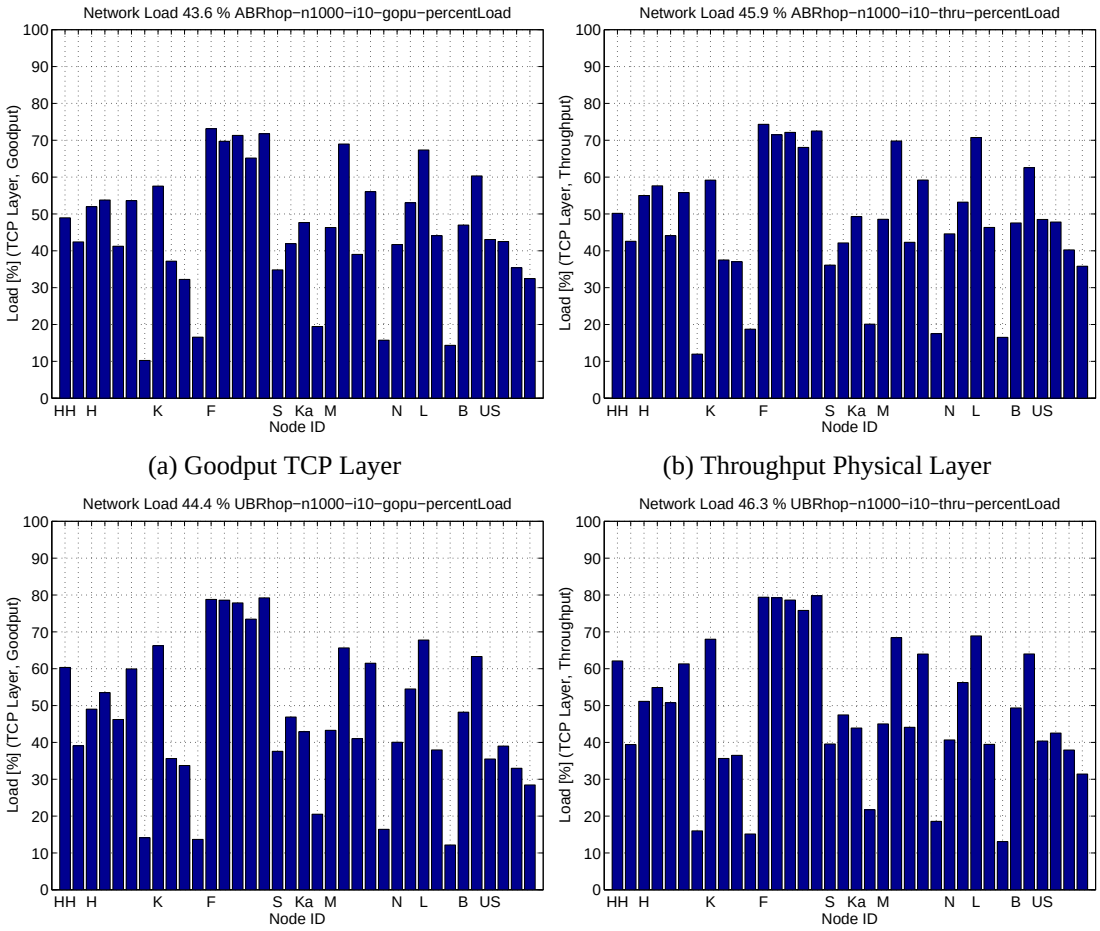
(a) Goodput TCP Layer

(b) Throughput Physical Layer

Abbildung 8.4: Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5$, $I = 6$.

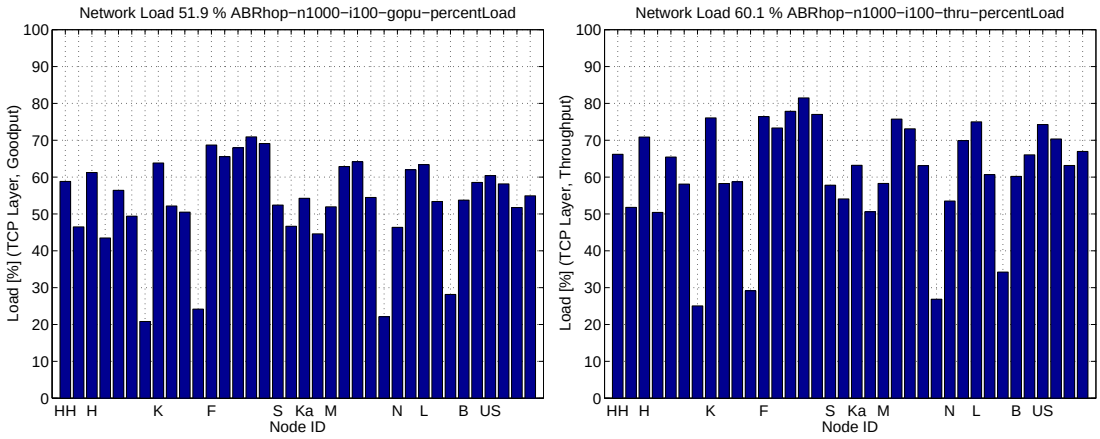
Tabelle 8.1: Mittlerer Goodput G und Durchsatz T im UBR- und ABR-Fall in Prozent in Abhängigkeit von der Anzahl der HTTP “Includes” I.

I	G_{UBR}	T_{UBR}	G_{UBR}/T_{UBR}	G_{ABR}	T_{ABR}	G_{ABR}/T_{ABR}
2	23.41	23.57	0.993	23.37	23.52	0.994
4	35.14	35.56	0.988	35.35	35.81	0.987
6	42.63	43.58	0.978	40.43	41.51	0.971
10	44.42	46.31	0.959	43.60	45.89	0.950
100	52.69	62.09	0.849	51.95	60.14	0.864



(a) Goodput TCP Layer (b) Throughput Physical Layer

Abbildung 8.5: Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5$, $I = 10$.



(a) Goodput TCP Layer (b) Throughput Physical Layer

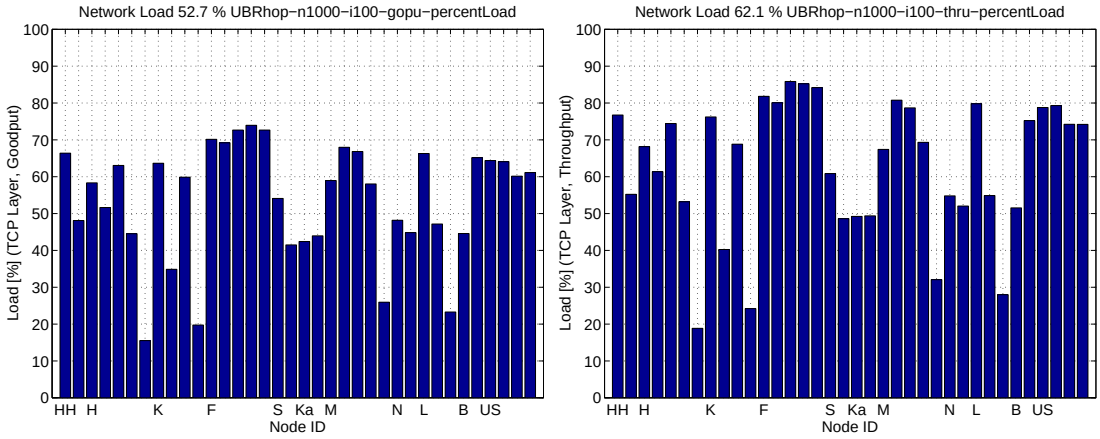


Abbildung 8.6: Goodput und Durchsatz auf den Links mit ABR (oben) und UBR (unten); $\alpha = 1.5$, $I = 100$.

Kapitel 9

Einfluss von dynamischem Routing

Betrachtet man das Konzept von IP über ATM, so wird als hervorragende Eigenschaft gerne herausgestellt, das alle IP berührenden Mechanismen unverändert erhalten bleiben. Insbesondere findet ein Hop-by-Hop Routing zwischen den Routern statt, das allerdings auch als Nachteil der Methode betrachtet werden kann, weil statt des schwerfälligen Routens über Router ein Bypass auf ATM Ebene den Endgeräten direkt die gewünschte Konnektivität verschaffen könnte. In diese Richtung zielten Ansätze wie MPOA [af-97] oder die verschiedenen Varianten von MPLS [RVC].

In diesem Kapitel wird der Frage nachgegangen, was grenzwertmässig mit einer direkten Konnektivität auf ATM-Ebene erreicht werden kann. Die Vorstellung in diesem Modell ist also, dass von den Endgeräten IP Pakete versandt werden, die Teil einer TCP Verbindung (On-Zeit des TCP Quellenmodells) sind. Mit jeder neuen Verbindung erfolgt ein Rufaufbau und eine Wegesuche, wobei für die Wegesuche die in Abschnitt 2.1.2 eingeführte Metrik bestimmend ist. Die Wegesuche selbst erfolgt nach dem Dijkstra Algorithmus. Für die Simulationen in diesem Kapitel wurde das in Abschnitt 4 beschriebene Netzmodell mit ATM Layer verwendet. Da direkte ATM-Verbindungen von Quelle zu Empfänger geschaltet werden, wird kein IP-Router verwendet.

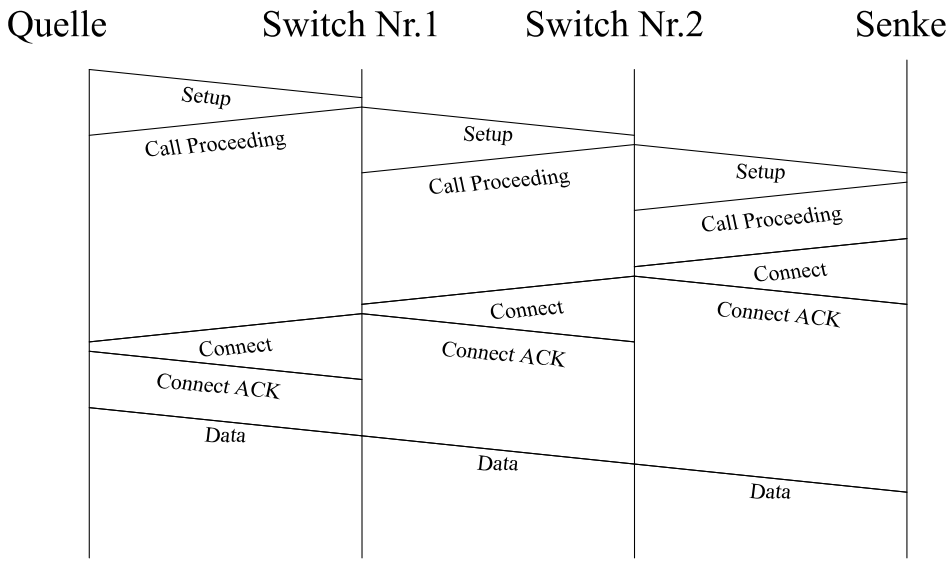


Abbildung 9.1: Schematische Darstellung des Verbindungsaufbaus.

9.1 Verbindungsaufbau und -abbau im ATM Netz

In Bild 9.1 ist der Verbindungsaufbau schematisch dargestellt. Die Quelle sendet zur Einleitung des Verbindungsaufbaus die UNI-Nachricht SETUP zum ersten Switch. Der leitet diese Nachricht seinerseits weiter zum nächsten Switch und bestätigt den Erhalt der Setup-Nachricht mit der UNI-Nachricht CALL PROCEEDING. Der nächste Switch sendet die Setup-Nachricht weiter in Richtung der Senke und quittiert dem Vorgängerswitch den Erhalt der Setup-Nachricht. Die Setup-Nachricht wird solange durch das Netz weiter geleitet, bis sie die Senke erreicht hat. Die Senke bestätigt, genau wie die Switches, den Erhalt der Setup-Nachricht und schickt ihrerseits die UNI-Nachricht CONNECT zur Quelle zurück um zu bestätigen, dass sie die Verbindung ebenfalls aufbauen möchte. Während sich die Connect-Nachricht ihren Weg durch das ATM-Netz sucht, wird diese Nachricht, genau wie die Setup-Nachricht, von jedem Switch bestätigt und zwar durch die UNI- bzw. NNI-Nachricht CONNECT-ACK. Nachdem die Quelle die Connect-Nachricht von der Senke erhalten hat bzw. nachdem sie ihrerseits die Connect-Nachricht quittiert hat, ist sie in der Lage Daten zur Senke zu übertragen. Die Verbindung steht.

In Bild 9.2 ist der Verbindungsabbau schematisch dargestellt. Nachdem alle Daten von der Quelle zur Senke übertragen wurden, kann die Verbindung abgebaut werden. Dieser Verbindungsabbau wird durch die UNI bzw. NNI-Nachricht RELEASE eingeleitet, die

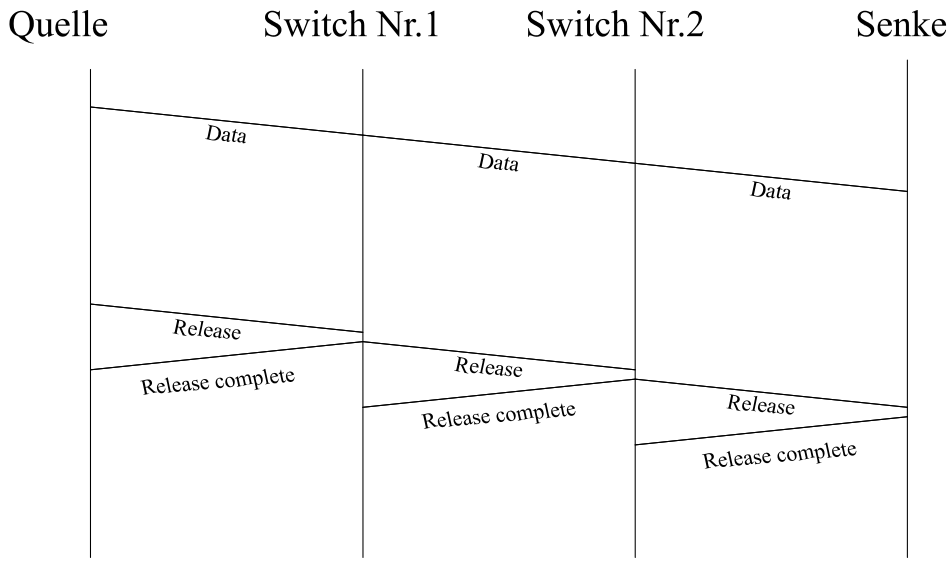


Abbildung 9.2: Schematische Darstellung des Verbindungabbaus.

über die einzelnen Switches zur Senke übertragen wird. Jeder Switch und auch die Senke quittieren die Release-Nachricht mit der UNI bzw. NNI-Nachricht RELEASE COMPLETE. Auf der Seite der Quelle ist die Verbindung mit dem Erhalt der Release-Complete-Nachricht geschlossen. Auf der Seite der Senke ist die Verbindung erst nach dem Empfang der Release-Nachricht beendet bzw. nachdem diese bestätigt wurde.

Das Bild 9.3 zeigt das Zustandsdiagramm eines Teilnehmers zum Aufbau und Abbau einer ATM-Verbindung. Von dem Ruhe-Zustand IDLE aus kann der Zustand des Teilnehmers entweder in den aktiven Zustand CALL INITIATED oder in den passiven Zustand CALL PRESENT übergehen. Aktiver Zustand bedeutet, dass der Teilnehmer durch das Senden der Nachricht SETUP zur Quelle wird. Der passive Zustand wird durch das Eintreffen der Nachricht SETUP zur Senke. Der linke Teil des in Bild 9.3 dargestellten Zustandsdiagramms zeigt die aktiven Zustände und der rechte die passiven Zustände.

Wenn sich der aktive Zustand CALL INITIATED eingestellt hat, kann entweder der Zustand IDLE nach dem Ablauf eines entsprechenden Timers (dies ist nur dann der Fall, wenn der erste Switch nicht antwortet) oder im "normalen" Fall der Zustand OUTGOING CALL PROCEEDING nach Erhalt der Nachricht CALL PROCEEDING erreicht werden. Durch das Eintreffen der Nachricht CONNECT gelangt der Teilnehmer in den Zustand ACTIVE und die Verbindung ist aus der Sicht der Quelle zustande gekommen. Trifft die

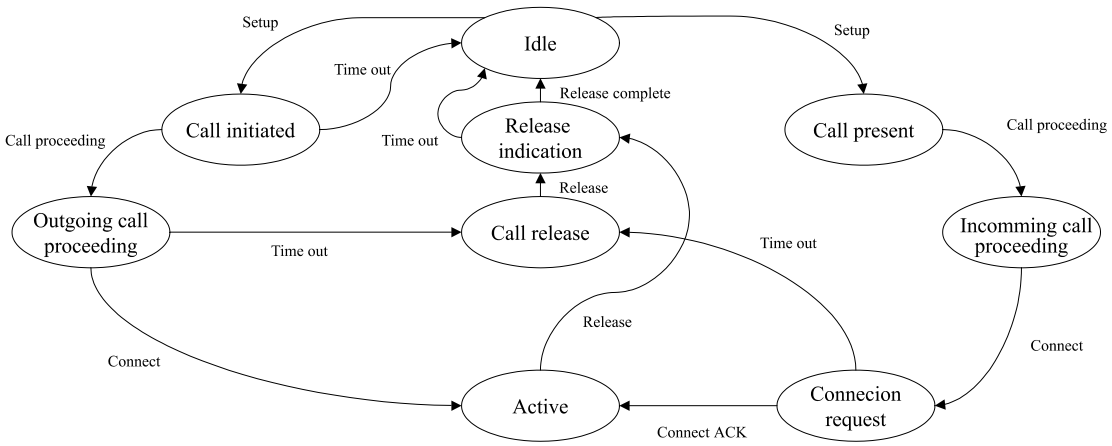


Abbildung 9.3: Zustandsdiagramm eines Teilnehmers für den Verbindungsauf- und -abbau.

Connect-Nachricht jedoch nicht vor Ablauf des entsprechenden Timers ein, wird der Zustand CALL RELEASE eingenommen, der das sofortige Senden der Nachricht RELEASE einleitet. Danach wird der Zustand RELEASE INDICATION automatisch erreicht. Dieser geht entweder nach dem Timerablauf oder nach dem Erhalt der Nachricht RELEASE COMPLETE in den Zustand IDLE zurück.

Wenn sich der passive Zustand CALL PRESENT eingestellt hat, geht die Quelle nach dem senden der Nachricht CALL PROCEEDING in den Zustand INCOMMING CALL PROCEEDING über. Es folgt das Senden der Nachricht CONNECT, wodurch der Zustand CONNECTION REQUEST erreicht wird. Es existieren von dort aus zwei Möglichkeit in andere Zustände zu gelangen. Die erste besteht darin, dass ein Timer abläuft, dann wird der Zustand CALL RELEASE eingenommen und von dort aus werden die oben für den aktiven Teil beschriebenen Zustände durchlaufen, bis sich wieder der Zustand IDLE einstellt hat. Die zweite Möglichkeit besteht darin, dass durch Erhalt der Nachricht CONNECT ACK der Zustand ACTIVE erreicht wird. Die Verbindung ist nun auf der Empfängerseite hergestellt. Der Zustand ACTIVE wird durch das Senden bzw. Empfangen der Nachricht RELEASE verlassen und es stellt sich der Zustand RELEASE INDICATION ein. Dieser geht wie oben beschrieben in den Zustand IDLE über.

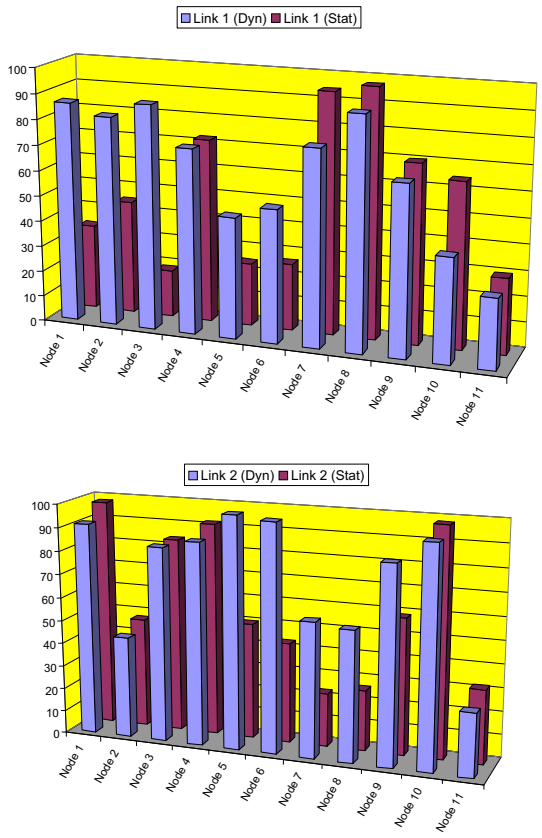


Abbildung 9.4: Relative Linklast auf dem ersten bzw. zweiten Link von 11 Knoten des B-WiN; Simulationsergebnisse mit 440 Quellen und 440 Senken für statisches und dynamisches Routing.

9.2 Ergebnisse

Die Randbedingungen und Parametersetzungen für die Simulationsexperimente entsprechen den in Kapitel 4, Tabelle 4.4 genannten. Aus simulationstechnischen Gründen wurde jedoch die Anzahl der Quellen auf 440 begrenzt.

Die beiden Diagramme des Bildes 9.4 zeigen die mittlere Last am ersten bzw. am zweiten Link eines jeden Knotens im B-WiN. Es zeigt sich, dass keine Systematik zu erkennen ist, ob für einen einzelnen Link das statische oder das dynamische Routing zu einer höheren Auslastung führt.

Interessant sind daher nur die Netzmittelwerte, die in den Bildern 9.5 und 9.6 zu sehen sind. Diese Bilder zeigen, dass Durchsatz und Goodput mit dem Verkehrsangebot steigen, solange man noch hinreichend weit von der Sättigung des Netzes entfernt ist. In dem

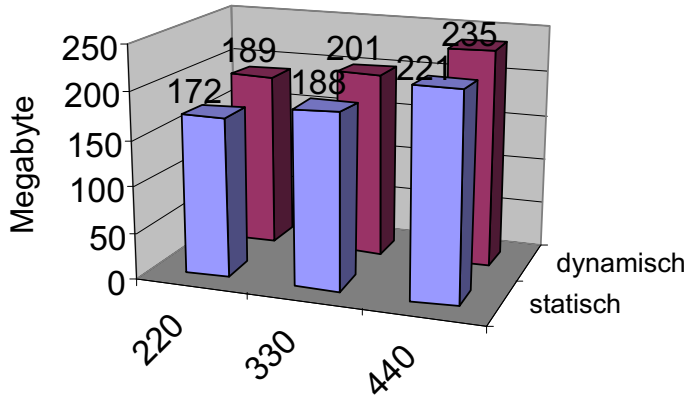


Abbildung 9.5: Vergleich des durch das Netz übertragenen Datenvolumens in Megabyte (Goodput).

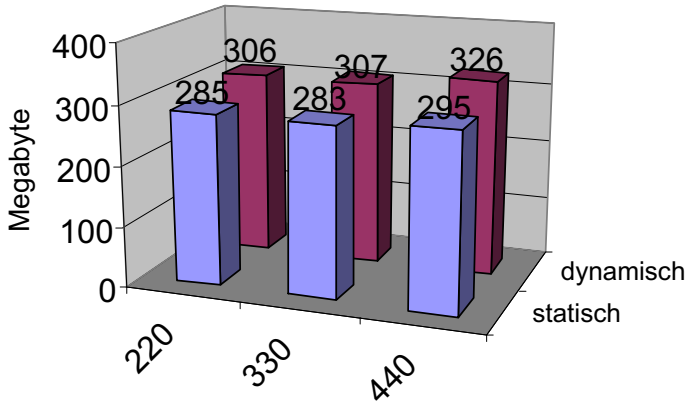


Abbildung 9.6: Vergleich der absoluten Netzlast im Megabyte (Throughput).

betrachteten Bereich führt das dynamische Routing auf Verbesserungen von Durchsatz und Goodput, die im Bereich von 6% liegen. Die Verbesserungen treten signifikant auf, legen aber noch nicht die Empfehlung nahe, ihretwegen eine neue Technik einzuführen.

Neben einer stärkeren Netzauslastung erwartet man beim Einsatz des dynamischen Routings gegenüber dem statischen Routing eine größere Anzahl an Verbindungen, die bei gleichen Voraussetzungen angenommen werden können. Verbindungen, die im statischen Fall abgelehnt werden müssen, weil z.B. nur eine einzige Teilstrecke ausgelastet ist, werden beim dynamischen Routing an dieser überlasteten Teilstrecke vorbeigeleitet. Diese Erwartung wird durch die in Bild 9.7 dargestellten Simulationsergebnisse bestätigt.

Die Anzahl der Verbindungen wächst annähernd linear mit der Anzahl der Quellen im dynamischen Fall und im statischen Fall bleibt die Anzahl der Verbindungen ungefähr konstant, was auch nicht weiter verwunderlich ist, denn wenn im statischen Fall alle sta-

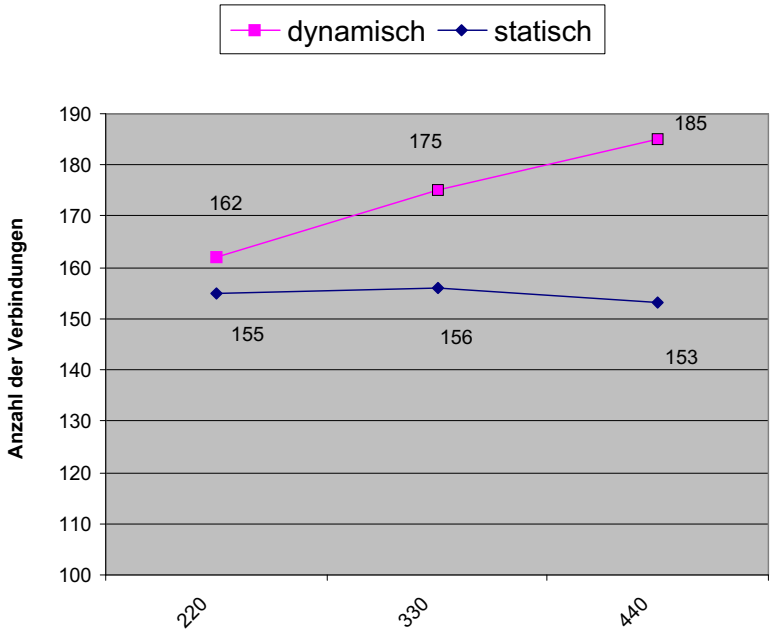


Abbildung 9.7: Vergleich der Anzahl der aufgebauten Verbindungen in Abhängigkeit von der Quellenzahl.

tisch gerouteten Verbindungen belegt sind, kann eine größere Anzahl der Quellen nicht zu einer größeren Zahl an Verbindungen führen. Im dynamischen Fall sieht das anders aus. Dort können erst dann keine Verbindungen mehr angenommen werden, wenn auch alle alternativen Routen voll ausgelastet sind und das sind mehr Verbindungen. Natürlich muss auch beim dynamischen Routing eine Sättigung einsetzen. Die tritt jedoch bei Verwendung von dynamischen Routing wesentlich später ein.

Kapitel 10

Zusammenfassung

In dieser Studie wurde untersucht, auf welche Art und Weise sich der Netzdurchsatz im B-WiN am effektivsten steigern lässt und welche Dienstgüteeigenschaften dabei beobachtet werden können. Dazu wurden die Möglichkeiten, die sich durch die Verwendung der ATM Konzepte ABR, UBR (mit und ohne EPD) sowie mit dynamischem Routing ergeben, ermittelt. Entsprechend den Messungen der Verkehrsstatistiken der letzten Jahre wurde eine realistische Modellierung mit selbstähnlichem Verkehr gewählt.

Das Konzept von Pilotquellen, an denen die Messungen durchgeführt werden, in Verbindung mit aggregiertem Hintergrundverkehr zum Auffüllen der Leitungen wurde als wenig tragfähig erkannt, da zum Einen nicht ersichtlich ist, wie die Wegelenkung des Hintergrundverkehr gestaltet werden soll und zum Anderen die Reaktivität dieses großen Verkehrsanteils nicht einfach vernachlässigt werden kann. Stattdessen wurde eine realistischere Modellierung gewählt, wo jede einzelne Quelle einen kompletten TCP/IP bzw. TCP/IP/ATM Protokoll Stapels besitzt und daher die Dynamik der realen Quellen vollständig nachbildet.

Es wurden Simulationen an einem Netzmodell mit ATM und an einem reinen IP Netzmodell mit jeweils 1000 aktiven Quellen analysiert. Nach dem Wissen der Autoren ist es das erste Mal, dass so ein komplexes Netz wie das B-WiN (11 Knoten, 18 bidirektionale Links mit einer Gesamtkapazität von 3.9 Gbit/s) mit einer so hohen Quellenzahl für Forschungszwecke simuliert wird.

Die Problematik der realistischen Lasterzeugung bei reaktiven TCP Quellen wurde in dieser Studie erkannt und in Kapitel 6 diskutiert. Der Vergleich von zwei Ansätzen, die beide keine perfekten Ergebnisse erzielen können belegt die Komplexität des Optimierungsproblems.

Die erzielten Ergebnisse lassen sich wie folgt zusammenfassen:

- Es zeigte sich, dass der Einfluss der Selbstähnlichkeit (bzw. des power-tail Index α) bei hoher Last auf den Pufferbedarf und das Verzögerungsverhalten im Netz vernachlässigt werden kann.
- Die erhofften Verbesserungen der Netzleistung durch Einführung von EPD für den UBR Dienst haben sich eingestellt. Es zeigten sich Verbesserungen sowohl bei dem Goodput als auch im Verzögerungsverhalten. Allerdings rechtfertigen die im einstelligen Prozentbereich beobachteten Verbesserungen keine größeren Investitionen.
- Die Untersuchungen zum ABR zeigen, dass hier nur marginale Verbesserungen - wenn überhaupt - zu erzielen sind. Letztlich arbeiten hier zwei Regelungen, die je nach Parametersetzungen und Totzeiten sich ergänzen oder gegeneinander arbeiten. Die maximal erzielten Durchsätze waren in diesen Experimenten jedoch nur bei 60 %.
- Auch das dynamische Routing kann nur in Einzelfällen einen marginalen Gewinn im Goodput erzielen und wird daher nicht als eine verfolgenswerte Option betrachtet.

Zusammenfassend erscheint es sinnvoll, eine UBR Dienst mit EPD anzubieten, um die Netzauslastung zu verbessern. Die anderen Optionen (ABR, dynamisches Routing) erscheinen wenig geeignet, um die Netzauslastung zu verbessern.

Anhang A

Selbstähnlichkeit und Hurst Parameter

A.1 Der Zufallsprozess

- Kontinuierliche Zufallsvariable:

$$Y = Y(t), \qquad t \geq 0 \tag{A.1}$$

- Diskrete Zufallsvariable:

$$X_i \qquad i = 1, 2, 3, \ldots \tag{A.2}$$

  Beispiel: Zwischenankunftszeiten, Zählprozeß

- Verteilungsfunktion:

$$F(x) = P\{X \leq x\} \tag{A.3}$$

- Komplementäre Verteilungsfunktion (engl.: “Reliability function”):

$$R(x) = 1 - F(x) = P\{X > x\} \tag{A.4}$$

- Verteilungsdichtefunktion:

$$f(x) = \frac{dF(x)}{dx}$$

- Aggregation von Diskreten Zufallsvariablen (Mittelung über nicht überlappende Bereiche der Größe m):

$$X_i^{(m)} = \frac{1}{m} \sum_{j=(i-1)m+1}^{im} X_j \tag{A.5}$$

A.2 Selbstähnlichkeit

Es existieren in der Literatur verschiedene, nicht äquivalente Definitionen der Selbstähnlichkeit. Im folgenden werden die unterschiedlichen Definitionen angegeben. Die Notation $Z \stackrel{d}{=} X$ bedeutet hier, dass die beiden Zufallsvariablen Z und X die gleiche Verteilungsfunktion (vom engl. “distribution”) besitzen.

- Selbstähnlichkeit der Zeitkontinuierlichen Zufallsvariablen

$$Y(t) \stackrel{d}{=} a^{-H} Y(at), \quad \forall t \geq 0, \quad \forall a > 0, \quad 0 < H < 1 \tag{A.6}$$

- Selbstähnlichkeit der aggregierten Zeitdiskreten Zufallsvariablen

$$X \stackrel{d}{=} m^{1-H} X^{(m)} \tag{A.7}$$

A.3 Selbstähnlichkeit Zweiter Ordnung

Dies ist die wichtigste Definition der Selbstähnlichkeit in Bezug auf das Leistungsverhalten von Netzen, da sich hier die Korrelation und Varianz sehr stark auswirken.

- Selbstähnlichkeit zweiter Ordnung: Die Autokorrelation der Zufallsvariablen ist der Autokorrelation der aggregierten Zufallsvariablen gleich.

$$r^{(m)}(k) = r(k) \quad \text{fr} \quad m, k \to \infty \tag{A.8}$$

Weitere Eigenschaften von Zufallsvariablen, die obige Gleichung (A.8) erfüllen:

- "Long Range Dependence" (LRD):
  - Langsam abfallende Autokorrelationsfunktion:

$$r(k) \sim i^{-\beta} \quad \text{mit} \quad 0 < \beta < 1 \tag{A.9}$$

  - Autokorrelationsfunktion ist nicht summierbar:

$$\sum_{i=-\infty}^{+\infty} r(k) = \infty \tag{A.10}$$

- Langsam mit steigendem Aggregationslevel abfallende Varianz:

$$Var\{X_k^{(m)}\} \sim m^{-\beta} \quad mit \quad 0 < \beta < 1 \tag{A.11}$$

A.4 Heavy- / Power- / Long-Tailedness

Für eine "heavy- / power- /long-tail" verteilte Zufallsvariable gilt:

$$R(x) \sim cx^{-\alpha} \quad \text{fr}\, x \to \infty, 0 < \alpha < 2\, c > 0 \tag{A.12}$$

- $1 < \alpha < 2$: der Prozess hat einen endlichen Mittelwert und eine *unendlich große Varianz*
- $0 < \alpha < 1$: der Prozess hat einen *unendlich großen Mittelwert* und eine unendlich große Varianz

Verteilungen dieser Art werden benutzt, um durch die Überlagerung mehrerer Quellen solcher Verteilung Selbstähnlichen Verkehr zu erzeugen (siehe Kapitel 3).

A.5 Umrechnung von Hurst Parameter H / α / β

$$H = \quad (3-\alpha)/2 \quad = 1-\beta/2 \tag{A.13}$$

$$\alpha = \quad 3-2H \quad = 1+\beta \tag{A.14}$$

$$\beta = \quad 2(1-H) \quad = \alpha-1 \tag{A.15}$$

A.6 Benutzte Hurst Parameter Schätzer

A.6.1 Variance-Time Plot

In diesem Verfahren wird die in Gl. (A.11) beschriebene Eigenschaft benutzt um den Hurst Parameter einer Zufallsvariablen mit Selbstähnlichkeit zweiter Ordnung zu ermitteln. In einem doppelt logarithmischen Plot wir die Varianz über dem Aggregationslevel m aufgetragen (siehe Bild A.1). Ist die Eigenschaft (A.11) erfüllt, so beschreibt der Graph für große Aggregationslevel m eine abfallende Gerade mit der Steigung $-\beta$. In dieser Version ist es möglich, nach einer Voransicht einen möglichst großen Bereich von Hand auszuwählen, wo die Steigung näherungsweise konstant ist. Für kleine und für sehr große m werden in der Praxis immer Abweichungen von der idealen Linie auftreten. Bei kleinen m kann eine Kurzeitkorrelation (engl. “Short Range Dependence”, SRD) die Messwerte verfälschen, bei großen Werten ist die Genauigkeit aufgrund der limitierten Anzahl an Messwerten begrenzt. Daher ist die Auswahl des Bereichs mit näherungsweise konstanter Steigung sinnvoll, um die Genauigkeit der Schätzung zu erhöhen.

A.6.2 R/S-Plot

Die R/S Statistik, auch “ReScaled adjusted range“ genannt, wird wie folgt hergeleitet:

- partielle Summe:

$$Y(n) = \sum_{i=1}^{n} X_i \tag{A.16}$$

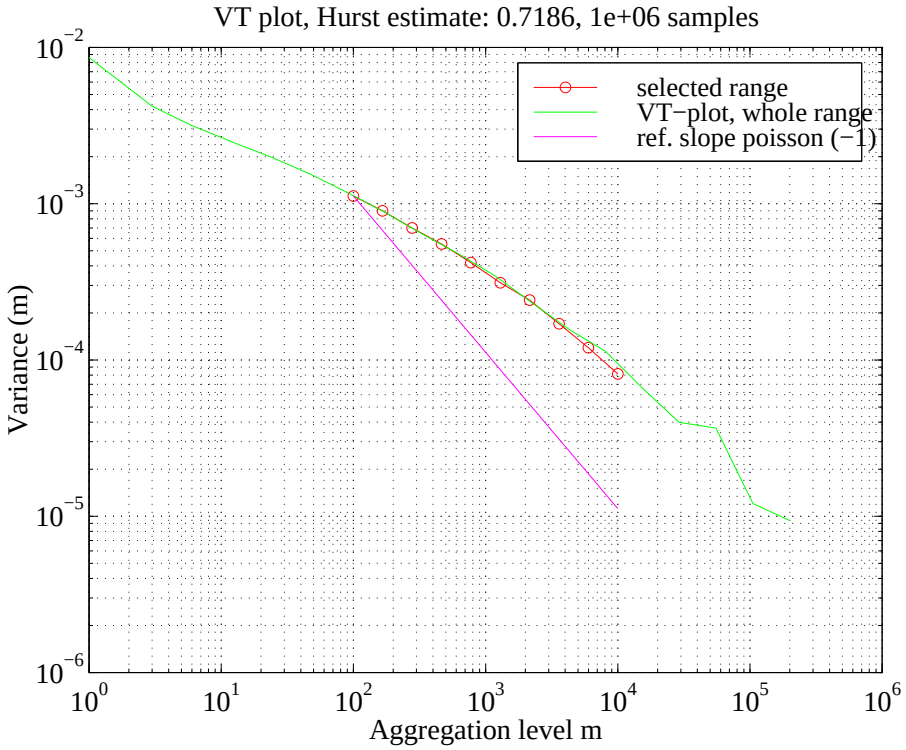


Abbildung A.1: Beispiel eines Variance-Time Plots.

- Varianz der Stichprobe:

$$\sigma^2(n) = \frac{1}{n}\left(\sum_{i=1}^{n} X_i^2 - \frac{1}{n}Y(n)^2\right) \tag{A.17}$$

- R/S:

$$R/S(n) = \frac{1}{S(n)}\left[\max_{o\le t\le n}\left(Y(t) - \frac{t}{n}Y(n)\right) - \min_{o\le t\le n}\left(Y(t) - \frac{t}{n}Y(n)\right)\right] \tag{A.18}$$

- für “fractional Gaussian noise” oder “fractional ARIMA” gilt:

$$E[R/S(n) \sim n^H \tag{A.19}$$

Auch bei diesem Plot ist es möglich, nach einer Voransicht einen möglichst großen Bereich von Hand auszuwählen, wo die Steigung näherungsweise konstant ist.

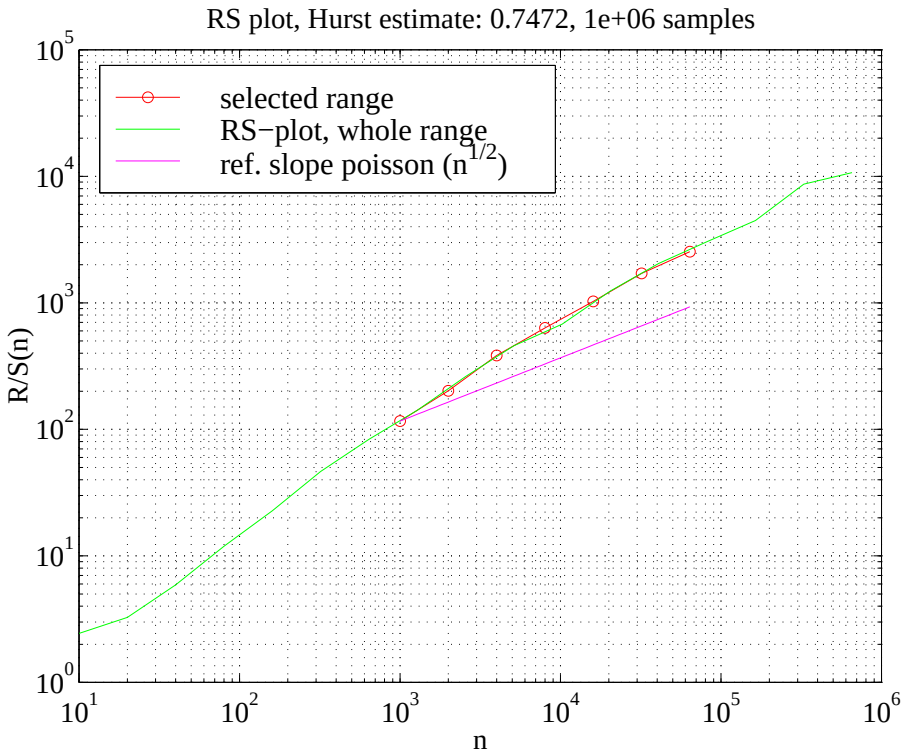


Abbildung A.2: Beispiel eines R/S-Plots.

Anhang B

Überlastabwehr in TCP

B.1 Slow Start und Congestion Avoidance

Dies sind die beiden Basisalgorithmen im TCP, mit denen die Größe des Congestion Window reguliert wird. Da sie schon immer miteinander kombiniert implementiert wurden, werden sie in der Literatur auch meistens zusammenhängend beschrieben. Sie haben allerdings ganz unterschiedliche Aufgaben.

In den BSD-Versionen 4.1 und 4.2 konnte eine neue Verbindung sofort so viele Daten abschicken, wie es das Empfangsfenster der Zielstation zuließ. Dies führte aber oft gleich zu mehrfachen Verlusten, wenn die Segmente unterwegs auf eine langsamere Teilstrecke trafen. Im Zuge der ständig wachsenden Netzlast im Internet war das bald nicht mehr akzeptabel, weshalb im TCP Tahoe die beiden hier beschriebenen Algorithmen implementiert wurden. Sie sind erstmals in [JK88], einem der Meilensteine in der Entwicklung des TCP, veröffentlicht worden. Zur Ergänzung sei auch noch auf [Jac88] verwiesen.

B.1.1 Slow Start

Der Slow Start Algorithmus löst dabei das eben beschriebene Problem. Er wird immer dann benutzt, wenn auf einer Verbindung noch nicht oder schon lange nicht mehr gesendet wurde. In beiden Fällen kommen keine Bestätigungen an, sodass der Sendetakt gefunden werden muss, ohne dabei das Netz zu überlasten.

Der Sender setzt sein Congestion Window dazu auf einen Anfangswert, der gewöhnlich bei einem Segment liegt und vergrößert es mit jeder ankommenden neuen Bestätigung um jeweils ein weiteres Segment [APS]. Das Fenster öffnet sich dabei exponentiell. Die Bestätigung für das erste Segment kommt nach einer Umlaufzeit an, worauf zwei Segmente gesendet werden können. Nach einer weiteren Umlaufzeit kommen zwei Bestätigungen an, für die dann insgesamt vier Segmente gesendet werden können und so weiter. Das Congestion Window wird also innerhalb jeder Umlaufzeit verdoppelt.

B.1.2 Congestion Avoidance

Der Congestion Avoidance Algorithmus sollte dagegen immer dann benutzt werden, wenn gerade Segmente unterwegs sind und ein Verlust auftritt. Der Algorithmus besteht aus zwei Teilen. Zunächst muss bei einem Verlust die Senderate und damit das Congestion Window um einen bestimmten Betrag verringert werden, da vor der letzten Vergrößerung ja noch keine Verluste aufgetreten sind.

Der Standard schreibt hierfür eine Halbierung des Congestion Window vor, was bei mehreren Verlusten hintereinander eine exponentielle Verkleinerung bewirkt. In [JK88] wird darauf hingewiesen, dass dies in Netzen mit einer endlichen Anzahl an Verbindungen bei beliebigem Verkehrsaufkommen eine Mindestanforderung für Stabilität darstellt.

Zur Veranschaulichung sei angenommen, dass das Congestion Window vor dem Verlust durch den Slow Start Algorithmus verdoppelt worden ist. Dies stellt den ungünstigsten aller mögliche Fälle dar und würde durch diese Maßnahme nun rückgängig gemacht werden.

Da sich bei der Aktivierung des Congestion Avoidance Algorithmus mehr Segmente im Netz befinden können als das nun halbierte Fenster eigentlich zulässt, muss der Sender gegebenenfalls warten, bis genug Segmente bestätigt wurden und er weitersenden darf.

Anschließend wird das Congestion Window dann nicht mehr exponentiell, sondern nur noch linear vergrößert, um sich der maximal möglichen Fenstergröße langsamer zu nähern. Dies wird dadurch erreicht, dass pro Sendefenster und damit normalerweise pro Umlaufzeit ein Segment hinzuaddiert wird.

B.1.3 Slow Start Threshold

Werden beide Algorithmen zusammen implementiert, so muss sich eindeutig bestimmen lassen, welcher gerade benutzt wird. Dazu wird beim Auftreten eines Verlustes der halbierte Wert des Congestion Window in einer weiteren Variablen abgelegt, die nach unten auf zwei Segmente begrenzt wird. Unterhalb dieser Grenze, der so genannten “Slow Start Threshold”, macht die Verbindung einen Slow Start, ansonsten macht sie Congestion Avoidance [APS].

Es hängt nun von der Art und Weise der Verlusterkennung ab, ob das Congestion Window im Verlustfall halbiert oder auf den Startwert gesetzt wird. Das TCP benutzt dazu im wesentlichen zwei Methoden, die in den beiden folgenden Unterabschnitten behandelt werden. Im Anschluss daran wird der Vollständigkeit wegen noch eine weitere Methode beschrieben, die jedoch normalerweise nicht eingesetzt wird.

B.2 Fast Retransmit und Fast Recovery

Wenn Segmente außerhalb der Reihenfolge ankommen, dann sendet das TCP immer Bestätigungen für das letzte lückenlos erhaltene Segment. So eine doppelte Bestätigung kann also zunächst zwei Ursachen haben:

- Ein Segment wurde unterwegs von einem anderen überholt.
- Ein Segment ist unterwegs verlorengegangen.

Je mehr doppelte Bestätigungen ankommen, desto unwahrscheinlicher wird die erste Möglichkeit und desto eher kann ein Verlust angenommen werden. Das TCP macht letzteres nach drei unmittelbar hintereinander empfangenen, also nach insgesamt vier gleichen lückenlosen Bestätigungen. Dies ist ein Erfahrungswert, der in Versuchen im Mittel den größten effektiven Durchsatz erzielte und deshalb standardisiert wurde [Pax97]. Da er stark von der jeweiligen Verkehrscharakteristik abhängt, kann er nicht eindeutig analytisch bestimmt werden. In einem ATM-Netzwerk mit statischem Routing könnte natürlich

prinzipiell bereits nach einer einzigen doppelten Bestätigung ein Verlust angenommen werden, da dort dann keine Umordnung möglich ist.

Der Fast Retransmit Algorithmus, der erstmals in [JK88] vorgestellt wurde, wiederholt nun das erste unbestätigte Segment und setzt wie bereits beschrieben die Slow Start Threshold auf den halben Wert des Congestion Window.

Das TCP Tahoe benutzt dann daraufhin den Slow Start Algorithmus, das heißt es setzt das Congestion Window auf ein Segment, löscht alle Timer und startet den für das wiederholte Segment neu. Diese Vorgehensweise macht aber nur bedingt Sinn, da noch einige Segmente unterwegs sein können und in diesem Fall dann eher Congestion Avoidance angebracht ist.

Das TCP Reno benutzt daher stattdessen den Fast Recovery Algorithmus, der in [Jac90] vorgeschlagen wurde und bereits während der Wiederholungsphase versucht, die im Netz befindliche Datenmenge der entsprechenden Verbindung zu halbieren. Dabei wird nur der Timer für das wiederholte Segment gelöscht und neu gestartet, alle anderen laufen weiter.

Solange das wiederholte Segment noch nicht bestätigt worden ist, bleibt die Menge der unbestätigten Daten im Netz in jedem Fall konstant. Für jede doppelte Bestätigung muss aber ein Segment das Netz verlassen haben.

Der Fast Recovery Algorithmus schätzt nun die Datenmenge im Netz, indem er zu dem zuvor halbierten Congestion Window für jede doppelte Bestätigung ein Segment addiert. Sobald nun das Congestion Window wieder seinen alten Wert erreicht, hat also die Hälfte der zuvor im Netz befindlichen Daten dieses verlassen und der angestrebte Zustand ist erreicht. Für jede weitere doppelte Bestätigung, die jetzt noch ankommt, kann ein neues Segment gesendet werden, sodass der erreichte Zustand erhalten bleibt.

Dieser Vorgang basiert aber deshalb lediglich auf einer Schätzung, weil das TCP dabei wie bereits an früherer Stelle erwähnt Fehler macht. Es weiß nämlich nicht, wieviele Segmente verlorengegangen sind. Sind es mehrere, so ist die halbe Netzlast bereits früher erreicht als angenommen, da nur eines durch die Wiederholung ersetzt wurde. Dieses Problem wird in [Pit96] aufgegriffen.

Der Fast Recovery Algorithmus ermöglicht es zwar, bereits neue Segmente zu senden, während das wiederholte Segment noch gar nicht bestätigt ist. Bei mehrfachen Verlusten und mehreren Fast Retransmit hintereinander kann aber auch immer nur ein Segment pro Umlaufzeit wiederholt werden. Folgt stattdessen ein Slow Start, so existiert diese Einschränkung nicht, dafür werden aber unter Umständen Segmente wiederholt, die sich bereits im Empfangspuffer der Gegenseite befinden.

Die gemeinsame Implementierung von Fast Retransmit und Fast Recovery sieht also folgendermaßen aus:

1. Bei der dritten doppelten Bestätigung wird das erste unbestätigte Segment wiederholt und

$$SST \leftarrow \frac{CW}{2}$$

$$CW \leftarrow SST + 3$$

2. Bei jeder weiteren doppelten Bestätigung wird

$$CW \leftarrow CW + 1$$

3. Wenn möglich werden neue Segmente gesendet

4. Bei der Bestätigung des wiederholten Segmentes wird

$$CW \leftarrow SST$$

Wie später noch gezeigt wird, ist dieser kombinierte Algorithmus bei mehrfachen Verlusten innerhalb einer Umlaufzeit problematisch. Geht nur wenig verloren, so funktioniert er dagegen ausgezeichnet. In Bild B.1 ist beispielhaft der Verlauf von Congestion Window und Slow Start Threshold für eine Übertragungszeit von einer Minute dargestellt[1].

[1] Die Länge der Warteschlangen beträgt zehn Segmente, die Begrenzung des Fast Retransmit Algorithmus ist abgeschaltet.

Die oberen beiden Bilder zeigen das Verhalten des TCP Tahoe, wo nach jedem Fast Retransmit ein Slow Start folgt. Die unteren beiden Bilder zeigen das Verhalten des TCP Reno, wo dagegen Fast Recovery folgt, was bei dieser Zeitauflösung jeweils an den Spitzen im Congestion Window erkennbar ist. Aufgrund von mehrfachen Verlusten innerhalb einer Umlaufzeit ist das Protokoll hier aber immer wieder auf Timeouts angewiesen. Der effektive Durchsatz, der in Kapitel 4 definiert ist, beträgt in diesen beiden Beispielen 38 % für das TCP Tahoe und 63 % für das TCP Reno. Es sei jedoch angemerkt, dass diese Werte extrem vom jeweiligen Simulationsszenario abhängen. Einige Angaben dazu sind auch in [AD98] enthalten.

Bei der Verwendung des Fast Retransmit Algorithmus taucht außerdem grundsätzlich dann ein Problem auf, wenn der Fast Recovery Algorithmus nicht implementiert ist oder ein Timeout ausgelöst wird. In diesen beiden Fällen können infolge eines Slow Start Segmente wiederholt werden, die gar nicht verlorengegangen sind und sich bereits im Puffer des Empfängers befinden.

Wird dabei mit einem wiederholten Segment eine Lücke geschlossen, so wird dieses zusammen mit den im Puffer befindlichen bestätigt. Die nachfolgenden wiederholten Segmente, deren Originale bereits angekommen waren, erzeugen dann doppelte Bestätigungen. Kommen davon wiederum drei am Sender an, so lösen diese unnötigerweise einen Fast Retransmit aus. Dies hat vor allem dann fatale Folgen, wenn danach wieder ein Slow Start folgt.

In [Flo95] wurde dieses Problem beschrieben und eine Lösung in Form einer Begrenzung des Fast Retransmit Algorithmus vorgeschlagen. Sie wurde von der IETF im April 1999 in [FH] mit dem Status "'Experimental"' dokumentiert. Dass sie nicht standardisiert wurde ist insofern erstaunlich, da sie die Effizienz des Fast Retransmit Algorithmus erheblich verbessern kann und deshalb auch in einigen kommerziellen Implementierungen eingesetzt wird.

Zur Lösung des Problems wird gemäß [FH] zunächst eine weitere Variable eingeführt. Wenn nun ein Timeout ausgelöst wird, dann wird die höchste zu diesem Zeitpunkt gesendete Sequenznummer gespeichert. Dies geschieht ebenfalls dann, wenn ein Fast Retransmit ausgelöst wird und kein Fast Recovery implementiert ist. Die Bedingung von drei

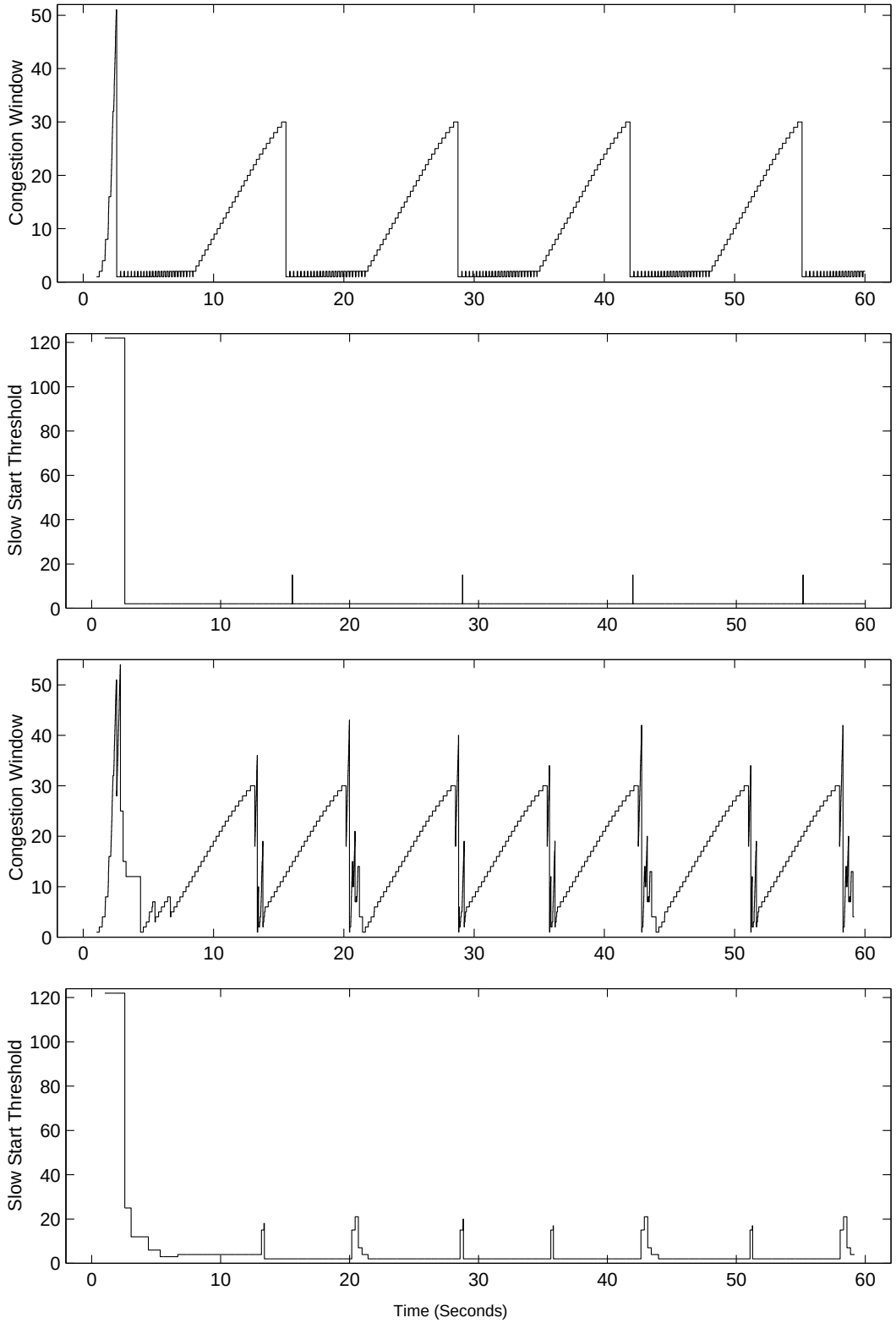


Abbildung B.1: TCP Tahoe und TCP Reno ohne Begrenzung des Fast Retransmit Algorithmus.

doppelten Bestätigungen für einen Fast Retransmit wird nun dahingehend erweitert, dass auch die gespeicherte Sequenznummer bestätigt sein muss.

In Bild B.2 ist wieder beispielhaft der Verlauf von Congestion Window und Slow Start Threshold für eine Übertragungszeit von einer Minute dargestellt, jetzt aber mit der eben beschriebenen Begrenzung des Fast Retransmit Algorithmus[2]. Man erkennt, dass sich diese im TCP Tahoe stark bemerkbar macht, da dort nach einem Fast Retransmit immer ein Slow Start folgt. Im TCP Reno ändert sich dagegen aufgrund des Fast Recovery Algorithmus fast nichts. Der effektive Durchsatz beträgt in diesen beiden Beispielen nun 75 % für das TCP Tahoe und wieder 63 % für das TCP Reno.

In Bild B.3 sind deshalb noch einmal die Auswirkungen der Begrenzung auf das TCP Tahoe in einem anderen Simulationsszenario dargestellt[3]. Es tritt hier nur ein einzelner Verlust auf, der einen Fast Retransmit mit anschließendem Slow Start zur Folge hat. Während die Begrenzung in dem oberen Bild ausgeschaltet ist, was eigentlich dem Standard entspricht, ist sie in dem unteren Bild aktiviert. Der effektive Durchsatz beträgt oben 16 % und unten 45 %.

In beiden Bildern erkennt man darüberhinaus, dass zunächst immer erst mehrere Segmente gesendet werden und dann eine kurze Pause eintritt. Dieser Gruppeneffekt entsteht dadurch, dass das Sendefenster noch wesentlich kleiner als das Bandbreite-Verzögerungs-Produkt der Verbindung ist.

[2] Die Länge der Warteschlangen beträgt zehn Segmente.
[3] In dem oberen Bild ist die Begrenzung des Fast Retransmit Algorithmus abgeschaltet.

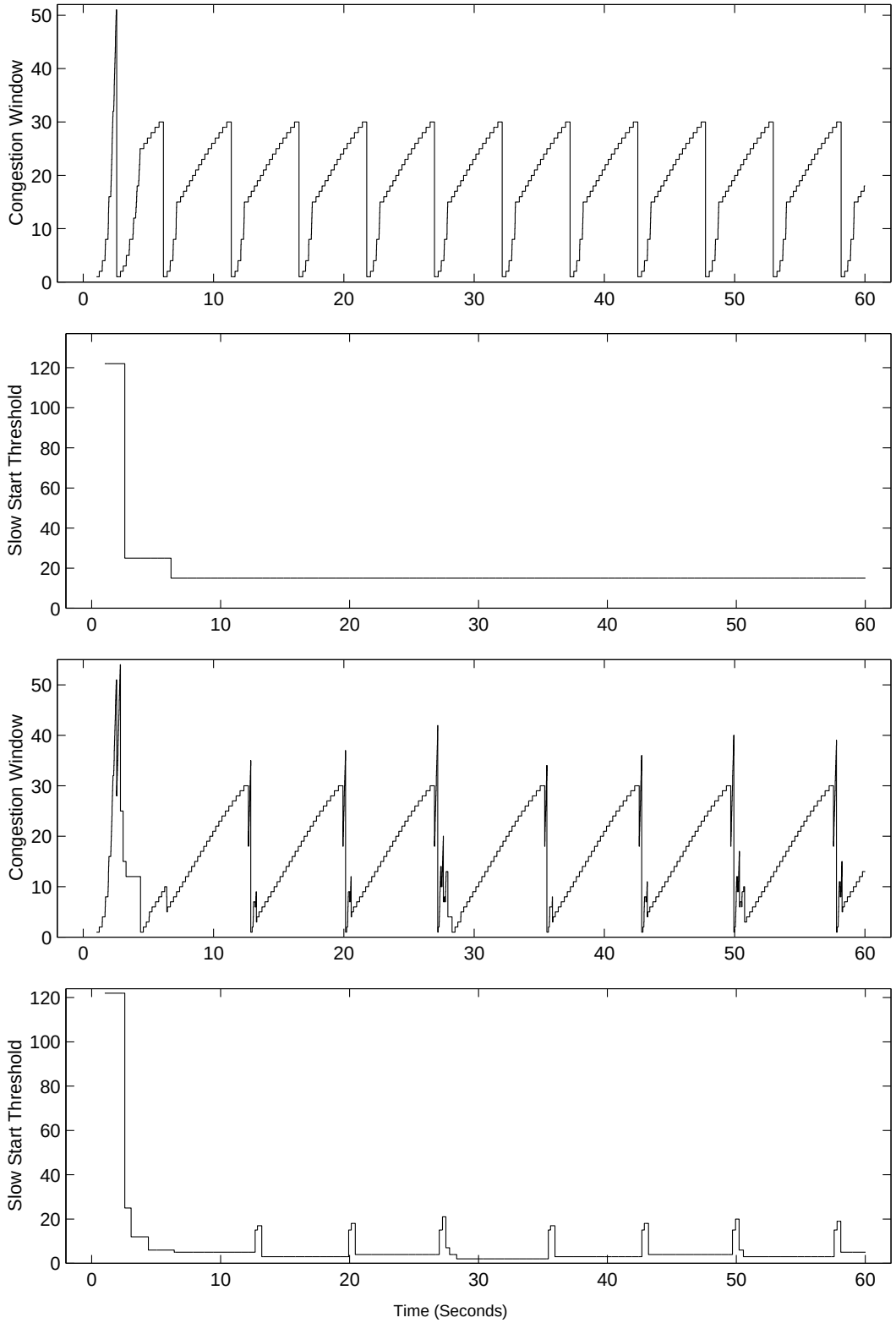


Abbildung B.2: TCP Tahoe und TCP Reno mit Begrenzung des Fast Retransmit Algorithmus.

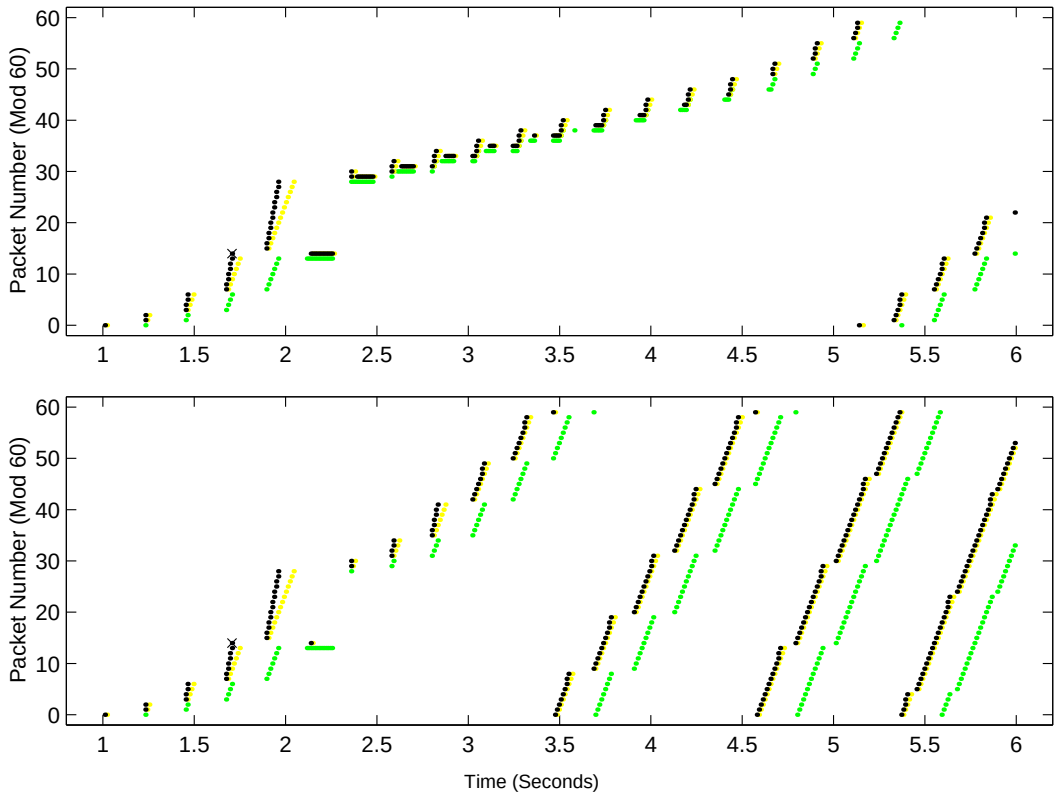


Abbildung B.3: TCP Tahoe ohne/mit Begrenzung des Fast Retransmit Algorithmus.

Anhang C

Die “Free Bit Rate” (FBR) Dienstkategorie

Die ATM-Technologie wurde entwickelt, um durch das Zusammenspiel von schnellen physikalischen Datenleitungen mit einfachen aber schnellen Vermittlungen effiziente Kommunikationsnetze aufbauen zu können. Um dabei die Konzepte einfach zu halten und gleichzeitig die Kosten für ihre Implementierung zu senken, wird in ihnen überhaupt keine Flusskontrolle durchgeführt. Geringe Datenverlustwahrscheinlichkeiten werden erreicht, indem Wissen über die statistischen Eigenschaften neu aufzubauender Verbindungen genutzt wird, diese im Rahmen des Signalisierungsprotokolls anzunehmen oder abzulehnen. Die vorhandenen Übertragungkapazitäten werden dabei den einzelnen Verbindungen a priori, also beim Verbindungsaufbau zugewiesen. Die hierfür verwendeten Verfahren sind in der Regel konservativ, das heißt es ergeben sich zunächst ungenutzte Netzkapazitäten. Diese Kapazitäten trotzdem nutzbar zu machen und während des Bestehens einer Verbindung dieser a posteriori eine der Lastsituation im Netz angepasst Bandbreite zuzuweisen, waren die beiden Hauptforderungen für die Einführung einer neuen Dienstkategorie. Daraus entstanden ist zum einen die bereits beschriebene ABR-Dienstkategorie und als Weiterentwicklung, bzw. Modifikation daraus die FBR-Dienstkategorie, deren Vorteile und Unterschiede zu ABR im folgenden beschrieben werden.

Der ABR-Dienst füllt die beim herkömmlichen, statistischen Multiplexen notwendigen

Leerzellen auf den einzelnen ATM-Links mit ABR-Zellen auf, die zuvor in dem jeweils sendenden Knoten in großen Puffern zwischengespeichert wurden. Neben den durch das statistische Multiplexen entstehenden Leerzellen kann auch durch “normale” ATM-Verbindungen gemäß CAC ungenutzte Bandbreite unter den ABR-Verbindungen aufgeteilt werden. So kann die Linkauslastung im Idealfall bis zu 100% betragen, von denen aber wiederum zwischen 3 und 7% der durch ABR genutzten Kapazität für die Regelung der Quellen benötigt werden. Die Regelung sorgt dabei dafür, dass die ABR-Puffer weder über- noch leerlaufen.

Im Gegensatz dazu nutzt der FBR-Dienst die für das statistische Multiplexen notwendigen Leerzellen nicht, wohl aber die gemäß CAC noch nicht genutzte Bandbreite. Dazu werden FBR-Verbindungen wie CBR-Verbindungen gehandhabt, denen aber, der momentanen Last im Netz und dem momentanen Bedarf der Quelle entsprechend, laufend variierende Bandbreiten zugewiesen werden. Beim ausschließlichen Multiplexen von CBR-Verbindungen kann jedoch die verfügbare Linkkapazität ebenfalls zu fast 100% ausgenutzt werden, und dies mit den “normalen” ATM-Puffern. Auch hier werden allerdings wiederum einige Prozent der verfügbaren Bandbreite für die Regelung benötigt.

Sowohl ABR wie auch FBR eignen sich besonders gut als so genannter “best effort service” für die Übertragung von Daten im Internet, wobei ABR bei paralleler Nutzung von VBR vorhandene Linkkapazitäten im Prinzip besser ausnutzen kann, da sämtliche und nicht nur die gemäß CAC freien Leerzellen nutzbar sind. Wenn sich allerdings nur FBR und CBR ein Link teilen ist die für das statistische Multiplexen notwendige ungenutzte Bandbreite gering.

Im weiteren wird auf drei wesentliche Aspekte der FBR-Dienstkategorie eingegangen: Zunächst werden in Abschnitt C.1 die wichtigsten der in den einzelnen Netzknoten für die Regelung zur Verfügung stehenden Zustandsvariablen bzw. Messwerte vorgestellt und ihre Relevanz bezüglich ABR und FBR erläutert. In Abschnitt C.2 wird dem gegenseitige Einfluss von Rufauf- und -abbau auf der einen und den Regelzeiten der FBR-Verbindungen auf der anderen Seite nachgegangen.

C.1 Datenbasis für die Regelung von FBR-Quellen

Wie beim herkömmlichen ATM – also nicht bei ABR – basiert bei FBR die Vergabe von Bandbreite allein auf Informationen, die a priori verfügbar sind. Es wird also sozusagen der “traffic contract” der (CAC basierten) Lastsituation im Netz und dem momentanen Bandbreitebedarf der Quelle dynamisch angepasst. Die von ABR benötigten, gemessenen Datenraten der ABR-Quellen, Pufferbelegungen und gemessenen für ABR verfügbaren Bandbreiten werden nicht verwendet.

Statt dieser Messwerte gehen die ohnehin vorhandenen Informationen der Rufannahme einschließlich der “minimum cell rate” (MCR), sowie die momentan von den FBR-Quellen zusätzlich gewünschte und ggf. bereits von anderen Knoten im Netz begrenzte Bandbreite in die Zuteilung der für FBR verfügbaren Bandbreite einzelner Verbindungen in die den Quellen zugewiesenen Bandbreiten ein. Natürlich werden hierbei auch die jeweilige “peak cell rate” (PCR) und eine mögliche Begrenzung der maximal zu empfangeden Bandbreite seitens der Datensenke berücksichtigt.

Ein neuer “traffic contract” mit verringerter Datenrate wird dabei erst gültig, nachdem er den Weg vom begrenzenden Knoten bis zur Quelle zurückgelegt hat. Noch länger dauert die Erhöhung der Datenrate, da dieser alle Knoten – sowohl “upstream” als auch “downstream” – zustimmen müssen. Es kann daher sinnvoll sein, zusätzlich Informationen über die Umlaufzeiten von Regelimpulsen zwischen dem betrachteten Netzknoten und der Quelle sowie zwischen Quelle und Senke insgesamt zu ermitteln. Aufgrund dieser können dann kurze und damit schnell reagierende Verbindungen beim Ausgleichen von plötzlichen Lastwechseln bevorzugt und damit die Netzauslastung insgesamt gesteigert werden.

In die Berechnung der Umlaufzeiten gehen dabei die Verzögerungen in Leitungen und Netzknoten sowie der Abstand der RM-Zellen als auch der Takt mit dem die Regelung erfolgt. Die Verzögerungen in den Netzknoten setzen sich dabei aus der eigentlichen Bearbeitungszeit der Zellen und den maximalen durch die Pufferung verursachten Wartezeiten zusammen, die zunächst zwar als beliebig aber dennoch fest angenommen werden. Der Abstand der RM-Zellen ist bei der ABR-Dienstkategorie durch insgesamt drei Re-

geln mit der Momentanen Datenrate der Quelle und einem zusätzlichen Zeitgeber verbunden. Sie bilden derzeit auch die Grundlage für die Erzeugung von RM-Zellen von FBR-Verbindungen, können hierfür aber vermutlich noch weiter optimiert werden. Der Berechnungstakt hängt wesentlich von der Leistungsfähigkeit der verwendeten Prozessoren, aber auch von der Frequenz ab, mit der RM-Zellen generiert bzw. verarbeitet werden.

Aus den oben beschriebenen Größen berechnet schließlich die Rufannahme bzw. die Regelung die für FBR verfügbare Bandbreite und teilt diese möglichst effizient und fair unter den FBR-Verbindungen auf. Die so erzeugten Regelimpulse ermöglichen es schließlich zusammen mit der Information über eventuell extern begrenzte Verbindungen eine Vorhersage über die zukünftige Auslastung des betrachteten Links. Diese wiederum kann zur Optimierung der Regelung herangezogen werden.

C.2 Einfluss auf den Rufaufbau “normaler” ATM-Verbindungen

Die Vergabe von Bandbreite an FBR-Verbindungen erfolgt im Prinzip wie bei der Rufannahme herkömmlicher ATM-Verbindungen, allerdings mit dem Unterschied, dass den jeweiligen Verbindungen so viel Kapazität wie möglich zugewiesen wird. Im Normalfall ist daher das betrachtete Link bezogen auf die gemäß CAC zur Verfügung stehenden Bandbreite gänzlich ausgelastet. Kommt nun eine neue Verbindung hinzu, müsste diese eigentlich abgelehnt werden, da alle verfügbare Bandbreite bereits vergeben ist. Dies geschieht natürlich nicht von vorneherein, sondern es wird stattdessen geprüft, ob die für die neue Verbindung geforderte (minimale) Kapazität durch eine entsprechende Herabregelung der bestehenden FBR-Verbindungen bereitgestellt werden kann. Das Ergebnis dieser Prüfung entscheidet dann über Annahme oder Ablehnung des neuen Rufs. Alternativ kann immer eine gewisse Bandbreite für neue Verbindungen freigehalten werden, die bei deren Aufbau dann sofort zur Verfügung steht.

Die eben beschriebene Herabregelung der FBR-Quellen *bevor* eine neue Verbindung zugelassen wird bedingt so unter Umständen eine zeitliche Verzögerung der Rufannahme

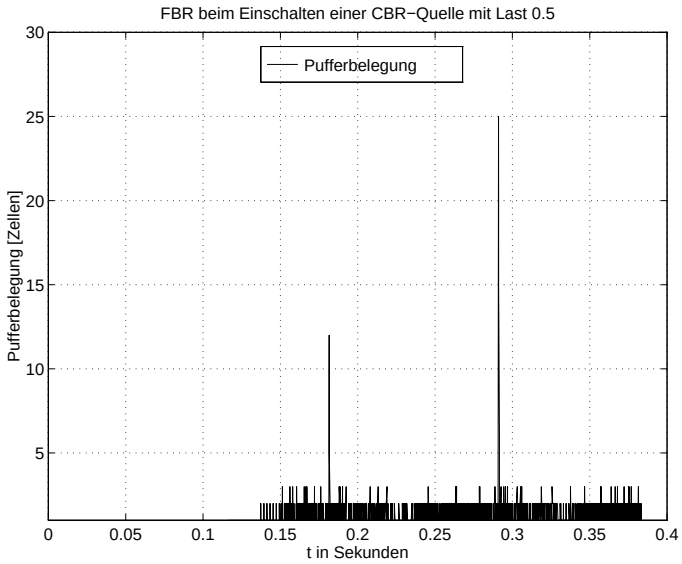


Abbildung C.1: Pufferbelegung beim Einschalten % einer CBR-Quelle ohne Verzögerung.

bzw. der eigentlichen Aktivierung der neuen Quelle. Da der Rufaufbau über alle Knoten der zukünftigen Verbindung selber Zeit benötigt, sind Szenarien möglich bei denen die Quelle sofort nach der unverzögerten Annahme des Rufs Datenzellen senden darf. Ist jedoch die für den Aufbau des neuen Rufes benötigte Zeit kurz und die für die Regelung der betroffenen FBR-Verbindungen erforderliche Zeit lang, muss der Beginn des eigentlichen Datentransfers verzögert werden. Geschieht dies nicht, können wie in Bild C.1 beim Aufbau zweier CBR-Verbindungen parallel zu drei FBR-Verbindungen dargestellt wird.

Je nach zu erwartendem Verkehr kann sowohl die verzögerte Rufannahme als auch das “Caching” von freier Bandbreite die effektivere Lösung sein. Im Sinne eines flexiblen Netzes ist es daher sinnvoll beide Verfahren zu implementieren und den Übergang vom einen zum anderen dem Verkehr entsprechend zu parametrisieren.

Anhang D

Abkürzungen

AAL	ATM Adaption Layer
ABR	Available Bit Rate
ACK	Acknowledgment
ARIMA	Auto Regression Integrated Moving Average
AS	Autonomous Systems
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband ISDN
BGP	Border Gateway Protocol
BRM	Backward RM (cells)
B-WiN	Breitband-Wissenschafts Netz
CAC	Call Admission Control
CBR	Constant Bit Rate
CCR	Current Cell Rate
CDV	Cell Delay Variation
CI	Congestion Indication
CLR	Cell Loss Rate
CRC	Cyclic Redundancy Check
DFN	Deutsches ForschungsNetz (Verein)
EFCI	Explicit Forward Congestion Indication
EPD	Early Packet Discard

ERICA	Explicit Rate Indication for Congestion Avoidance
FBR	Free Bit Rate Service Category
FDDI	Fiber Distributed Data Interface
FIFO	First In First Out
FTP	File Transfer Protocol
FRM	Forward RM (cells)
HOF	Higher Order Functions
HTML	HyperText Markup Language
HTTP	Hyper Text Transport Protocol
IAT	Inter-Arrival Time
IETF	Internet Engineering Task Force
IGRP	Internet Gateway Routing Protocol
IP	Internet Protocol
ISDN	Integrated Services Digital Network
KR	Kunden Router
KSS	Kunden-Service-Switch
LAN	Local Area Network
LRD	Long Range Dependence
MAN	Metropolitan Area Network
MCR	Minimum Cell Rate
MPI	Message Passing Interface
MPOA	Multi-Protocol Over ATM
MPLS	Multi-Protocol Label Switching
MSS	Maximum Segment Size (maximale Segmentgröße in TCP)
MTU	Maximum Transfer Unit
Nrm	Number of data cells between RM cells
NNI	Network Network Interface
NT	Network Termination
OSPF	Open Shortest Path First
PCR	Peak Cell Rate

PDU	Protocol Data Unit
PNNI	Private Network to Network Interface
PPD	Partial Packet Discard
PTI	Payload Type Identifier
QoS	Quality of Service
RDF	Rate Decrease Factor
RFC	Request For Comment
RIF	Rate Increase Factor
RM	Resource Management (cells)
RTT	Round Trip Time
SDU	Service Data Unit
SRD	Short Range Dependence
SupFRP	Superposition of Fractal Renewal Processes
TCP	Transmission Control Protocol
TE	Terminal Equipment
TPT	Truncated Power-Tail
TUHH	Technische Universität Hamburg-Harburg
UBR	Unspecified Bit Rate service category
UNI	User Network Interface
VBR	Variable Bit Rate
VC	Virtual Channel
VCI	Virtual Channel Identifier
VP	Virtual Path
VPI	Virtual Path Identifier
WAN	Wide Area Network
WR	WiN-Router
WWW	World Wide Web
ZSS	Zentraler Service-Switch

Literaturverzeichnis

[AD98] M. Aron und P. Druschel. TCP: Improving Startup Dynamics by Adaptive Timers and Congestion Control. Technical report, Rice University, jun 1998. http://cs-tr.cs.rice.edu/Dienst/UI/2.0/Describe/ncstrl.rice_cs/TR98-318.ps.gz.

[af-97] ATM Forum. *LAN Emulation Over ATM Version 2.0-LUNI Specification*, jul 1997. http://www.atmforum.org/atmforum/specs.approved.html.

[APS] M. Allman, V. Paxson und W. Stevens. RFC 2581: TCP Congestion Control. http://www.ietf.org/rfc/rfc2581.txt.

[AX96] Ambalavanar Arulambalam und Xiaoquiang. Allocating Fair Rates for Available Bit Rate Service in ATM-Networks. *IEEE Communications Magazine*, ISSN 01636804(34 (11)):92–101, 1996.

[Ber94] Jan Beran. *Statistics for Long-Memory Processes*, Band 61 in *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1994. ISBN 0-412-04901-5.

[CB97] Mark E. Crovella und Azer Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on networking*, 5(6):835–846, dec 1997. http://www.cs.bu.edu/faculty/crovella/paper-archive/self-sim/journal-version.ps.

[CCLS01] Jin Cao, S. William Cleveland, Dong Lin und Don X. Sun. Internet Traffic Tends To Poisson and Independent as the Load Increases. Technical report,

Bell Labs, Murray, Hill, NJ, 2001. http://cm.bell-labs.com/stat/doc/ip.poissonindependent.pdf.

[CIS] CISCO. *IP Routing Protocols*. http://www.cisco.com/univercd/cc/td/doc/product/software/ssr91/csc_r/56294.pdf.

[DG96] Patrick W. Droz-Georget. *Traffic Estimation and Resource Allocation in ATM Networks*. Phd thesis no. 11462, Swiss Federal Institute of Technology Zurich, 1996. http://www.zurich.ibm.com/~dro/thesis.ps.gz.

[FH] S. Floyd und T. Henderson. RFC 2582: The NewReno Modification to TCP's Fast Recovery Algorithm. http://www.ietf.org/rfc/rfc2582.txt.

[Flo95] Sally Floyd. TCP and Successive Fast Retransmits. Technical report, Lawrence Berkeley Labarotory, One Cyclotron Road, Berkeley, may 1995. http://www.aciri.org/floyd/notes.html.

[GAN91] R. Gu'erin, H. Ahmadi und M. Naghshineh. Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, September 1991.

[GGS97] Helmut Gogl, Michael Greiner und Hans-Peter Schwefel. Model Calibration. Technical report, Technische Universität München, Institut für Informatik, Dezember 1997. http://wwwjessen.informatik.tu-muenchen.de/forschung/leistung/ATM_Project/repms8.ps.gz.

[Gre97a] Michael Greiner. How to Model Telecommunications (and Other) Systems Where S̈elf-SimilarB̈ehavior is Observed. Technical report, Technische Universität München, Institut für Informatik, sep 1997. http://wwwjessen.informatik.tu-muenchen.de/forschung/leistung/ATM_Project/rep_extra.ps.gz.

[Gre97b] Michael Greiner. Requirements on Traffic Source Models for ATM Networks. Technical report, Technische Universität München, Institut für Informatik, September 1997. http://wwwjessen.informatik.tu-muenchen.de/forschung/leistung/ATM_Project/repms3.ps.gz.

[Jac88] Van Jacobson. More on TCP Congestion Control. Email, Lawrence Berkeley National Laboratory, Februar 1988. ftp://ftp.ftp.ee.lbl.gov/email/vanj.88feb23.txt.

[Jac90] Van Jacobson. Modified TCP Congestion Control Avoidance Algorithm. Technical report, Lawrence Berkeley National Laboratory, apr 1990. ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt.

[JK88] Van Jacobson und Michael J. Karels. Congestion Avoidance and Control. *ACM SIGCOMM*, S. 314–329, aug 1988. http://ana.lcs.mit.edu/nrg/papers/congavoid.ps.

[Kal97] Shivkumar Kalayanaraman. *Traffic Management for Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks*. Dissertation, The Ohio State University, 1997. http://www.cis.ohio-state.edu/~jain/theses/shiv.htm.

[KK96] Bo-Kyoung Kim und G.Byung Kim. Comparison of ABR Control Algorithms with Explicit Rate Feedback. In *Proceedings of Technical Conference on Telecommunications R&D in Massachusetts*, March 1996. http://www.cs.uml.edu/~bkim/research.html.

[Mor00] Robert Morris. Scalable TCP Congestion Control. In *IEEE INFOCOM 2000, Tel Aviv*, S. 1176–1183, März 2000. http://www.pdos.lcs.mit.edu/~rtm/papers/tp.pdf.

[MR99] L. Massouli'e und J. Roberts. Arguments in favour of admission control for TCP flows. In *ITC 16, Edinburgh*, 1999. http://research.microsoft.com/users/lmassoul/itc.ps.

[Pax97] Vern Paxson. *Measurement and Analysis of End-to-End Internet Dynamics*. Dissertation, Lawrence Berkeley National Laboratory, apr 1997. ftp://ftp.ee.lbl.gov/papers/vp-thesis/dis.ps.z.

[Pit96] Pitsburgh Supercomputing Center. *Forward Acknowledgement: Refining TCP Congestion Control*, aug 1996. ftp://ftp.psc.edu/pub/networking/papers/fack.9608.ps. Proceedings of ACM Sigcomm.

[PKC96] Kihong Park, Gitae Kim und Mark Crovella. On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic. Technical report, Boston University, jul 1996. http://www.cs.bu.edu/faculty/crovella/paper-archive/files-protocols/TR-\%96-016.ps.

[RMV96] James Roberts, Ugo Mocci und Jorma Virtamo, Hrsg. *Broadband Network Teletraffic, Final Report of Action COST 242*. Number 1155 in Lecture Notes in Computer Science. Springer, 1996.

[RN97] Bong K. Ryu und Mahesan Nandikesan. Real-Time generation of Fractal ATM Traffic: Model, Algorithm, and Implementation. Technical report, Comet, 1997. http://www.comet.ctr.columbia.edu/~mahesan/pub/fractal.ps.gz.

[RVC] E. Rosen, A. Viswanathan und R. Callon. RFC 3031: Multiprotocol Label Switching Architecture. http://www.ietf.org/rfc/rfc3031.txt.

[Sch97] Hans-Peter Schwefel. Modeling of Packet Arrivals Using Markov Modulated Poisson Processes with Power-Tail Bursts. *Diplomarbeit, TU München*, August 1997. http://wwwjessen.informatik.tu-muenchen.de/~schwefel/da.ps.gz.

[SSO96] Shirish Sathaye, Vijay Samalam und Jim Ormord. Traffic Management Specification - Version 4.0. Technical report, The ATM Forum - Technical Committee, April 1996.