Universität Stuttgart

INSTITUT FÜR
NACHRICHTENVERMITTLUNG
UND DATENVERARBEITUNG
Prof. Dr.-Ing. Dr. h. c. mult. P. J. Kühn

Entwurf und Bewertung von Puffermechanismen für Optical Burst Switching

Studienarbeit

Michael Scharf

Betreuer

[redacted]

Stuttgart, Juli 2001

Zusammenfassung

Der Einsatz optischer Übertragung und Vermittlung („Photonische Netze“) wird für die künftige Entwicklung des Internets eine Schlüsselrolle spielen. Optical Burst Switching (OBS) ist ein neuer, vielversprechender Ansatz, Wellenlängenkanäle dynamisch zu belegen. Diese Netzarchitektur bietet die Möglichkeit, Internet-Verkehr direkt über die optische Schicht zu übertragen und es kann dabei auch verschiedenen Dienstgüteanforderungen entsprochen werden.

Im Rahmen dieser Studienarbeit werden zur Verbesserung der Systemeigenschaften von OBS-Netzen optische Puffermechanismen eingeführt. Ausgehend von in der Literatur vorgeschlagenen optischen Knotenarchitekturen werden Pufferstrukturen modelliert und ein bestehendes Simulationsprogramm wird dahingehend erweitert. Es werden zwei verschiedene Pufferstrategien vorgeschlagen und anschließend in umfangreichen Simulationsstudien vergleichend bewertet.

Dazu wird zunächst die Blockierungsbehandlung durch einfache Verzögerungsleitungen betrachtet. Es zeigt sich, dass schon mit einfachen Puffern die Burstverlustwahrscheinlichkeit deutlich gesenkt werden kann. Anschließend werden die Untersuchungen auf komplexere Pufferstrukturen ausgedehnt. Mit diesen sind weitere Verbesserungen erreichbar – allerdings auf Kosten zunehmender Transferzeiten. Die vorgeschlagenen Pufferstrategien sind für verschiedene Pufferstrukturen unterschiedlich gut geeignet. Abschließend wird der Einfluss von Puffern auf die Verwendung Offset-basierter Verkehrsdifferenzierung in OBS betrachtet.

Abstract

Optical transmission and switching (photonic networks) will play a key role in the future development of the Internet. Optical Burst Switching (OBS) is a new, promising paradigm that dynamically allocates wavelength channels. With this network architecture Internet traffic can be transmitted directly over the optical layer while providing Quality of Service.

In this study project optical buffer mechanisms are implemented into OBS networks in order to improve the system performance. Based on optical switch architectures known from literature buffers are modelled and an existing simulation tool is extended accordingly. Two different buffer strategies are proposed and then analyzed by extensive simulations.

First contention resolution by simple fiber delay lines is considered. It can be shown that even simple buffers reduce the burst loss probability significantly. Subsequently, the analysis is extended to more complex buffer architectures. By this mean further improvements can be achieved – at the expense of increasing delays. The suggested buffer strategies are particularly suitable for different buffer structures. Finally, the effects of buffers on OBS systems with offset-based service differentiation are studied.

Inhaltsverzeichnis

Inhaltsverzeichnis **i**

Abbildungsverzeichnis **iii**

Tabellenverzeichnis **iv**

Abkürzungen **v**

Formelzeichen **vi**

1 Einleitung **1**

1.1 Motivation 1

1.2 Photonische Netze 1

1.3 Überblick über diese Arbeit 2

2 Optical Burst Switching **3**

2.1 Übersicht 3

2.2 Netz-Architektur 3

2.3 Switch-Architekturen 5

2.4 Reservierungsverfahren 7

3 Grundlagen optischer Puffermechanismen **9**

3.1 Übersicht 9

3.2 Optische Verzögerungsleitungen 9

3.3 Optische Puffer 11

3.3.1 Architektur und Struktur 11

3.3.2 Puffermanagement 14

3.4 Zusammenfassung 18

4 Untersuchung optischer Puffermechanismen **19**

4.1 Übersicht 19

4.2 Modell und Parameter 19

4.3 Abschätzung 22

4.4 Einmaliger Pufferdurchlauf 23

4.4.1 Burstverlustwahrscheinlichkeit 23

4.4.2 Durchlaufzeit 27

4.4.3 Bewertung 29

4.4.4 Untersuchung anderer Längenverteilungen 31
4.5 Mehrmaliger Pufferdurchlauf 34
4.5.1 Burstverlustwahrscheinlichkeit 34
4.5.2 Durchlaufzeit 37
4.5.3 Bewertung 39
4.6 Puffer mit mehreren Verzögerungsleitungen 40
4.6.1 Burstverlustwahrscheinlichkeit 40
4.6.2 Durchlaufzeit 45
4.6.3 Dimensionierung der Verzögerungsleitungen 47
4.6.4 Bewertung 48
4.7 Pufferung bei differenzierter Behandlung von Verkehrsklassen 50
4.7.1 Prinzip der Klassentrennung 50
4.7.2 Zwei Verkehrsklassen 50
4.7.3 Verallgemeinerung 54
4.7.4 Bewertung 55

5 Zusammenfassung und Ausblick 56

A Simulationsprogramm 58
A.1 Überblick 58
A.2 Architektur 58
A.3 Implementierung 61
A.4 Bedienung 66

B Optimiertes JET-Verfahren 71
B.1 Theoretische Überlegungen 71
B.2 Implementierung 73
B.3 Untersuchungsergebnisse 73
B.4 Bewertung 75

Literaturverzeichnis 76

Abbildungsverzeichnis

2.1 Architektur eines OBS Netzes 4
3.1 Eigenschaften von Verzögerungsleitungen 10
3.2 Klassifizierung von optischen Puffern 11
3.3 Puffer-Architekturen 12
3.4 Szenario eines Puffer-Vorgangs 14
3.5 Datenflussdiagramm für PriorRes und PostRes 17
4.1 Verlustwahrscheinlickkeiten über Last 24
4.2 Verlustwahrscheinlickkeiten über Verzögerung (1) 25
4.3 Verlustwahrscheinlickkeiten über Verzögerung (2) 26
4.4 Blockier- und Wiedereintrittswahrscheinlichkeiten 27
4.5 Anteil gepufferter Bursts über Last 28
4.6 Mittlere Durchlaufzeit über Verzögerung 28
4.7 Puffer-Belegung über Verzögerung 29
4.8 Verlustwahrscheinlichkeit in Abhängigkeit der Burstlänge 31
4.9 PriorRes mit anderen Burstlängenverteilungen 32
4.10 PostRes mit anderen Burstlängenverteilungen 33
4.11 Verlustwahrscheinlichkeiten bei mehreren Durchläufen 34
4.12 Normierte Verlustwahrscheinlichkeit über Kapazität 35
4.13 PostRes-Verlustwahrscheinlichkeiten über Kapazität 36
4.14 Mittlere Durchlaufzeit bei mehreren Durchläufen 37
4.15 Komplementäre Verteilungsfunktion der Transferzeiten 38
4.16 Verlustwahrscheinkichkeiten für mehrere FDL 41
4.17 PriorRes-Verlustwahrscheinlichkeiten für mehrere FDL 42
4.18 PostRes-Verlustwahrscheinlichkeiten für mehrere FDL 43
4.19 PostResRandom-Verlustwahrscheinlichkeiten für mehrere FDL 44
4.20 Belegung einzelner FDL 45
4.21 Mittlere Durchlaufzeiten für mehrere FDL 46
4.22 Komplementäre Verteilungsfunktion der Durchlaufzeiten 46
4.23 Zusammenhang Verlustwahrscheinlichkeit und Durchlaufzeit (1) . . . 47
4.24 Zusammenhang Verlustwahrscheinlichkeit und Durchlaufzeit (2) . . . 47
4.25 Verlustwahrscheinlichkeiten bei zwei Verkehrsklassen 51
4.26 hp-Burstverlustwahrscheinlichkeiten bei PostRes 53
4.27 hp-Burstverlustwahrscheinlichkeiten bei PriorRes 54
A.1 Architektur eines OBS Knotens 59

A.2 Klassendiagramm CBlockMgr 61
A.3 Klassendiagramm CDelayLine 63
A.4 Klassendiagramm CBufferEvent 64
A.5 Sequenz-Diagramm 65
B.1 JETOpt-Szenario 72
B.2 Vergleich von JET und JETOpt 74

Tabellenverzeichnis

3.1 Vergleich von Puffer-Architekturen 13
4.1 Abschätzung der Verlustwahrscheinlichkeiten 23
4.2 Auswirkung von Puffern bei PriorRes 52
A.1 Parameter für Blockierungsmanager 70
A.2 Parameter für Pufferstrategie 70
A.3 Übersicht über FDL-Typen 70

Quelltexte

Parameterdatei 66

Abkürzungen

ASE	Amplified Spontaneous Emission
ATM	Asynchronous Transfer Mode
AWG	Arrayed Waveguide Grating
D-WDM	Dense WDM
EDFA	Erbium-Doped Fiber Amplifier
FDL	Fiber Delay Line
FIFO	First In, First Out
IAT	Inter Arrival Time
IP	Internet Protokoll
JET	Just-Enough-Time
JIT	Just-In-Time
LWL	Lichtwellenleiter
MPLS	Multi Protocol Label Switching
MPλS	Multi Protocol Lambda Switching
OBS	Optical Burst Switching
OPS	Optical Paket Switching
PSC	Passive Star Coupler
QoS	Quality of Service
RAM	Random Access Memory
SCDT	Separate Control Delayed Transmission
SDH	Synchronous Digital Hierarchy
SOA	Semiconductor Optical Amplifier
SONET	Synchronous Optical Network
TWC	Tunable Wavelength Converter
UML	Unified Modeling Language
WDM	Wavelength Division Multiplex

Formelzeichen

$E[\cdot]$	Erwartungswert einer Zufallsvariable
c_T	Variationskoeffizient einer Verteilung
n	Anzahl Wellenlängen
n_b	Anzahl Wellenlängen im Puffer
N	Anzahl an FDLs im Puffer
b	Verzögerung durch eine FDL
B	maximal mögliche Verzögerung eines Puffers
Q	maximal zulässige Anzahl an Pufferdurchläufen
r	Datenrate (in Bit/s)
ρ	Last
$A = \rho\ n$	Angebot
t_IAT	Zwischenankunftszeit (*inter arrival time*)
g	Anzahl an verwendeten Verkehrsgeneratoren
L	Burstlänge (in Bytes)
$h = L/r$	Burstübertragungsdauer (in Sekunden)
δ	Offset zwischen Kontrollteil und Datenteil eines Bursts (in Sekunden)
Δ	Dauer zwischen Reservierung und Datenteil eines Bursts (in Sekunden)
T_Q	Durchlaufzeit (*queueing time*) durch einen Knoten (in Sekunden)

1 Einleitung

1.1 Motivation

Die Zahl der Internetbenutzer und der Internetverkehr hat in den letzten Jahren explosionsartig zugenommen, und ein Ende dieser Entwicklung ist nicht in Sicht. Der Bedarf an Bandbreite nimmt auch durch neue Multimedia-Dienste, Internet-Telefonie, Online-Spiele usw. ständig zu. Gleichzeitig existiert eine steigende Nachfrage nach garantierter Dienstgüte (*quality of service*, QoS).
Es zeichnet sich ab, dass nur photonische Netze in Verbindung mit Wellenlängenmultiplex (*wavelength division multiplex*, WDM) in der Lage sind, diesen Anforderungen gerecht zu werden.

1.2 Photonische Netze

Von photonischen Netzen wird gesprochen, wenn sowohl die Übertragung als auch die Vermittlung von Daten optisch realisiert sind. Bisher übernehmen hardwarebasierte elektrische Hochgeschwindigkeitsrouter bzw. -switche letztere Aufgabe, bleiben dabei aber zunehmend hinter den Fähigkeiten von D-WDM (*dense WDM*) mit 80-120 genutzten Wellenlängen und Datenübertragungsraten von mehreren Gbit/s zurück. Fortschritte bei optischen Systemen und Komponenten wie schnellen Koppeleinrichtungen, integrierten optischen Halbleiterverstärkern (*semiconductor optical amplifier*, SOA), verstellbaren Wellenlängenkonvertern (*tunable wavelength converter*, TWC), Filtern und Lasern machen optische Vermittlungssysteme möglich, die besser als elektrische Router skalieren und die Engpässe („bottlenecks“) im elektrischen Bereich überwinden können.
Durch die Vermeidung von E/O- und O/E-Wandlungen können Daten bezüglich Bitrate und Codierungsformaten transparent übertragen werden. Die Bitrate wird daher nur durch physikalische Effekte (z. B. Dispersion) begrenzt. Praktisch wird die Entwicklung photonischer Netze aber dadurch erheblich erschwert, dass mit der momentan bzw. auch in absehbarer Zukunft verfügbaren Technologie allerhöchstens eine rudimentäre optische Signalverarbeitung möglich ist. Zur Pufferung bzw. Verzögerung von Daten stehen zwar Faserschleifen zur Verfügung, ein optischer Speicher mit wahlfreiem Zugriff (RAM) ist aber nicht verfügbar. Daher ist eine Übertragung der im elektrischen Bereich bekannten Verfahren auf die optische Schicht nur sehr eingeschränkt möglich.
Im Internet dominiert das Internet Protocoll (IP). Um die Komplexität und den

Overhead zu reduzieren, der sich z. B. bei der Übertragung von IP über ATM über SDH/SONET ergibt, gibt es Bestrebungen, IP direkt (bzw. mit nur wenigen schlanken Zwischenschichten) über WDM zu übertragen (IP-over-WDM, *optical internet*) [10]. Das Internet basiert zur Zeit auf dem „best effort"-Prinzip, es besteht aber zunehmend die Notwendigkeit, Dienstgüte zur Verfügung stellen zu können. Dafür existieren bereits verschiedene Ansätze [11]. Eine IP-over-WDM-Lösung sollte daher QoS-fähig sein, d. h. neben einer hohen Bandbreite und einer großen Verfügbarkeit auch eine niedrige Verlustwahrschlichkeit, geringe Verzögerung und wenig Verzögerungsvarianz (*jitter*) bereitstellen können. Ein vielversprechender derartiger Ansatz, der die spezifischen Eigenschaften der WDM-Schicht ausnutzt, ist Optical Burst Switching (OBS). Um die Systemeigenschaften zu verbessern und Blockierungen zu vermeiden, können optische Puffermechanismen eingesetzt werden. Diese werden im Rahmen dieser Arbeit untersucht.

1.3 Überblick über diese Arbeit

Eine Einführung in Optical Burst Switching ist in Kapitel 2 enthalten. Um die Verwendung von optischen Puffern im Zusammenhang mit OBS untersuchen zu können, befinden sich in Kapitel 3 einige prinzipielle Überlegungen zu optischen Puffermöglichkeiten. Im vierten Kapitel werden diese für einfache Szenarien anhand von Simulationsstudien untersucht; anschließend wird die Betrachtung auf mehrere Verkehrsklassen ausgedehnt. Neben einer Zusammenfassung und einem Ausblick enthält die Arbeit noch zwei Anhänge: Anhang A dokumentiert das verwendete Simulationstool, während sich Anhang B mit einem optimierten Reservierungsverfahren für OBS befasst.

2 Optical Burst Switching

2.1 Übersicht

In bisher existierenden Netzen werden prinzipiell zwei verschiedene Vermittlungsprinzipien verwendet: Systeme mit Leitungsvermittlung (*circuit switching*), die zur Datenübertragung eine Verbindung auf- und wieder abbauen, und paketvermittelnde Netze, die auf Speichervermittlung (*store-and-forward*-Prinzip) beruhen. Beide Ansätze lassen sich auch auf photonische Netze übertragen:
Durchschaltevermittlung (*optical pipes*, *light paths*, *optical circuit switching*) wird bereits in SDH/SONET-Netzen angewendet, ist aber unflexibel und für typischen Internet-Verkehr (unsymmetrisch und büschelförmig in jeder Zeitskala) wenig effizient. Angesichts der nur eingeschränkt möglichen optischen Signalverarbeitung ist eine reine optische Paketvermittlung (*optical packet switching*, OPS) [10, 18] auf absehbare Zeit nicht realisierbar – sowohl mit festen als auch mit variablen Paketlängen.
Optical Burst Switching [12] kombiniert beide Ansätze: Pakete werden zu größeren Einheiten (Bursts) zusammengesetzt und dann gemeinsam übertragen. OBS stellt somit eine Weiterentwicklung des schon für elektrische Systeme bekannten Fast Circuit Switching [8] dar. Diese Grundidee steht auch hinter dem momentan entwickelten MPLS [14]. Ziel von MPLS (bzw. auch MPλS) ist es, innerhalb eines Flusses aus IP-Paketen anhand von „Labels“ einen virtuellen Kanal aufzubauen, so dass eine Granularität erreicht wird, die größer als die eines einzelnen IP-Paketes ist. Eine Realisierung von Optical Burst Switching in Verbindung mit MPLS (oder auch *optical label switching*) ist daher möglich und erscheint attraktiv.

2.2 Netz-Architektur

Das Haupteinsatzfeld von photonischen Netzen werden Kernnetze sein, z. B. *backbone networks*. Ein solches optisches Netz, das auf den Verbindungsfasern im Wellenlängenmultiplex-Verfahren (WDM) arbeitet, ist in Abbildung 2.1 dargestellt.
In einem OBS-Netz sind zwei Arten von Knoten vorhanden: Kernknoten (*core nodes*) fungieren als reine Transitknoten. Ihre Aufgabe ist es, für die Dauer eines Bursts einen physikalischen optischen Pfad zwischen den entsprechenden Eingangs- und Ausgangsanschlüssen herzustellen, ohne dabei die Bursts aufwendig verarbeiten zu müssen. Auf die Architektur dieser Knoten wird in Abschnitt 2.3 näher eingegangen. In den Randknoten (*edge nodes*) werden Pakete aus den umliegenden Netzen

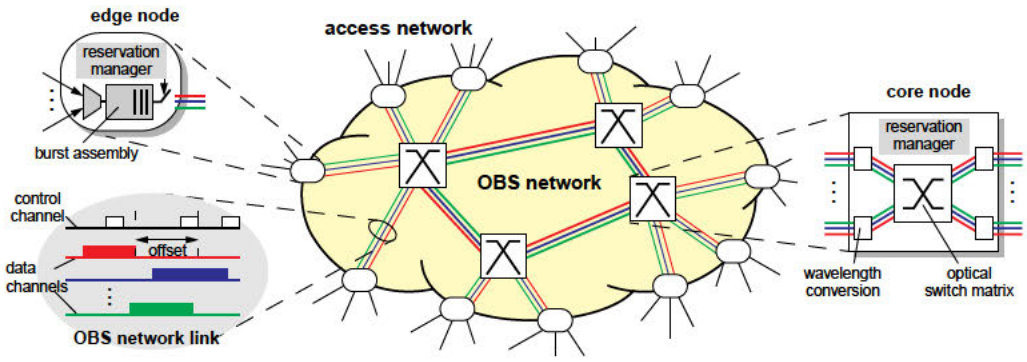

Abb. 2.1: Architektur eines OBS Netzes

gesammelt und zu größeren Einheiten zusammengefasst (*burst assembly*). Die bei dieser Verkehrsaggregation entstehenden Bursts haben bei OBS eine variable Länge, die deutlich über der eines einzelnen IP-Paketes liegen wird. Eine Verwendung von Bursts fester Länge wäre zwar leichter zu implementieren, würde aber zu mehr Overhead führen und auch weniger flexibel sein, insbesondere bei dem Zusammensetzen der Bursts. Eine der wichtigsten Aufgaben bei der Dimensionierung von OBS-Netzen wird die Ermittlung von geeigneten *assembly*-Strategien für die Randknoten sein, da dadurch fast alle anderen Verkehrsparameter im Netz beeinflusst werden und auch eine gezielte Verkehrsformung erreicht werden kann. Erste Untersuchungen zu diesem Thema finden sich in [3, 13].

Ein Burst besteht aus einem Kontrollpaket (*header*) und einem Datenteil (*payload*). Zwischen Kontrollpaket und dem Beginn der Datenübertragung kann eine Verzögerung, Offset genannt, existieren. Weil Optical Burst Switching auf Außerbandsignalisierung basiert, werden die Kontrollpakete in einem paketvermittelten, hybriden Netz übertragen, das eigene Kanäle, z. B. getrennte Wellenlängen, verwendet. Die Knoten verarbeiten die Kontrollpakete elektrisch und sind so in der Lage, die Vermittlungseinrichtungen auf die optische Übertragung des Datenteils einzustellen. Diese Aufspaltung hat auch den Vorteil, dass für beide Burstteile nicht das gleiche Datenformat bzw. die gleiche Datenübertragungsrate eingesetzt werden muss. Die Kontrollpakete müssen die Zieladresse des Bursts bzw. ein Label enthalten sowie Informationen über den verwendeten Offset, die Burstlänge und andere Parameter (z. B. das verwendete Modulationsverfahren für den Datenteil). Hier kann – wie schon erwähnt – MPLS eingesetzt werden, in [16] wird aber auch vorgeschlagen, für diese Pakete das ATM-Zell-Format zu verwenden. Außerdem sind vermutlich Synchronisationssequenzen vorzusehen.

Erreicht ein Kontrollpaket einen Knoten, so wird versucht, zu dem geforderten Zeit-

punkt die notwendigen Ressourcen bereitzustellen. Das Kontrollpaket wird dann an den nächsten Knoten weitergeleitet. Dies hat zur Folge, dass mit jedem durchlaufenen Knoten die Zeitdauer zwischen dem Kontrollpaket und dem eigentlichen Burst um die Verarbeitungszeit im Knoten reduziert wird. Der Offset besteht daher anfangs aus einem Basiswert, der groß genug sein muß, um die Verarbeitungszeiten in allen Knoten zu kompensieren und zusätzlich auch noch die Einstellzeiten einer Vermittlungsmatrix zu berücksichtigen. Zusätzlich kann man den Offset um einen weiteren Anteil vergrößern, einen sogenannten QoS-Offset. Dies ermöglicht die Unterscheidung zwischen verschieden priorisierten Verkehrsklassen und damit die Bereitstellung von Dienstgüte. Deterministische Garantien sind allerdings dadurch nicht möglich. Die Behandlung von differenzierten Verkehrsszenarien ist Gegenstand von Kapitel 4.7.
Ein Offset wäre nicht notwendig, wenn der Datenteil am Eingang jedes Knotens gepuffert würde, bis die Informationen im Kopf ausgelesen sind. Optical Burst Switching wird damit zu den SCDT-Verfahren gezählt (*separate control, delayed transmission*).
Bei optischer Nachrichtenübertragung im GBit/s-Bereich ist die Signallaufzeit über längere Strecken – mehrere hundert Kilometer – wesentlich größer als die Übertragungsdauer eines Bursts, selbst wenn dieser einige Megabyte umfasst. Der Aufbau einer Verbindung, z. B. mit einem *hand-shake*-Protokoll dauert daher sehr lange. Aus diesem Grund wird bei OBS auf eine Bestätigung der in dem Kontrollpaket enthaltenen Reservierungsanforderung verzichtet (*one-pass-reservation*). Bei dieser asynchronen Übertragung ohne Rückkanal kann es zwar vorkommen, dass ein Burst trotz vorheriger Reservierung mangels Ressourcen verworfen werden muss. Sie ermöglicht aber – trotz des Offsets – sehr geringe Verzögerungszeiten und ist auch gut für unsymmetrische Verkehrsaufkommen geeignet.
In Optical Burst Switching-Netzen betrachtet man eine Faser als *shared medium* und nutzt die Wellenlängen im asynchronen Zeitmultiplexverfahren, wenn auch mit größerer Granulatität wie in klassischen paketvermittelnden Netz-Architekturen. Dies entspricht einem statistischen Multiplexen und nutzt insbesondere für Internetverkehr die von WDM zur Verfügung gestellte Bandbreite effizient aus.

2.3 Switch-Architekturen

Ein Vermittlungsknoten besteht prinzipiell aus einem Eingangswerk, Vermittlungseinrichtungen, einem Ausgangswerk und möglicherweise zentralen Steuereinrichtungen. Für elektrische Paketvermittlungsknoten sind drei prinzipielle Architekturen

bekannt [8]: Knoten, in denen jedem Ausgang ein Puffer zugeordnet ist (*output buffer*), vermitteln ein eintreffendes Paket sofort zum Zielausgang. In dessen Puffer wird das Paket so lange in einer FIFO-Warteschlange gespeichert, bis eine Übertragung möglich ist. Eine dazu analoge Anordnung besteht aus Puffern an jedem Knoteneingang (*input buffer*). Bei dieser Struktur besteht aber das Problem des sogenannten *head-of-line-blocking*, das den Durchsatz beschränkt. Der dritte Ansatz beruht auf *shared media* und wird oft mit Hilfe von RAM realisiert.
In den letzten Jahren wurden umfangreiche Forschungen zum Thema optische Switch-Architekturen durchgeführt und zahlreiche Architekturvorschläge erarbeitet. Diese werden in [7, 15, 18, 19] vorgestellt und in ihren technologischen Eigenschaften verglichen. Wie schon erläutert wurde, steht in absehbarer Zukunft kein optischer Speicher mit wahlfreiem Zugriff zur Verfügung, so dass eine Übertragung eines *shared buffer*-Prinzips auf optische Vermittlung nicht ohne weiteres möglich ist. Die meisten Vorschläge für Switches basieren daher auf einem *output buffer* oder einer Architektur, die diesem Prinzip ähnelt (*simulated output buffer*).
Ein optischer Transitknoten verarbeitet Kontroll- und Datenteil eines Bursts getrennt. Wird eine Reservierungsanforderung empfangen, so wird diese in einer Steuereinrichtung elektrisch verarbeitet und anhand der enthaltenen Informationen die entsprechende Ausgangsadresse ermittelt. Dann wird der optische Switch angewiesen, für die Dauer des Datenteils eine Verbindung zwischen Eingang und Ausgang herzustellen. Ist dies nicht möglich, können Mechanismen zur Behandlung dieser Zugriffskonflikte vorgesehen werden (*contention resolution*). Das Kontrollpaket selbst wird mit angepasstem Informationsteil auf den Kontrollkanal des entsprechenden Ausgangs vermittelt und weiter verschickt. Viele bislang vorgeschlagene Architekturen setzen eine Übertragung in (gleich langen) Zeitschlitzen voraus, so dass vor der Übertragung eine Synchronisation nötig ist.
Für die eigentliche optische Vermittlung existieren drei verschiedene Prinzipien: In einem *space switch* existieren räumlich getrennte Wege zu den jeweiligen Ausgängen, die durch Koppler und optische Tore getrennt ausgewählt werden können. Bei *wavelength routing* wird mit *arrayed waveguide gratings* (AWGs) anhand der Wellenlänge vermittelt. Dazu wird am Eingang durch abstimmbare Wellenlängenkonverter (*tunable wavelength converter*, TWC) die richtige Wellenlänge eingestellt. In einem *broadcast-and-select*-Switch werden Pakete aller Eingangsports – jeweils auf verschiedenen Wellenlängen – z. B. mit einem passiven Sternkoppler (*passive star coupler*) kombiniert und an alle Ausgangsports weitergeleitet. Diese wählen dann mit Filtern anhand der Wellenlänge die sie betreffenden aus. Diese Anordnung hat insbesondere

den Vorteil, dass ohne weiteren Aufwand ein *multicast*-Betrieb möglich ist.
In einem OBS-Knoten kann der Fall auftreten, dass mehr als ein Paket auf den gleichen Ausgang zuzugreifen versucht (externe Blockierung). Die naheliegende Lösung für diesen Zugriffskonflikt ist, optische Puffer zur Verfügung zu stellen. Darauf wird in den folgenden Kapiteln eingegangen. In vermaschten Netzen besteht aber auch die Möglichkeit, einen nicht direkt übertragbaren Burst an einen anderen Knoten weiterzuleiten (*deflection routing*), also eine andere Route zu wählen. Dieser Ansatz hängt aber stark von der Netztopologie ab, führt je nach Netz zu einer erheblichen Verzögerung und wird daher hier nicht weiterverfolgt. Eine dritte Möglichkeit besteht in der Nutzung des Wellenlängenbereichs: Ist die Ziel-Wellenlänge belegt, kann evtl. eine Konvertierung auf eine andere Wellenlänge der gleichen Leitung vorgenommen und der Bursts dort übertragen werden. Dieser Ansatz ist aber nur für Switch-Architekturen sinnvoll, die im Switch keine Wellenlängenkonvertierung vornehmen müssen, dazu aber optional in der Lage sind, beispielsweise also *space switches*. In anderen Architekturen ist eine beliebige Zuordnung von Eingangs- und Ausgangswellenlängen möglich. In diesem Fall ist entweder eine Übertragung auf irgendeiner Wellenlänge möglich, oder es steht überhaupt keine freie Wellenlänge zur Verfügung.
Optische Vermittlungseinrichtungen können einstufig (*single stage*) oder mehrstufig (*multi stage*) aufgebaut sein. Die meisten vorgeschlagenen Architekturen bestehen aus einer einzigen optisch realisierten Koppelanordnung, in der auch schon Verzögerungsleitungen als Puffer integriert sein können [7]. Es gibt aber auch einige mehrstufige Ansätze. Diese nutzen z. B. kaskadierte 2x2-Koppler, die relativ leicht zu konstruieren sind, aber höhere Dämpfungen aufweisen.
Prinzipiell kommt Optical Burst Switching völlig ohne Puffermechanismen in den Transitknoten aus, da im idealen Fall für die Dauer eines Bursts eine physikalische Ende-zu-Ende-Verbindungen aufgebaut wird. Dennoch ist es sinnvoll zu untersuchen, in wie weit Puffer sich zur Auflösung von Zugriffskonflikten und Überlastsituationen eignen. Diesen Untersuchungen sind die folgenden Kapitel gewidmet. Dabei wird stets davon ausgegangen, dass ein intern blockierungsfreier Switch zur Verfügung steht und keinerlei Einschränkungen bezüglich der Konvertierung von Wellenlängen bestehen.

2.4 Reservierungsverfahren

Bei Optical Burst Switching wird im Knoten nach der Verarbeitung des Kontrollpakets für die Übertragung der Daten eine Wellenlänge reserviert. Dafür wurden

verschiedene Verfahren vorgeschlagen:

1. Das einfachste Verfahren *Just-In-Time* (JIT) speichert zu einer Wellenlänge lediglich die Information, ob diese gerade belegt ist oder nicht.

2. Beim *Horizon*-Verfahren [16] wird ein sogenannter Reservierungshorizont eingeführt, d. h. ein Zeitpunkt, der für jede Wellenlänge das Ende der letzten vorliegenden Reservierung kennzeichnet. Eine Reservierung einer Wellenlänge ist dann möglich, wenn der Übertragungszeitraum erst nach diesem Zeitpunkt beginnt.

3. *Just-Enough-Time* (JET) geht auf Qiao zurück (u. a. [13]). Das Verfahren beruht auf dem *reserve-a-fixed-duration*-Prinzip und reserviert Wellenlängen genau für die Dauer des Bursts. Die schon erwähnte Verkehrsdifferenzierung durch einen QoS-Offset ist nur in Verbindung mit JET möglich.

Zu den drei Reservierungsverfahren liegen bereits umfangreiche Untersuchungen vor [4, 5]. JIT ermöglicht keinerlei Überlappung der Offsetzeit mit anderen Bursts, so dass es – je nach Wahl des Offsets – sehr ineffizient ist, dafür aber mit sehr wenig Aufwand realisiert werden kann. Das Horizon-Verfahren liefert für viele Anwendungsfälle ähnliche Resultate wie JET, insbesondere bei Verwendung von Puffern sind aber auch signifikante Unterschiede feststellbar [17]. JET ist die flexibelste der drei genannten Varianten und ermöglicht eine sehr effiziente Nutzung der Leitungen, da hier auch bei einer schon teilweise belegten Wellenlänge nachträglich Bursts eingefügt werden können. Allerdings ist die Implementierung von JET relativ aufwendig. Es ist noch ungeklärt, ob eine Verwendung des JET-Verfahrens unter den vorhandenen Randbedingungen, d. h. einer möglichst schnellen elektrischen Verarbeitung der Kontrolldaten, möglich sein wird.
In den Untersuchungen, die im Kapitel 4 vorgestellt werden, wird stets das JET-Verfahren verwendet, da bei diesem Verfahren die charakteristischen Einflüsse der Puffer deutlich feststellbar sind. Bei einer Verwendung anderer Verfahren muss mit Effekten gerechnet werden, die zusätzlich durch Reservierungsverfahren verursacht werden.

3 Grundlagen optischer Puffermechanismen

3.1 Übersicht

In diesem Kapitel werden die grundlegenden Eigenschaften und Charakteristika von optischen Puffern vorgestellt. Die Betrachtung beginnt mit einer einfachen Verzögerungsleitung und wird dann für allgemeine Pufferstrukturen erweitert. Nicht behandelt werden Fragen zur technologischen Realisierung, da diese stark von der konkret verwendeten Switch-Architektur abhängt und zu diesem Thema bereits zahlreiche Veröffentlichungen existieren [7, 15, 18, 19].
Ebensowenig berücksichtig wird hier die Möglichkeit, anstelle von optischen Puffern elektrische Speicher vorzusehen, die auch einen wahlfreien Zugriff erlauben würden. Dies würde eine aufwendige Datenkonvertierung erfordern, die in photonischen Netzen generell vermieden werden soll.

3.2 Optische Verzögerungsleitungen

Die einzige bislang entwickelte Möglichkeit, ein optisches Signal rein optisch zu verzögern, besteht darin, es über optische Verzögerungsschleifen (*fiber delay line*, FDL) zu schicken. Eine solche Schleife ist im einfachsten Fall ein Lichtwellenleiter (LWL) mit einer bestimmten Länge. Damit können Daten zwar verzögert werden, eine echte Speicherung ist aber nicht möglich, da die Informationen nur zu einem bestimmten Zeitpunkt ausgelesen werden können und deren Reihenfolge auch nicht verändert werden kann (sequentieller Speicher).
In einer Verzögerungsleitung können – wie in jedem Lichtwellenleiter – mit Hilfe von WDM parallel mehrere Wellenlängen genutzt werden. Dies ist sinnvoll, denn dadurch können in einer FDL mehrere Bursts gleichzeitig gespeichert werden, was die Anzahl der in einem Knoten benötigten Verzögerungsleitungen drastisch senken kann. Aufgrund des Aufwandes, der für eine Wellenlängenänderung notwendig ist, besteht grundsätzlich das Ziel, einen Burst stets auf der gleichen Wellenlänge zu übertragen – auch im Puffer. Auch dies ist bei mehreren Wellenlängen pro Puffer möglich, sofern die geforderte Wellenlänge nicht anderweitig belegt ist.
In Abbildung 3.1 wird versucht, die wesentlichen Eigenschaften einer einzelnen optischen Verzögerungsschleife zu klassifizieren. Ebenfalls enthalten ist ein Überblick, wie diese Eigenschaften für die späteren Untersuchungen modelliert wurden. Für weitere Details zu dem Simulationstool sei auf Anhang A verwiesen.
Aus physikalischer Sicht ist eine FDL zunächst durch ihre Länge gekennzeichnet.

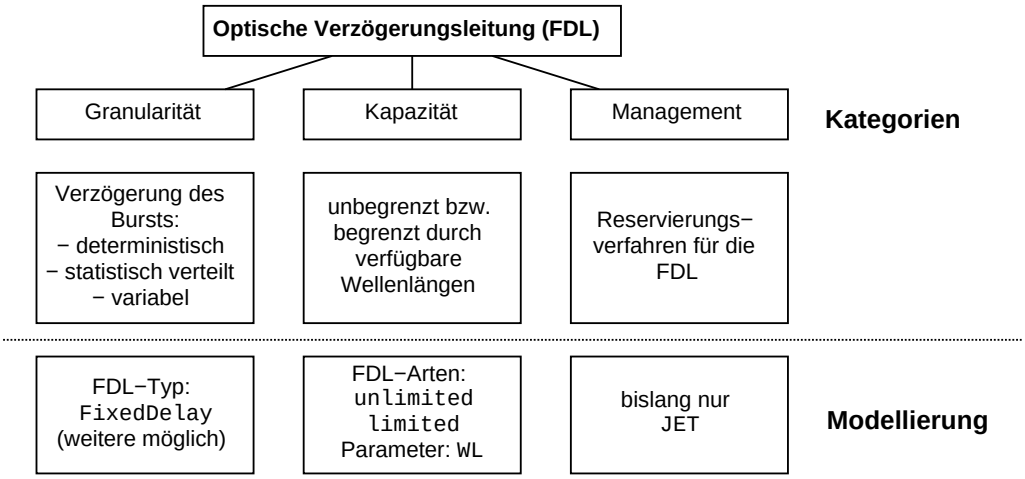

Abb. 3.1: Eigenschaften von optischen Verzögerungsleitungen

Für die Datenübertragung ist aber nur die durch sie hervorgerufene Verzögerung von Interesse, die – ohne Dispersion und unter gleichen äußeren Rahmenbedingungen – als konstant angenommen werden kann. Variable Verzögerungen sind für eine einzelne Leitung nur theoretisch denkbar, für eine Modellierung komplexer Pufferstrukturen kann es aber durchaus sinnvoll sein, derartige Leitungen anzunehmen. Von dieser Möglichkeit wird aber in den nachfolgenden Untersuchungen nicht Gebrauch gemacht.

Die Anzahl der verfügbaren Wellenlängen ist eine weitere charakteristische Kenngröße einer FDL. Technologisch ist diese Anzahl und damit die Kapazität stets begrenzt, diese Einschränkung kann aber bei einer hinreichend großen Anzahl irrelevant sein, so dass theoretisch auch eine Faser mit unbeschränkter Kapazität angenommen werden kann.

Die dritte in Abbildung 3.1 erwähnte Kategorie, nämlich das angewendete Reservierungsverfahren, scheint zunächst keine spezifische Eigenschaft einer Lichtwellenleiter-Schleife zu sein. Bei der Modellierung von optischen Puffermechanismen ist es aber sinnvoll, eine einzelne FDL als allgemeineres Konzept aufzufassen, d. h. als eine „blackbox“ mit einer bestimmten Verzögerung und einer definierten Kapazität. Dann ist auch der (hypothetische) Fall denkbar, dass in mehreren Verzögerungsleitungen unterschiedliche Reservierungsverfahren angewendet werden, so dass dieses für die einzelne eine charakteristische Eigenschaft sein kann.

3.3 Optische Puffer

Ein optischer Puffer muss nicht nur aus einer einzigen Verzögerungsleitung bestehen. Je nach Aufwand sind durch Kombination von Verzögerungsleitungen fast beliebig komplizierte Puffer möglich. Während eine FDL Bursts lediglich verzögert, wird unter dem Begriff „Puffer“ nachfolgend die Gesamtheit aller einer Vermittlungseinrichtung zugeordneten Verzögerungsmöglichkeiten verstanden. Aufgabe eines Puffers ist es, in Konfliktfällen Bursts so lange zwischenzuspeichern, bis eine benötigte Ressource (Switch oder Leitung) wieder frei ist. Abbildung 3.2 gibt einen Überblick über die gemeinsamen Eigenschaften eines derartigen Puffers. Diese werden dann in Unterkapiteln detaillierter betrachtet.

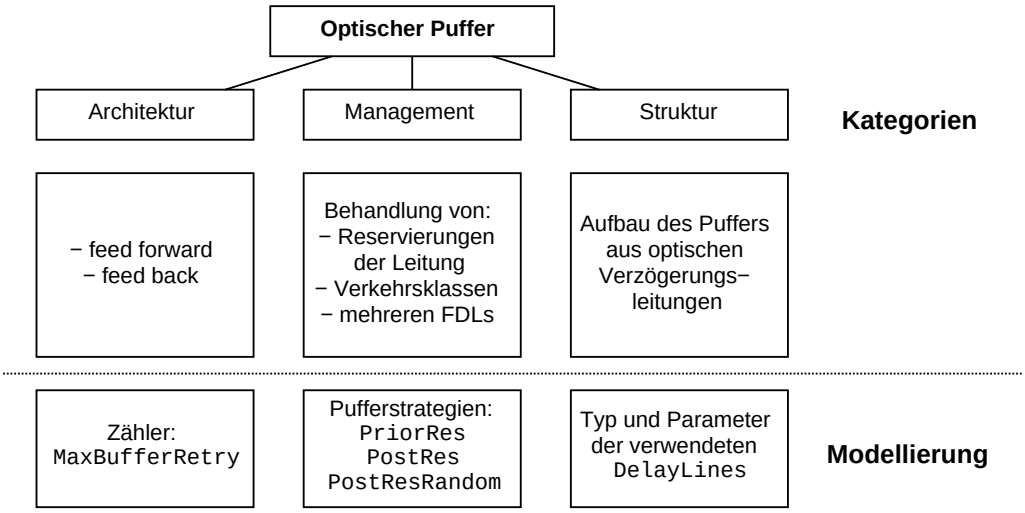

Abb. 3.2: Klassifizierung von optischen Puffern

3.3.1 Architektur und Struktur

Schon in Kapitel 2.3 wurde erläutet, dass Puffer in vielen Knoten-Architekturen integraler Bestandteil des Switch sind bzw. sein können. Schließlich stellen schon zwei verschieden lange LWL prinzipiell einen (einfachen) „Puffer“-Mechanismus dar. Eine eindeutige Trennung des Puffers von der reinen Vermittlungsfunktion eines Knotens ist daher auf technologischer Ebene oftmals schwierig.
Dennoch gibt es zwei grundsätzlich verschiedene Möglichkeiten, einen Puffer an die übrigen Komponenten eines Knotens anzubinden. Diese beiden Konzepte sind in Abbildung 3.3 dargestellt.
Der naheliegenste Ansatz zur Verwendung eines Puffers ist, Bursts um eine gewisse Dauer b zu verzögern, falls Ressourcen belegt sind und eine Behandlung da-

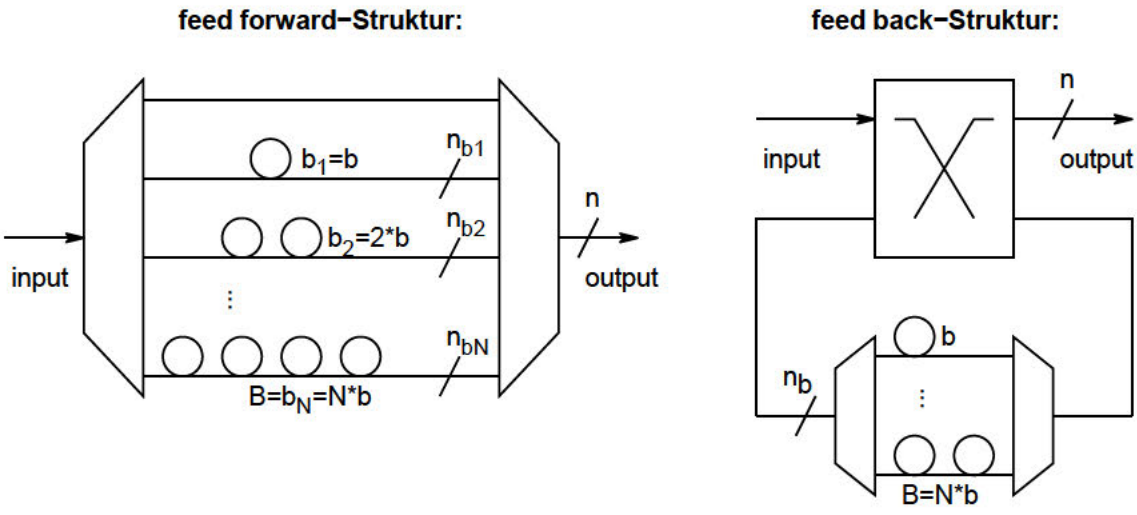

Abb. 3.3: Verschiedene Puffer-Architekturen

her momentan nicht möglich ist. Dies wird erreicht, indem Bursts wahlweise direkt übertragen oder durch eine FDL geleitet werden. Um eine größere Flexibilität zu erreichen, bietet es sich an, mehrere Verzögerungsleitungen unterschiedlicher Länge zur Verfügung zu stellen, so dass verschieden lange Verzögerungen b_i möglich sind. Dann erhält man die in Abbildung 3.3 links dargestellte Struktur, die in [7] als *feed forward*-Anordnung bezeichnet wird. In der Literatur sind aber auch noch andere Bezeichnungen für diese Anordnung üblich, beispielsweise *single pass* oder *travelling buffer*.

Eine FDL kann natürlich durch Verwendung von WDM auf n_{bi} Wellenlängen gleichzeitig Bursts puffern. Dazu können allerdings vor und/oder nach dem Puffer Wellenlängenkonverter notwendig sein. Die Länge der einzelnen Verzögerungsleitungen ist beliebig wählbar. Ein oft verwendeter Ansatz ist, diese als ganzzahlige Vielfache einer Verzögerung b zu dimensionieren („Treppe“):

$$b_i = i\ b \tag{3.1}$$

Bei insgesamt N FDLs ergibt sich dann eine maximal mögliche Verzögerungszeit B:

$$B = b_N = N\ b \tag{3.2}$$

Neben dieser speziellen Wahl sind aber auch andere FDL-Längen denkbar. Um die Durchlaufzeit durch den Puffer so gering wie möglich zu halten, ist es sinnvoll, Bursts bevorzugt durch kurze Verzögerungsleitungen zu schicken. Wenn im Auswahlalgorithmus die einzelnen Leitungen sequenziell abgesucht werden, dann sollte

ganz allgemein folgende Bedingung erfüllt sein:

$$b_i \leq b_{i+1} \quad \forall\ i \tag{3.3}$$

Alle in *feed forward*-Anordnung möglichen Strukturen haben gemein, dass ein Burst den Puffer maximal einmal durchlaufen kann.
Diese Einschränkung kann durch eine *feed back*-Architektur überwunden werden, die eine mehrmalige Pufferung erlaubt und bisweilen auch *recirculation buffer* oder *loop buffer* genannt wird. Diese in Abbildung 3.3 rechts skizzierte Architektur ist flexibler, wirft aber auch zusätzliche Probleme auf: Durch den mehrfachen Puffer- und Switch-Durchlauf tritt eine größere Dämpfung auf, die Verstärker wie z. B. *semiconductor optical amplifiers* (SOA) oder *Erbium-doped fiber amplifiers* (EDFA) notwendig macht. Durch diese tritt zusätzliches Rauschen (*amplified spontaneous emission*, ASE) auf. Diese technologischen Randbedingungen begrenzen die Anzahl an möglichen Pufferdurchläufen.
Für die Struktur bestehen die gleichen Freiheitsgrade wie bei einem *feed forward*-Puffer. Da sich in diesem Ansatz bereits durch die unterschiedliche Anzahl an möglichen Pufferdurchläufen unterschiedliche Verzögerungszeiten ergeben, besteht aber im Allgemeinen keine Notwendigkeit, aufwendige Puffer zu verwenden. In Veröffentlichungen wird deshalb meist vorgeschlagen, nur FDLs mit einer einheitlichen Länge einzusetzen.

feed forward	**feed back**
• ein Pufferdurchlauf	• mehrere Pufferdurchläufe
• meist ohne Verstärker	• meist Verstärker notwenig
• eher komplexe Pufferstrukturen	• eher einfache Pufferstrukturen
• Zeitpunkt des Austritts bekannt	• Zeitpunkt des Austritts variabel

Tabelle 3.1: Vergleich der Puffer-Architekturen

Eine wesentliche Charakteristik der *feed back*-Architektur, die später noch von Bedeutung sein wird, ist die Tatsache, dass zum Zeitpunkt des „Einschreibens“ in den Puffer noch nicht fest steht, wann ein Burst den Puffer endgültig verlassen wird – ganz im Unterschied zu *feed forward*. In Tabelle 3.1 sind die prinzipiellen Eigenschaften dieser beiden verschiedenen Architekturen nochmals gegenüber gestellt. Unter

Annahme eines intern blockierungsfreien und verzögerungslosen Switch lässt sich *feed forward* als ein Sonderfall von *feed back* auffassen, bei dem nur maximal ein Pufferdurchlauf zulässig ist. Diese Modellierung wird im Folgenden stets angenommen.

3.3.2 Puffermanagement

Eine sehr entscheidende Frage bei der Verwendung von Puffern ist, wie diese verwaltet werden und wie darauf zugegriffen wird. In Abbildung 3.4 wird ein sehr einfacher Fall betrachtet: eine Leitung mit einer Wellenlänge ($n = 1$) und ein aus einer Verzögerungsleitung bestehender Puffer.

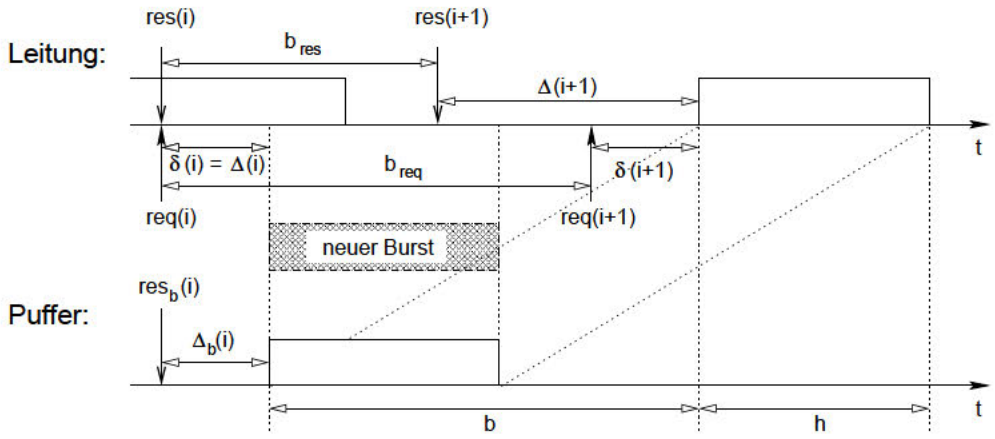

Abb. 3.4: Szenario eines Puffer-Vorgangs

Ein eintreffendes Kontrollpaket eines Bursts $req(i)$ mit dem Offset $\delta(i)$ fordert eine freie Leitung an $(res(i))$, kann diese aber nicht reservieren, da es zu einer Kollision mit einer vorherigen Belegung kommt. Eine Reservierung eines Zeitschlitzes im Puffer $(res_b(i))$ ist dagegen möglich. Wie in Abbildung 3.4 dargestellt, lässt sich der Puffervorgang durch 3 Verzögerungszeiten charakterisieren: Nach dem Eintreten in den Puffer wird das Datenpaket um die (in der Regel feste) Laufzeit b verzögert, nach der es wieder aus dem Puffer austritt. Um zu diesem Zeitpunkt eine freie Leitung zur Verfügung zu haben, muss diese durch eine Reservierung $res(i+1)$ belegt werden. Der Zeitpunkt dieser Reservierung kann variabel gewählt und durch eine Zeitverzögerung b_{res} gegenüber dem ersten Reservierungsversuch beschrieben werden. Insbesondere bei mehreren nacheinander durchlaufenen Knoten ist weiterhin die Zeitdauer b_{req} relevant, nach der das Kontrollpaket an den nachfolgenden Knoten weitergeleitet wird. Mit diesen Definitionen ergeben sich neben einem Offset δ

zwischen Kopf und Datenteil eines Bursts ein weiterer Offset Δ zwischen dem Reservierungsversuch und den Daten. Für den pufferlosen Fall bzw. das erste Eintreffen eines Bursts im Switch kann $\delta(i) = \Delta(i)$ angenommen werden.
Durch die äußeren Rahmenbedingungen sind die Parameter der Verzögerungsleitung Einschränkungen unterworfen: Der Zeitpunkt von $res(i+1)$ und $req(i+1)$ darf nicht nach dem Beginn der Datenübertragung liegen. Außerdem ist es sinnvoll, $b_{\text{res}} \leq b_{\text{req}}$ zu wählen, weil sonst der Fall auftreten könnte, dass nachfolgende Knoten eine Reservierung für einen Burst vornehmen, der aber gar nicht übertragen werden kann, da keine Leitung zur Verfügung steht. Daraus ergeben sich folgende sinnvolle Wertebereiche:

$$0 \leq b_{\text{req}} \leq \delta(i) + b \tag{3.4}$$

$$0 \leq b_{\text{res}} \leq b_{\text{req}} \tag{3.5}$$

Die unterschiedlichen Verzögerungen können den Offset zwischen Burst und Datenteil verändern. Für einen nachfolgenden Knoten hat dieser den folgenden Wert:

$$\delta(i+1) = \delta(i) + b - b_{\text{req}} \tag{3.6}$$

Eine Modifikation des Offsets durch die Pufferung würde dazu führen, dass neue Verkehrsklassen eingeführt (siehe Kapitel 4.7.1) und gepufferte Bursts im weiteren Netz entweder bevorzugt oder benachteiligt werden würden. Dieses Verhalten wird im Allgemeinen nicht erwünscht sein[1]. Unter dieser Voraussetzung ergibt sich aus Gleichung (3.6) die Beziehung $b_{\text{req}} = b$. Damit bleibt noch ein Freiheitsgrad bestehen: die Verzögerung b_{res}, für die gelten muss:

$$0 \leq b_{\text{res}} \leq b \tag{3.7}$$

Die nachfolgende Betrachtung konzentriert sich auf die beiden Extremfälle:

1. $b_{\text{res}} = 0$: Dieser Ansatz ist zunächst naheliegend. In dem Moment, in dem der Switch feststellt, dass ein Burst nicht direkt übertragen werden kann, versucht er festzustellen, ob eine Übertragung zu einem späteren Zeitpunkt (also nach der Verzögerung b) möglich ist. Falls ja, wird in der Reservierungsdatenbank

[1] Ein denkbarer Anwendungsfall wäre aber, den Offset einmal gepufferter Bursts zu erhöhen, um diese zu priorisieren und so die Wahrscheinlichkeit einer erneuten Pufferung zu vermindern. Außerdem kann auf diese Weise der Offset erhöht werden, falls er eine kritische Grenze unterschritten hat.

ein passender Zeitschlitz reserviert, der darauffolgenden Bursts nicht mehr zur Verfügung steht, und der Burst wird gepuffert. Falls keine freie Wellenlänge gefunden wird, wird der Burst sofort verworfen. Dieses Verfahren wird nachfolgend „PriorRes“ (zuerst Reservierung, dann Pufferung) bezeichnet.

2. Eine alternative Lösung besteht darin, einen blockierten Burst zunächst unabhängig von der Frage, ob eine spätere Reservierung der Leitung erfolgen kann, in den Puffer zu schicken. Die Belegung der Leitung wird erst nach einer Verzögerung $b_{\mathrm{res}} = b$ betrachtet. Damit besteht für die Reservierung keinerlei Unterschied zwischen Bursts, die neu im Knoten ankommen, und Bursts, die sich im Puffer befinden. Weil bei dieser Strategie der Reservierungsvorgang erst nach einer gewissen Zeitdauer erfolgt, erhält sie den Namen „PostRes“.

Damit die PriorRes-Strategie angewendet werden kann, muss schon vor dem Eintritt in den Puffer der Zeitpunkt der Burstübertragung bekannt sein. Dies ist bei einem *feed forward*-Puffer gewährleistet, nicht aber bei einer *feed back*-Anordnung mit mehreren zugelassenen Durchläufen. Eine Verwendung von PriorRes in der oben definierten Form mit *feed back*-Puffern ist daher schwierig, und es passt aufgrund seiner spezifischen Eigenschaften eher zu *feed forward*-Strukturen[2]. Für PostRes dagegen gilt diese Einschränkung nicht: Schlägt die Reservierung $res(i+1)$ fehl, so kann der Burst erneut gepuffert werden ($res_{\mathrm{b}}(i+1)$) und ein Zugriff zu einem späteren Zeitpunkt wiederholt werden ($res(i+2)$). Diese Prozedur kann solange wiederholt werden, solange Pufferspeicher zur Verfügung steht und die maximal mögliche Anzahl an Pufferdurchläufen nicht überschritten wird. Diese Vorgänge werden in Abbildung 3.5 nochmals veranschaulicht.
Bei Puffern, die aus mehreren Verzögerungsleitungen bestehen, muss eine Entscheidung getroffen werden, in welchen Puffer ein blockierter Burst eingeschrieben wird. Für Optical Burst Switching finden sich in der Literatur dazu bisher keine konkreten Vorschläge.
Diese Frage ist eng mit der Reservierungsstrategie verknüpft, da bei der Auswahl z. B. die schon auf einer Leitung vorliegenden Reservierungen berücksichtigt werden sollten. Aus den Reservierungsstrategien PriorRes bzw. PostRes kann direkt eine einfache FDL-Auswahl-Strategie abgeleitet werden, wenn ein einstufiger Puffer verwendet wird, d. h. eine Reihe von (unterschiedlich langen) Verzögerungsleitungen („Treppe“) vorliegt:

[2]Diese Beschränkung könnte durch Mischformen von PostRes und PriorRes überwunden werden.

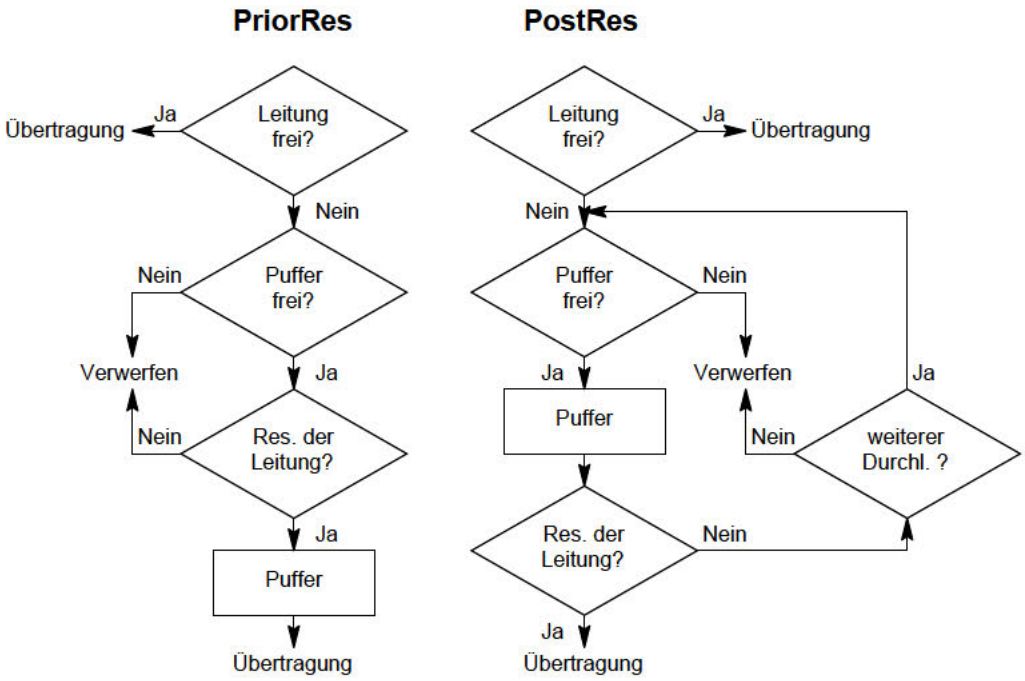

Abb. 3.5: Datenflussdiagramm für PriorRes und PostRes

1. Das Reservierungsverfahren PriorRes untersucht, ob der Burst nach Austritt aus dem Puffer erfolgreich übertragen werden kann. Bei mehreren Verzögerungsleitungen werden diese sukzessiv abgesucht, bis eine FDL mit einer Länge gefunden ist, die eine Reservierung ermöglicht. Falls keine entsprechende Leitung gefunden wird, wird der Burst verworfen.

2. Analog ist es auch möglich, bei PostRes mehrere Verzögerungsleitungen abzusuchen. Da bei PostRes aber völlig unabhängig von der Leitungsbelegung und der Länge der FDLs gepuffert wird, wird solange die erste FDL gefüllt, bis dort kein Platz mehr verfügbar ist, und dann mit der zweiten FDL fortgefahren. Hier entscheidet also alleine die Auslastung der FDL, welcher Verzögerungsschleife ausgesucht wird.

Die reine PostRes-Strategie nutzt damit die Eigenschaften eines Puffers mit mehreren FDLs kaum aus. Wird ein solcher Puffer in Zusammenhang mit PostRes verwendet, wären „intelligentere“ Auswahlstrategien durchaus sinnvoll: Eine sehr einfache Modifikation von PostRes für diesen Zweck stellt die statistische Auswahl einer der verfügbaren FDL dar. Nachfolgend wird unter der Bezeichnung „PostRes-Random“ der Ansatz verfolgt, dass jede vorhandene Verzögerungsleitung mit der gleichen Wahrscheinlichkeit ausgewählt wird. Ist die FDL belegt, so werden keine weiteren getestet. Alternativ wäre auch ein Ausweichen auf andere FDLs möglich.

Neben diesen direkt von den Reservierungsverfahren PriorRes und PostRes abgeleiteten Auswahlmechanismen sind sicher auch komplexere Algorithmen möglich, die evtl. noch weitere Parameter berücksichtigen. In der Regel setzt dies aber eine genaue Kenntnis der verwendeten Pufferstruktur voraus und geht daher über diese grundlegenden Untersuchungen hinaus.
Dies gilt auch für andere Möglichkeiten zum Einsatz von Puffern, die bislang nicht erwähnt wurden: Man könnte nur bestimme Bursts puffern, z. B. abhängig von deren Länge oder Verkehrsklasse, oder in Abhängigkeit von bestimmten Kriterien verschiedene Puffer ansteuern u. ä. Derartige Vorgehensweisen würden allerdings die Komplexität der Puffer deutlich erhöhen, ohne dass signifikante Verbesserungen zu erwarten wären.

3.4 Zusammenfassung

Die bisherigen Betrachtungen haben sich auf eine einstufige Pufferarchitektur (*single stage*) in Verbindung mit einer zentralen Koppelmatrix konzentriert. Selbstverständlich ist auch der allgemeinere Fall eines mehrstufigen *multi stage*-Puffers möglich, der noch flexibler ist. Derartige Strukturen werden z. B. in [20] behandelt und dort als sogenannte „hybride“ Puffer modelliert. In diesen Szenarien ist aber eine Trennung der Pufferfunktionalität von den übrigen Komponenten des Switches schwierig und die Anzahl an Freiheitsgraden im System noch wesentlich größer. Da Optical Burst Switching – wie schon erwähnt – in den Transitknoten auch ohne Puffer auskommen kann, sind derartige komplizierte Pufferkonzepte wahrscheinlich nicht notwendig; auf deren Modellierung und Untersuchung wurde somit verzichtet. Generell gibt es bei dem Entwurf und der Dimensionierung von Switch-Architekturen und Puffern relativ viele Freiheitsgrade, so dass eine Reduktion auf die wesentlichen Prinzipien und Mechanismen notwendig ist.
Ein Nebeneffekt von Puffern ist, dass dadurch die Reihenfolge von Bursts/Paketen verändert werden kann. Dieses Verhalten ist meist unerwünscht, kann aber kaum verhindert werden. Im Fall von OBS ist dies aber unkritisch, da aufgrund von möglichen Burstverlusten ohnehin eine Datensicherungsschicht vorgesehen werden muss, die auch die Reihenfolgeintegrität gewährleisten sollte.

4 Untersuchung optischer Puffermechanismen

4.1 Übersicht

In diesem Kapitel wird untersucht, welche Auswirkungen der Einsatz von Puffern in Optical Burst Switching-Netzen hat. Da der Puffermechanismus innerhalb eines Switch völlig transparent funktioniert, reicht es aus, einen einzigen Knoten zu betrachten.

4.2 Modell und Parameter

Ein OBS-Knoten ist durch zahlreiche Parameter gekennzeichnet. Eine Reihe dieser Kenngrößen ist direkt oder indirekt von der verwendeten Technologie abhängig, beispielsweise die Anzahl an Wellenlängen n, die auf den Übertragungsleitungen eingesetzt wird. Die verwendete Last ρ bzw. das Angebot $A = n\ \rho$ ist dagegen frei wählbar. Die Länge der Bursts ist stark von dem Verfahren zur Verkehrsaggregation abhängig. Für die mittlere Burstlänge wird wie in [5] angenommen, dass sie $E[L] = 100$ kBit $= 12,5$ kByte beträgt. Bei einer Datenübertragungsrate von $r = 10$ GBit/s entspricht dies einer Übertragungsdauer von $h_0 = E[h] = 10\ \mu$s. Sofern nicht explizit anders angegeben, wird für die Burstlänge eine negativ-exponenzielle Verteilung angenommen. In allen durchgeführten Simulationen wurden Poisson-Ankünfte verwendet. Diese Annahme ist gemäß [3, 6] für viele verschiedene Szenarien zu rechtfertigen. In den Simlationen werden immer $g = 100$ Verkehrsgeneratoren verwendet. Dann beträgt die Zwischenankunftszeit (*inter arrival time*, IAT) mit den bislang eingeführten Bezeichnungen:

$$E[t_{\mathrm{IAT}}] = \frac{L\ g}{\rho\ n\ r} \tag{4.1}$$

Für die Bewertung eines OBS-Systems sind die Burstverlustwahrscheinlichkeit P_{Loss} und die Ende-zu-Ende-Verzögerung der Bursts relevant. Die Verzögerung eines Bursts im Fall einer Übertragung ohne Puffer wird durch die optische Ausbreitungsgeschwindigkeit und durch die Dauer der Speicherung im Randknoten beeinflusst. Da die Ausbreitungsgeschwindigkeit konstant ist, ist eine Beeinflussung nur durch das im *ingress*-Router angewendete Aggregationsverfahren möglich. Unter der Annahme, dass der Switch intern keine Laufzeiten aufweist und nur ein Referenzpfad im Netz betrachtet wird, können Laufzeit-Verzögerungen generell vernachlässigt werden.

Die Einführung von Puffern erlaubt es, die Burstverlustwahrscheinlichkeit zu reduzieren, gleichzeitig wird aber die Ende-zu-Ende Verzögerung für die gepufferten Bursts erhöht. Werden aufwendigere und größere Puffer verwendet, kann – wie sich noch herausstellen wird – in der Regel die Verlustrate weiter verbessert werden, allerdings meist auf Kosten einer größeren mittleren Durchlaufzeit. Bei der Dimensionierung von Puffern ist also ein Kompromiss zwischen einer möglichst geringen Verlustwahrscheinlichkeit und akzeptablen Verzögerungswerten zu finden.
In einem OBS-Knoten ohne Puffer tritt ein Burstverlust auf, falls keine geeignete Ausgangsleitung zur Verfügung steht. Falls ein Switch verwendet wird, der nicht intern blockierungsfrei ist, können auch Blockierungen im Switch auftreten[3]. Sind in einem OBS-Knoten Puffer vorhanden, dann kann ein Burst, der nicht direkt vermittelt werden kann, zwischengespeichert werden. Ein Verwerfen dieses Burst kann dann zwei mögliche Ursachen haben:

1. Eine Pufferung ist unmöglich, weil kein Pufferplatz frei ist und dessen Reservierung fehlschlägt.

2. Es steht zwar ein Puffer zur Verfügung, aber die Ausgangsleitung ist auch zu einem späteren Zeitpunkt belegt.

Diese Phänomene lassen sich durch mehrere bedingte Wahrscheinlichkeiten beschreiben: Die Wahrscheinlichkeit dafür, dass eine direkte Übertragung eines im Switch ankommenden Bursts fehlschlägt, ist die *Blockierwahrscheinlichkeit* und wird als P_{Block} bezeichnet. P_{Buffer} ist die bedingte Wahrscheinlichkeit, dass ein blockierter Burst gepuffert werden kann (*Pufferwahrscheinlichkeit*). Aufgrund der unterschiedlichen Reservierungsstrategien muss hier zwischen PostRes und PriorRes unterschieden werden: Bei PostRes gilt $P_{\text{Buffer}} = 1$, wenn ein unbeschränkter Puffer verwendet wird, sonst hängt der Wert stark von der Dimensionierung (Kapazität) des Puffers ab. Da bei PriorRes nur Bursts gepuffert werden, bei denen eine Reservierung der Ausgangsleitung erfolgreich war, hängt hier P_{Buffer} auch von dem Erfolg des Reservierungsprozesses ab. Die *Wiedereintrittswahrscheinlichkeit* P_{Reentry} bestimmt die Wahrscheinlichkeit, mit der es einem erfolgreich gepufferten Burst gelingt, Leitungskapazität zu reservieren. Für PriorRes gilt definitionsgemäß $P_{\text{Reentry}} = 1$. Somit gibt $P_{\text{Block}}\ P_{\text{Buffer}}\ P_{\text{Reentry}}$ den Anteil der Bursts an, der gepuffert und anschließend übertragen wird. Die *Gesamtverlustwahrscheinlichkeit* ist dann:

$$P_{\text{Loss}} = P_{\text{Block}}\ (1 - P_{\text{Buffer}}\ P_{\text{Reentry}}) \tag{4.2}$$

[3]Die Option *deflection routing* sei hier vernachlässigt.

Diese Formel gilt nur für einen einmaligen Pufferdurchlauf. Wenn Q die maximale Anzahl an zulässigen Pufferdurchläufen ist, dann lässt sich diese Formel für den Fall $Q > 1$ verallgemeinern:

$$P_{\text{Loss}} = P_{\text{Block}} \left(1 - \sum_{i=1}^{Q} P_{\text{Buffer}}^{(i)} \, P_{\text{Reentry}}^{(i)} \prod_{j=1}^{i-1} P_{\text{Buffer}}^{(j)} \left(1 - P_{\text{Reentry}}^{(j)}\right)\right) \tag{4.3}$$

$P_{\text{Buffer}}^{(j)}$ bzw. $P_{\text{Reentry}}^{(j)}$ geben die Puffer- bzw. Wiedereintrittswahrscheinlichkeit für den j-ten Pufferdurchlauf an ($P_{\text{Buffer}}^{(Q+1)} = 0$). Aus obiger Formel folgt auch mathematisch, dass das Zulassen mehrerer Pufferdurchläufe ($Q > 1$) bei PriorRes ($P_{\text{Reentry}} = 1$) keinen Effekt hat.

Ohne besondere Mechanismen sind alle Bursts im Puffer gleichberechtigt, d.h. es gilt $P_{\text{Buffer}}^{(j)} = P_{\text{Buffer}}$ und $P_{\text{Reentry}}^{(j)} = P_{\text{Reentry}} \quad \forall\, j \in [1; Q]$. Dann lässt sich Gleichung (4.3) umformen:

$$P_{\text{Loss}} = P_{\text{Block}} \left(1 - P_{\text{Buffer}} \, P_{\text{Reentry}} \, \frac{1 - \left(P_{\text{Buffer}} \left(1 - P_{\text{Reentry}}\right)\right)^{Q}}{1 - P_{\text{Buffer}} \left(1 - P_{\text{Reentry}}\right)}\right) \tag{4.4}$$

Unter der Voraussetzung, dass wie bei PostRes auch nicht zwischen ankommenden und gepuffertern Bursts unterschieden wird, wären durch $P_{\text{Block}} = 1 - P_{\text{Reentry}}$ weitere Vereinfachungen möglich.

Die Wahrscheinlichkeiten P_{Block}, P_{Buffer} und P_{Reenty} hängen von der Anzahl N an verfügbaren Verzögerungsleitungen, deren Verzögerung b und deren jeweiliger Kapazität ab, die durch die Anzahl an Wellenlängen n_{b} gegeben ist. Mit Ausnahme der im nächsten Abschnitt vorgestellten groben Abschätzung sind in der Literatur keine Ansätze zur Berechnung dieser Wahrscheinlichkeiten zu finden. Im Rahmen dieser Arbeit wurde daher der Versuch unternommen, diese Zusammenhänge in ein analytisches Modell umzusetzen. Prinzipiell besteht insbesondere bei PriorRes kein großer Unterschied zwischen gepufferten Bursts und Bursts, die einer höherprioren Verkehrsklasse angehören. Die diffenzierte Behandlung von Verkehrsklassen wird z. B. in [4, 5] eingehend analytisch untersucht. Eine Übertragung der dortigen Vorgehensweise auf Puffermechanismen ist allerdings nicht gelungen, so dass auf eine analytische Beschreibung hier verzichtet werden muss.

Für den einfachsten Fall, einem Puffer bestehend aus einer FDL der Länge b, lässt sich die mittlere Durchlaufzeit $\mathrm{E}[T_Q]$ (*queueing time*) folgendermaßen bestimmen[4]:

$$\mathrm{E}[T_Q] = \frac{P_{\text{Block}} \, P_{\text{Buffer}} \, P_{\text{Reentry}}}{1 - P_{\text{Block}} \left(1 - P_{\text{Buffer}} \, P_{\text{Reentry}}\right)} \, b \tag{4.5}$$

[4] Vorausgesetzt, alle anderen Laufzeiten im Switch sind gegenüber b vernachlässigbar.

Bei aufwendigeren Puffern ist eine derartige Berechnung der Transferzeit durch den Knoten schwierig.

4.3 Abschätzung

In [4, 5] wird gezeigt, dass sich ein OBS-Knoten ohne Puffer bei Markoffschen Ankünften als M/G/n-Verlustsystem behandeln lässt. Die Burstverlustwahrscheinlichkeit lässt sich dafür aus der bekannten Erlang-Verlustformel [8] berechnen:

$$P_{\text{Loss,discard}} = B(A,n) = \frac{\frac{A^n}{n!}}{\sum_{i=0}^{n} \frac{A^i}{i!}} \tag{4.6}$$

Zur Untersuchung der Frage, wie weit die Burst-Verlustwahrscheinlichkeit durch Pufferung gesenkt werden kann, wird in [16] und [20] vorgeschlagen, ein M/M/n/D-Modell anzunehmen, wobei $D = n + n_{\text{b}} * N$ die maximal mögliche Anzahl an Bursts im Switch ist (sowohl direkt übertragen als auch gepuffert):

$$P_{\text{Loss,limit}} = B(A,n,D) = \frac{A^D}{n^{D-n} n!}\, p_0 \tag{4.7}$$

mit

$$p_0 = \left(\sum_{i=0}^{n-1} \frac{A^i}{i!} + \sum_{i=n}^{D} \frac{A^i}{n^{i-n} n!} \right)^{-1} \tag{4.8}$$

Für die meisten Fälle stellt diese Näherung eine untere Grenze für die Verlustwahrscheinlichkeit dar. In Tabelle 4.1 sind für den Fall von $n = 64$ Wellenlängen und einem verhältnismäßig kleinen Puffer, bestehend aus einer Verzögerungsleitung ($N = 1$) mit $n_{\text{b}} = 8$ Wellenlängen, diese beiden Formeln gegenübergestellt. Für relativ niedrige Auslastung werden Burstverlustwahrscheinlichkeiten erreicht, die für Kommunikationsnetze „akzeptabel“ sind, und durch Verwendung von Puffern ist eine Verbesserung um mehrere Größenordnungen zu erwarten. Bei hoher Last steigt die Burstverlustwahrscheinlichkeit sehr stark an, und das Puffern zeigt geringere Auswirkungen. Dies zeigt, dass für die Dimensionierung von Puffern insbesondere hohe Lastwerte entscheidend sind.

Für genauere Untersuchungen von Puffermechanismen sind Simulationen notwendig. Die in den nächsten Abschnitten präsentierten Ergebnisse basieren – sofern nicht anders angegeben – stets auf einem Szenario mit $n = 8$ Wellenlängen und einer relativ hohen Last ($\rho = 0.8$). Mit diesen Parametern gelingt es, in relativ kurzen Simulationszeiten eine akzeptable Genauigkeit zu erreichen. Tatsächlich verfügen WDM-Netze über wesentlich mehr Wellenlängen und werden auch bei einer geringeren Last betrieben. Dies ist aber unkritisch, da auch für die nachfolgenden Untersuchungen

Last ρ	0.4	0.6	0.8
$P_{\text{Loss,discard}}$	8.054e-11	4.139e-5	0.01174
$P_{\text{Loss,limit}}$	5.278e-14	6.952e-7	0.001895
$P_{\text{Loss,discard}}/P_{\text{Loss,limit}}$	1500	60	6

Tabelle 4.1: berechnete Burstverlustwahrscheinlichkeiten ohne/mit Pufferung ($n = 64$, $n_{\text{b}} = 8$)

die Feststellung aus [6] gilt, dass der Wertebereich von Größen bei anderen Vorgaben über etliche Größenordnungen variieren kann, die Effekte und der prinzipielle Verlauf der Kurven aber erhalten bleiben.

4.4 Einmaliger Pufferdurchlauf

4.4.1 Burstverlustwahrscheinlichkeit

Um die prinzipiellen Auswirkungen und Effekte von Puffermechanismen festzustellen, wird zunächst ein sehr einfacher Fall betrachtet: Zur Behandlung von Zugriffskonflikten wird im Switch eine einzige Verzögerungsleitung ($N = 1$) in *feed forward*-Anordnung vorgesehen. Deren Speicherkapazität wird als unbeschränkt angenommen, d. h. sie kann beliebig viele Bursts aufnehmen.
Das Diagramm in Abbildung 4.1 zeigt die Burstverlustwahrscheinlichkeiten, die sich bei $n = 8$ und variabler Last ergeben. Als Bezug stellt die mit „Discard“ gekennzeichete Kurve den pufferlosen Fall dar, der – mit den gleichen Parametern – schon in [5] untersucht wurde. Für den Fall mit Puffern wird sowohl das PostRes- wie auch das PriorRes-Verfahren verwendet, und zwar bei zwei unterschiedlichen Längen der Verzögerungsleitung. Zur Schreibvereinfachung wird in manchen der folgenden Diagramme eine Normierung eingeführt: Fehlt bei der Länge der FDL bzw. der durch sie verursachten Verzögerung b eine absolute Größenangabe, so ist sie stets auf die mittlere Burstübertragungsdauer $h_0 = 10\,\mu$s bezogen. h_0 entspricht ungefähr einer Lichtwellenleiterlänge von 2 km (bei einer Ausbreitungsgeschwindigkeit von $2 * 10^8$ m/s).
Sowohl ohne wie auch mit Puffern nehmen die Verluste mit zunehmender Last stark zu. Die Verlustwahrscheinlichkeit wird aber durch die Pufferung signifikant reduziert. Auch zu erkennen ist, dass der in Kapitel 4.3 behandelte Analyse-Ansatz nur eine sehr grobe Abschätzung liefert und die tatsächlichen Verlustwahrscheinlichkeiten teilweise deutlich größer sind. Diese Abschätzung liefert hier also nur eine grobe

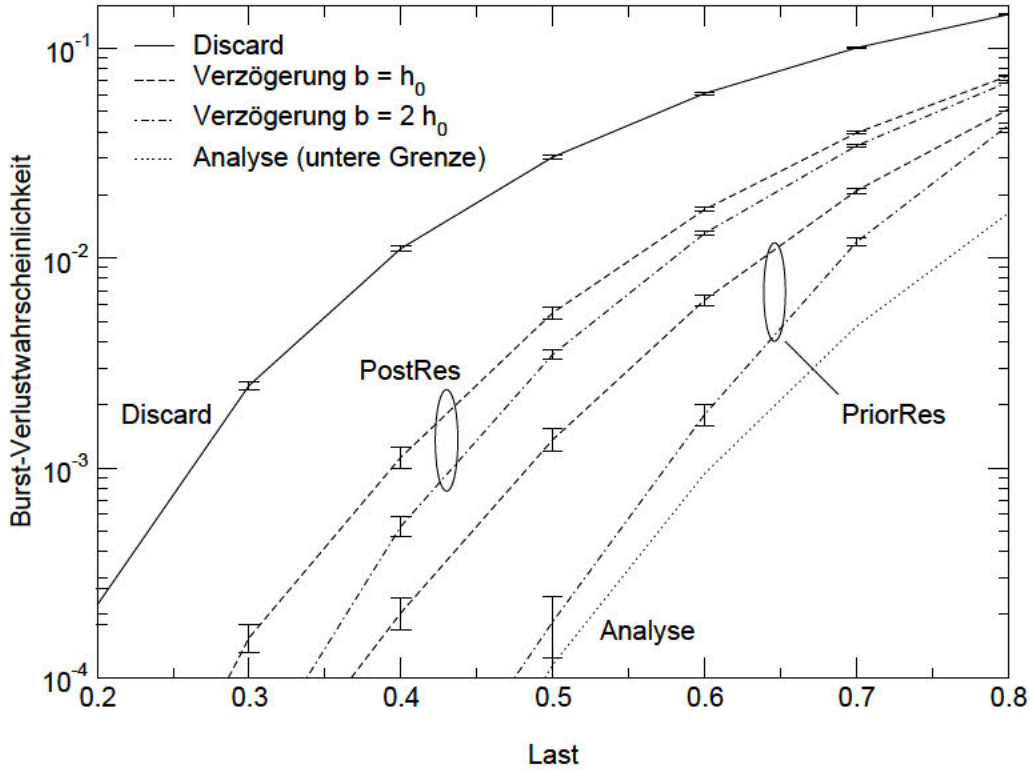

Abb. 4.1: Burstverlustwahrscheinlichkeiten in Abhängigkeit der Last

untere Grenze.

Weiterhin fällt auf, dass die Länge der Verzögerungsleitung durchaus eine Rolle spielt. In Abbildung 4.2 sind daher die Verlustwahrscheinlichkeiten über der Verzögerung b aufgetragen.

Wird der Switch im PostRes-Verfahren betrieben, so ist die Burstverlustwahrscheinlichkeit fast unabängig von der Länge der Verzögerungsleitung. Bei PostRes besteht der Puffermechanismus darin, einen Burst, bei dem eine erste, direkte Reservierung nicht möglich ist, zu verzögern und unter gleichen Bedingungen später einen zweiten Versuch zu starten. Die Dauer b spielt dabei offensichtlich keine wesentliche Rolle, solange sie groß genug gewählt ist, dass eine kurzzeitige Verstopfung aller Ausgangsleitungen mit großer Wahrscheinlichkeit abgeklungen ist. Für $b \geq h_0$ ist dies ziemlich sicher der Fall. Ist die FDL allerdings zu kurz, dann kann es sein, dass eine Überlastsituation auch nach einer Verzögerung b noch vorhanden ist. Dies verursacht den leichten Anstieg in der Verlustwahrscheinlichkeit für kleine b.

Bei PriorRes hängt die Burstverlustwahrscheinlichkeit dagegen signifikant von der FDL-Länge ab: Je länger die Verzögerungsleitung gewählt wird, desto weniger Bursts werden verworfen. Bei $b \geq 3\,h_0$ wird die durch die Abschätzung vorgegebene untere

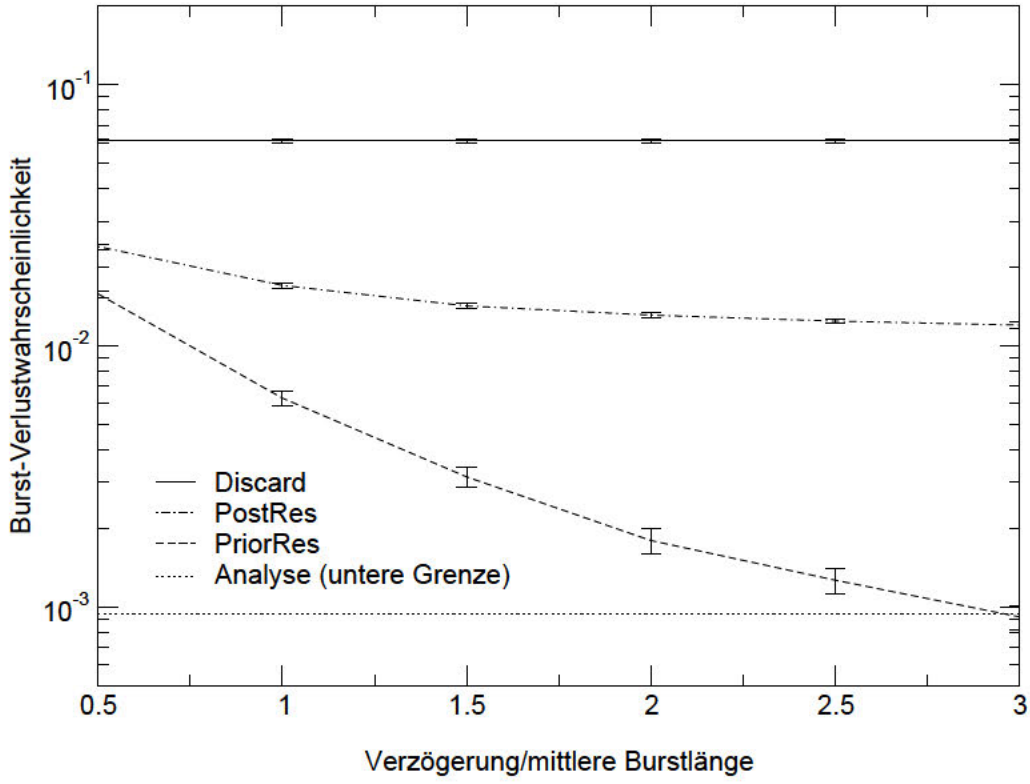

Abb. 4.2: Verlustwahrscheinlichkeiten für verschiedene Verzögerungen (Last 0.6)

Grenze erreicht bzw. sogar leicht unterschritten. Eine weitere Verlängerung würde noch eine weitere Absenkung bewirken, allerdings wird langsam eine Sättigung eintreten. Diese Abhängigkeit kann folgendermaßen erklärt werden: Bei PriorRes reserviert ein Burst beim Eintritt in den Puffer eine Ausgangsleitung. Dieser Zeitschlitz liegt – je nach Dauer der Verzögerung – relativ weit in der Zukunft. Je weiter man vorausreserviert, desto größer ist die Wahrscheinlichkeit, einen passenden Zeitschlitz zu finden. Dieses ist genau das Prinzip, das auch zur Trennung von Verkehrsklassen ausgenutzt wird. Für $b \gg h_0$ ist ziemlich sicher, dass die Übertragung von Bursts, die zum Reservierungszeitpunkt gerade die Wellenlängen belegen und eine Blockierung hervorrufen, beendet ist und Wellenlängen zur Verfügung stehen. Blockierte Bursts werden also durch die Tatsache, dass sie in die Zukunft reservieren, priorisiert. Die Wahrscheinlichkeit, dass eine derartige Reservierung und damit der Wiedereintritt funktioniert, ist sehr hoch, so dass die Verlustwahrscheinlichkeit absinkt.
Der Effekt, dass für $b \gg h_0$ die Simulation die durch die Analyse vorgegebene untere Grenze unterschreitet, wurde schon in [20] beobachtet und erklärt.
Bisher wurde eine Last von 0.6 angenommen. Wird wie in Abbildung 4.3 die Last auf 0.8 erhöht, so sind die prinzipiellen Kurvenverläufe weiterhin ähnlich. Bei dieser

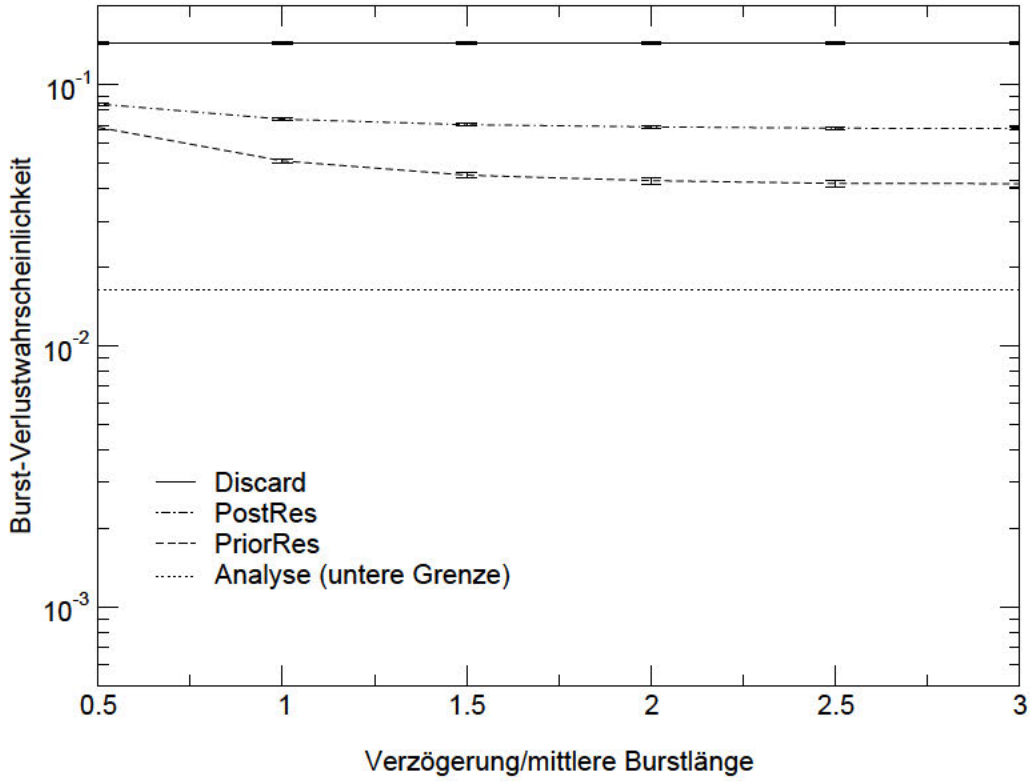

Abb. 4.3: Verlustwahrscheinlichkeiten für verschiedene Verzögerungen (Last 0.8)

relativ hohen Auslastung wird auch bei PriorRes schon bei kurzen Verzögerungen, ungefähr bei $b = 2\,h_0$, eine Sättigung erreicht, so dass darüber hinausgehende Werte kaum einen Effekt haben. Eine Verzögerungsleitung, die doppelt so lang wie die mittlere Burstlänge ist, stellt damit einen guten Kompromiss zwischen einerseits einer niedrigen Verlustwahrscheinlichkeit und andererseits einer noch relativ akzeptablen Verzögerung dar. Daher wird diese Dimensionierung nachfolgend angenommen, wenn nicht explizit etwas anderes angegeben ist.

Die Gesamtverlustwahrscheinlichkeit wird durch die zu Beginn dieses Kapitels eingeführten Wahrscheinlichkeiten bestimmt, die in Abbildung 4.4 eingetragen sind. Sowohl bei PostRes als auch bei PriorRes steigt durch die Verwendung von Puffern die Blockierwahrscheinlichkeit, d. h. die Wahrscheinlichkeit, dass eine direkte Leitungsreservierung zunächst fehlschlägt. Dies hängt offensichtlich damit zusammen, dass durch die Pufferung mehr Bursts übertragen werden können und damit die Belegung einer Wellenlänge zunimmt. Die Wiedereintrittswahrscheinlichkeit liegt bei PriorRes definitionsgemäß bei genau 1, bei PostRes ist sie signifikant kleiner. Dennoch beträgt sie auch hier mehr als 50 %, d. h. ein großer Anteil der gepufferten Bursts kann tatsächlich übertragen werden.

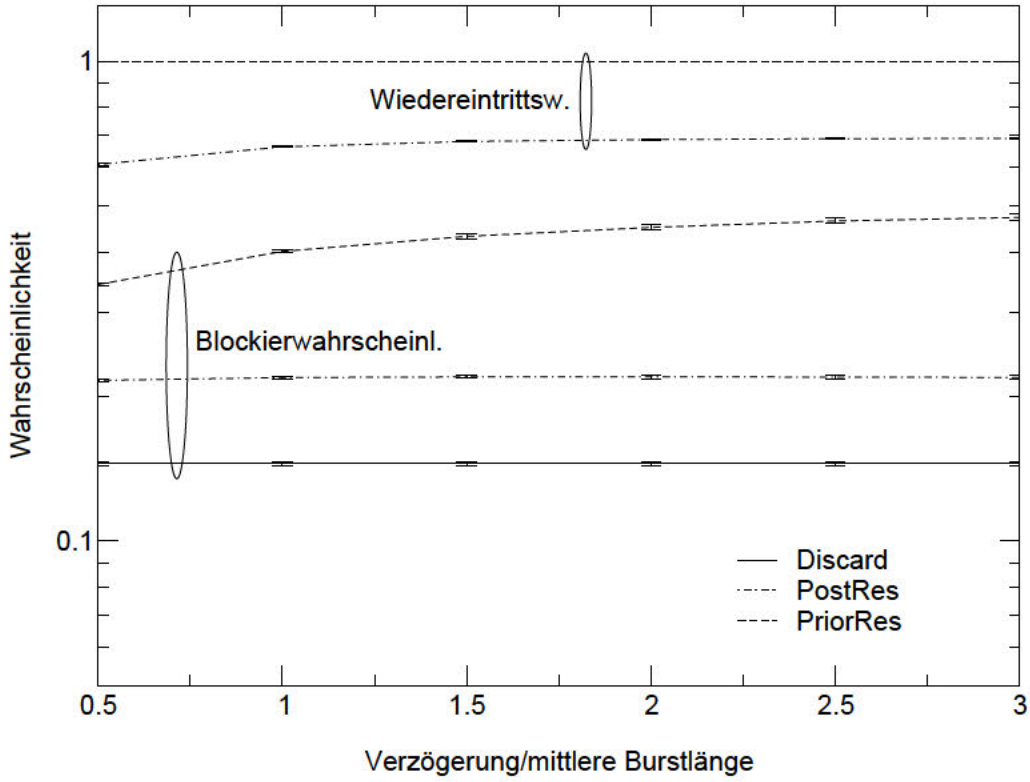

Abb. 4.4: Blockier- und Wiedereintrittswahrscheinlichkeiten (Last 0.8)

4.4.2 Durchlaufzeit

Zum Vergleich der beiden Verfahren werden in Abbildung 4.5 die Bursts betrachtet, die tatsächlich am Ausgang des Knotens ankommen, und der Anteil der gepufferten Bursts gebildet.

Bei niedriger Last treten nur sehr wenige Reservierungskonflikte auf und der Anteil an gepufferten Bursts ist verschwindend gering. Bei hoher Last gilt dies allerdings nicht mehr: Bei PostRes steigt der Anteil bis über 10 %, wobei hier ja nicht alle gepufferten Bursts am Ausgang ankommen. Bei PriorRes durchlaufen sogar fast die Hälfte der Bursts den Puffer. Der Puffer stellt dann nicht mehr nur eine Ausweichmöglichkeit dar, sondern wird zu einem regulären Weg für viele Bursts. Diese Tatsache ist auch festzustellen, wenn die mittlere Durchlaufzeit gemessen wird, die beim Durchlauf durch den Knoten entsteht (Abbildung 4.6).

Es ist offensichtlich, dass die mittlere Durchlaufzeit mit zunehmender Länge der Verzögerungsleitung zunehmen muss, weil der Teil der Bursts, der gepuffert wird, immer länger verzögert wird. Die Tatsache, dass dieser Anteil bei PriorRes größer ist, führt zu erheblich größeren mittleren Verzögerungen.

Ebenfalls beobachten kann man diesen Unterschied zwischen PriorRes und PostRes

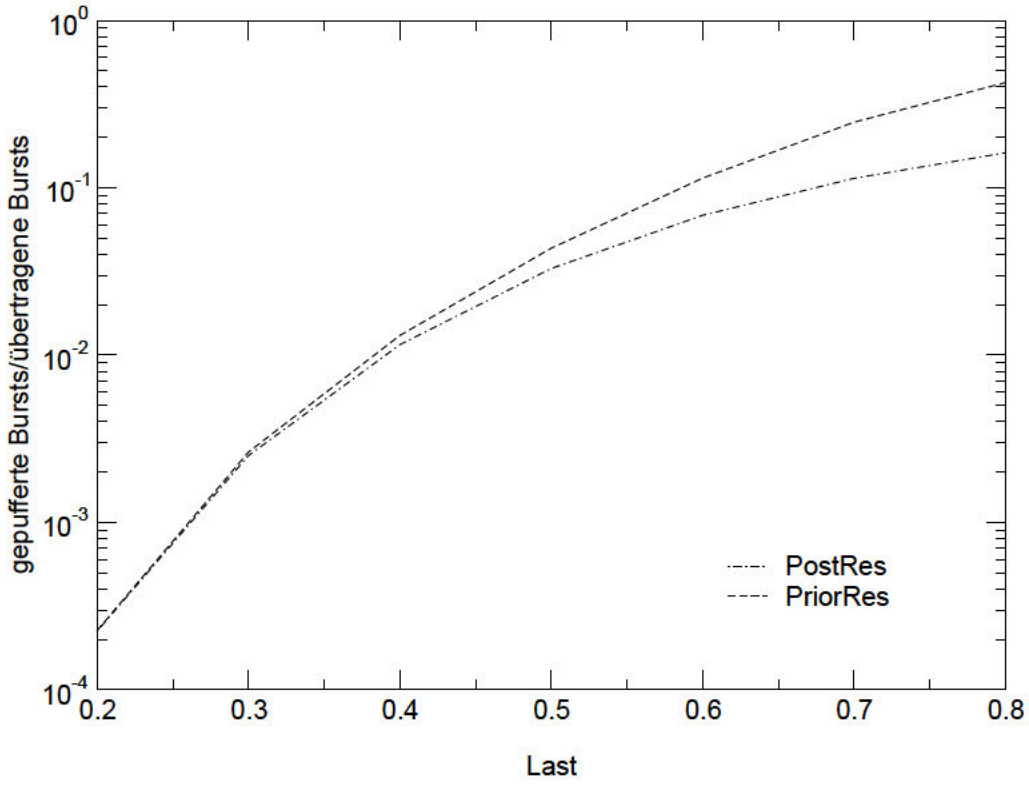

Abb. 4.5: Anteil der gepufferten Bursts in Abhängigkeit der Last

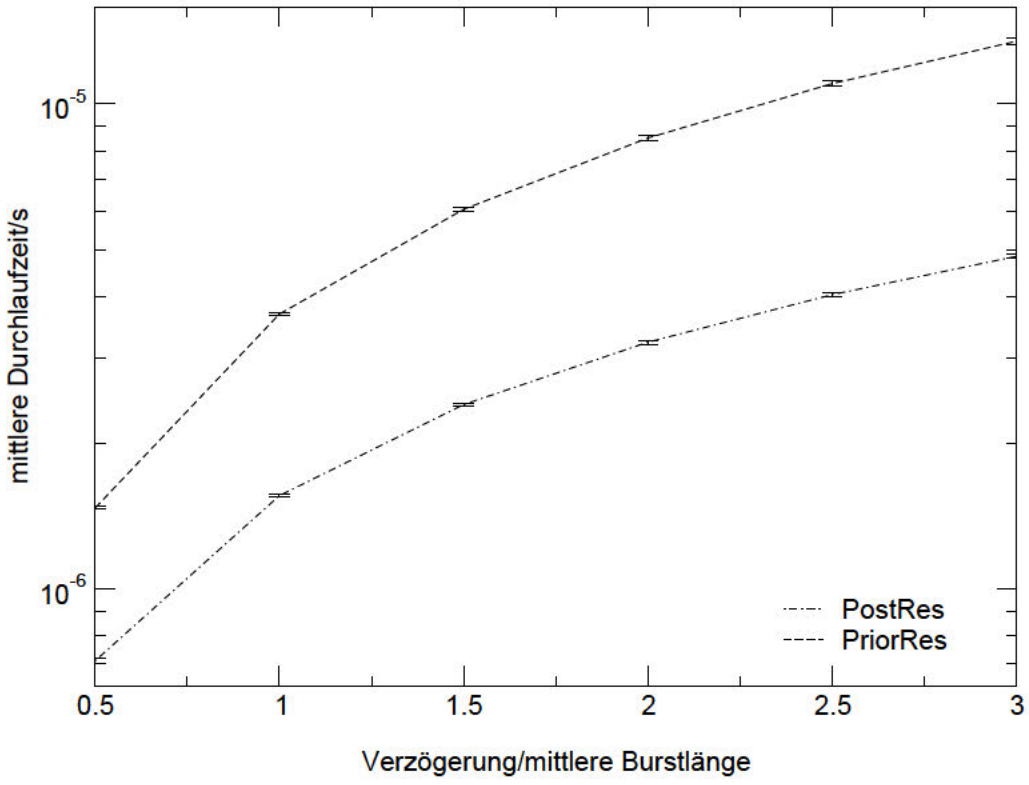

Abb. 4.6: Mittlere Durchlaufzeit in Abhängkeit der Verzögerung (Last 0.8)

bei einer Betrachtung der Pufferbelegung, die in Abbildung 4.7 dargestellt ist. Diese ist bei PriorRes – insbesondere bei hoher Last – wesentlich größer. Und das, obwohl PriorRes nur Bursts puffert, die auch übertragen werden können, während bei PostRes alle blockierten Bursts gepuffert werden, aber nur ein Teil davon später eine freie Wellenlänge findet.

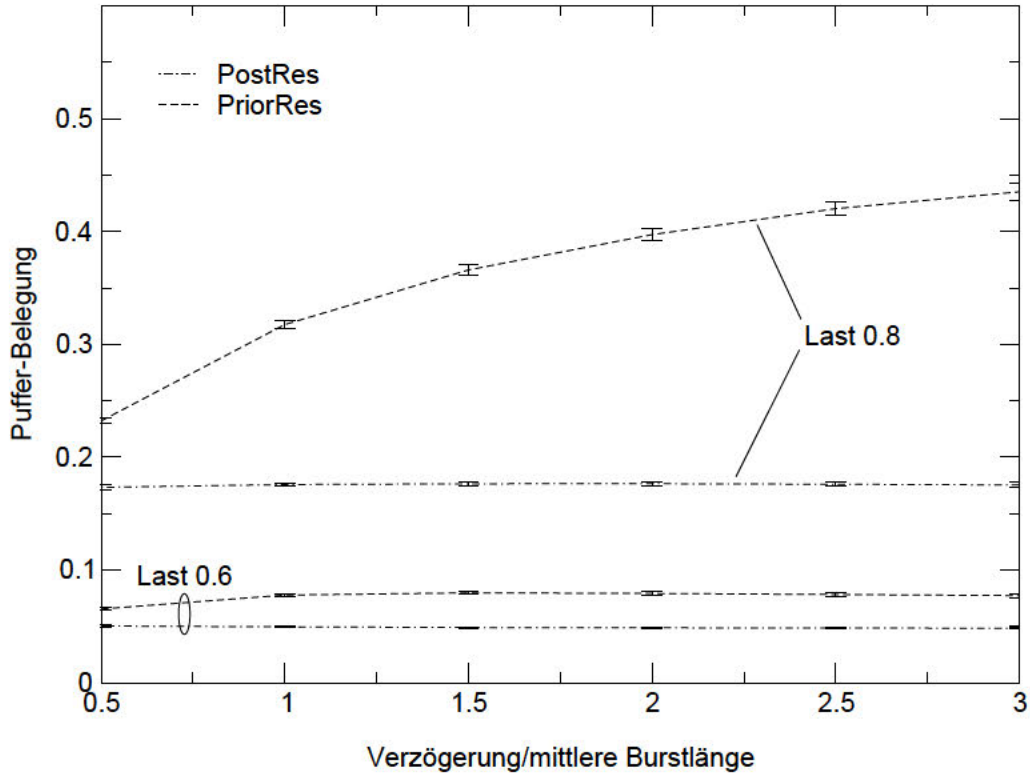

Abb. 4.7: Puffer-Belegung in Abhängigkeit der Verzögerung

4.4.3 Bewertung

PriorRes und PostRes unterscheiden sich zunächst nur in dem Zeitpunkt, zu dem eine Leitungsreservierung versucht wird. Dennoch kann aus den bisher dargestellten Simulationsergebnissen deutlich abgelesen werden, dass beide Verfahren signifikante Unterschiede aufweisen:

PriorRes führt in allen untersuchten Fällen zu niedrigeren Verlustwahrscheinlichkeiten als PostRes. Dies kann auf mehrere Ursachen zurückgeführt werden: Bei PostRes dient der Puffer dazu, blockierten Bursts nach einer gewissen Zeit eine zweite Zugriffschance zu geben. Ein Wiedereintritt kann durch zwei Burstarten behindert werden:

1. Extrem lange Bursts, die schon vor dem betrachteten Burst angekommen sind und die auch noch nach der maximalen Pufferzeit $B = b$ Wellenlängen blockieren.

2. Bursts, die den Knoten nach dem betrachteten Burst erreichen, direkt einen geeigneten Zeitschlitz finden und reservieren, so dass dieser nach dem Pufferaustritt nicht mehr verfügbar ist.

Da bei PriorRes blockierte Bursts einen Zeitschlitz in der Zukunft reservieren, spielen nachfolgend eintreffende Bursts überhaupt keine Rolle. Sehr lange, vorhergehende Burst können zwar auch hier eine Reservierung verhindern. Die Wahrscheinlichkeit, dass dies eintrifft, lässt sich aber durch Verzögerungen größer als die mittlere Burstdauer beschränken, was – wie schon erläutert – zu einer Priorisierung der gepufferten Bursts führt. Letztendlich konkurrieren bei großem b nur gepufferte Bursts um die Leitung. Durch die Vorausreservierungen sinkt die Wahrscheinlichkeit, dass eine direkte Übertragung möglich ist und die Blockierungen nehmen zu. Der Anteil an Bursts, der den Weg durch den Puffer nimmt, ist daher bei PriorRes relativ groß.
Die Tatsache, dass bei PriorRes blockierte Bursts faktisch mit einem Offset $\Delta = \delta + b$ reservieren können, führt zu einer starken Fragmentierung bei der Wellenlängenbelegung. Bursts, die länger als b sind, werden bei direkter Übertragung kaum einen geeigneten Zeitschlitz finden. Andererseits können kurze Bursts möglicherweise auch bei hoher Last noch eine für sie passende Lücke finden. Dies führt dazu, dass bei hoher Last lange Bursts, wenn überhaupt, nur über den Puffer übertragen werden können. Dieser Effekt ist sehr deutlich in Abbildung 4.8 zu erkennen: Je kürzer ein Burst beim PriorRes-Verfahren ist, desto niedriger ist die Verlustwahrscheinlichkeit. Bursts, die dagegen länger als die Pufferverzögerung b sind, werden häufig verworfen – unabhängig davon, wie lang der Burst konkret ist. Lange Bursts werden also bei PriorRes benachteiligt.
Für das PostRes-Verfahren hingegen, das nicht zwischen frisch im Switch ankommenden und gerade den Puffer verlassenden Bursts unterscheidet, sind Bursts jeder Länge gleichberechtigt.
An dieser Stelle ist anzumerken, dass diese Untersuchungsergebnisse für den Fall gelten, dass der Switch das JET-Reservierungsverfahren verwendet. Würde dagegen Horizon oder gar JIT verwendet werden, so würde dies für PriorRes dramatische Konsequenzen haben, da durch die weit in die Zukunft reichenden Reservierungen die dazwischenliegende Zeit ungenutzt verstreicht, was zu einer signifikanten Verschlechterung führt. Bei großer Verzögerung wird z.B. für Horizon praktisch der

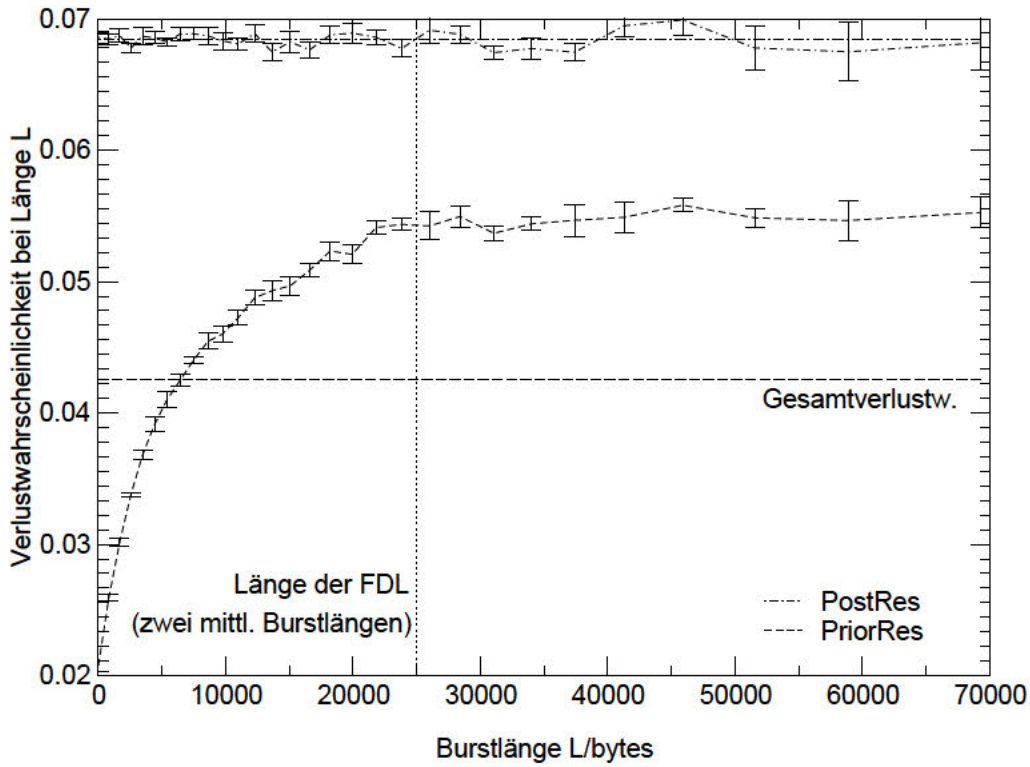

Abb. 4.8: Verlustwahrscheinlichkeit in Abhängigkeit der Burstlänge (Last 0.8)

pufferlose Fall erreicht, weil fast alle Bursts den Puffer passieren müssen und dieser dann lediglich als verlängerte Leitung dient (siehe [17]).
Somit ist PriorRes also ein Verfahren, mit dem sich niedrigere Verlustwahrscheinlichkeiten erzielen lassen, allerdings zu Lasten einer längenabhängigen Verlustwahrscheinlichkeit, einer relativ großen mittleren Durchlaufzeit und anderer Einschränkungen. PostRes dagegen ist in diesen Punkten wesentlich weniger kritisch.

4.4.4 Untersuchung anderer Längenverteilungen

Die Tatsache, dass verschieden lange Bursts bei PriorRes unterschiedlich erfolgreich sind, macht es interessant, neben der bisher verwendeten negativ-exponenziellen Burstlängenverteilungen andere Verteilungsfunktionen zu betrachten. Dies geschieht in Abbildung 4.9.
Eingetragen ist zunächst als Referenz die Burstverlustwahrscheinlichkeit ohne Puffer („Discard“). Wie von einem M/G/n-System zu erwarten, ist diese unabhängig von der Längenverteilung der Burstlängen (d. h. der Bedienzeiten). Neben der negativ-exponenziellen Verteilung wird unter „uniform“ eine gleichförmige Verteilung ($h \in [0, 2\,h_0]$) verstanden. „general 0.5“ bezieht sich auf eine verschobene negativ-expo-

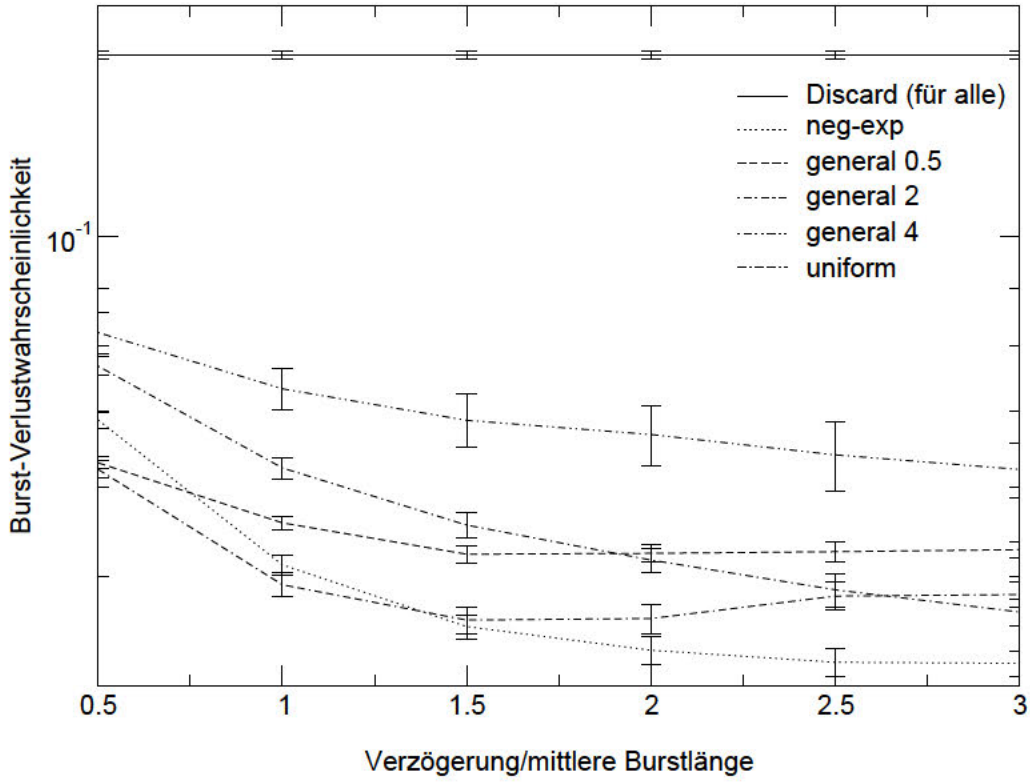

Abb. 4.9: PriorRes mit anderen Burstlängenverteilungen (Last 0.8)

nenzielle Verteilung 2. Ordnung mit Variationskoeffizient $c_T = 1/2$, „general 2“ bzw. „general 4“ auf eine hyperexponenzielle Verteilung 2. Ordnung mit $c_T = 2$ bzw. 4 (siehe auch [9]). Zu erkennen sind zwei Effekte:

1. Je größer der Variationskoeffizient c_T für „neg-exp“ bzw. „general 0.5“ usw. ist, d. h. je größer die Wahrscheinlichkeit ist, dass auch Bursts mit $h \gg h_0$ (und wegen $b = 2\,h_0$ auch $h \gg b$) auftreten, desto höher sind die Verlustwahrscheinlichkeiten.

2. Im Fall der gleichförmigen Verteilung und „general 0.5“ tritt für $b \approx 1.5\,h_0$ ein Minimum auf. Da hier keine Bursts länger als $2\,h_0$ vorkommen bzw. diese sehr unwahrscheinlich sind, bieten längere Verzögerungen für PriorRes keinerlei Vorteil.

Die gleichen Verkehrscharakteristika können auch in Verbindung mit dem PostRes-Verfahren untersucht werden (Abbildung 4.10). Aufgrund der Längenunabhängigkeit sollte hier die Verteilung der Burstlängen keine wesentliche Rolle spielen. Dennoch treten – wenn auch im Vergleich zu PriorRes geringere – Unterschiede in den Verlustwahrscheinlichkeiten auf. Dieser Effekt ist bislang ungeklärt.

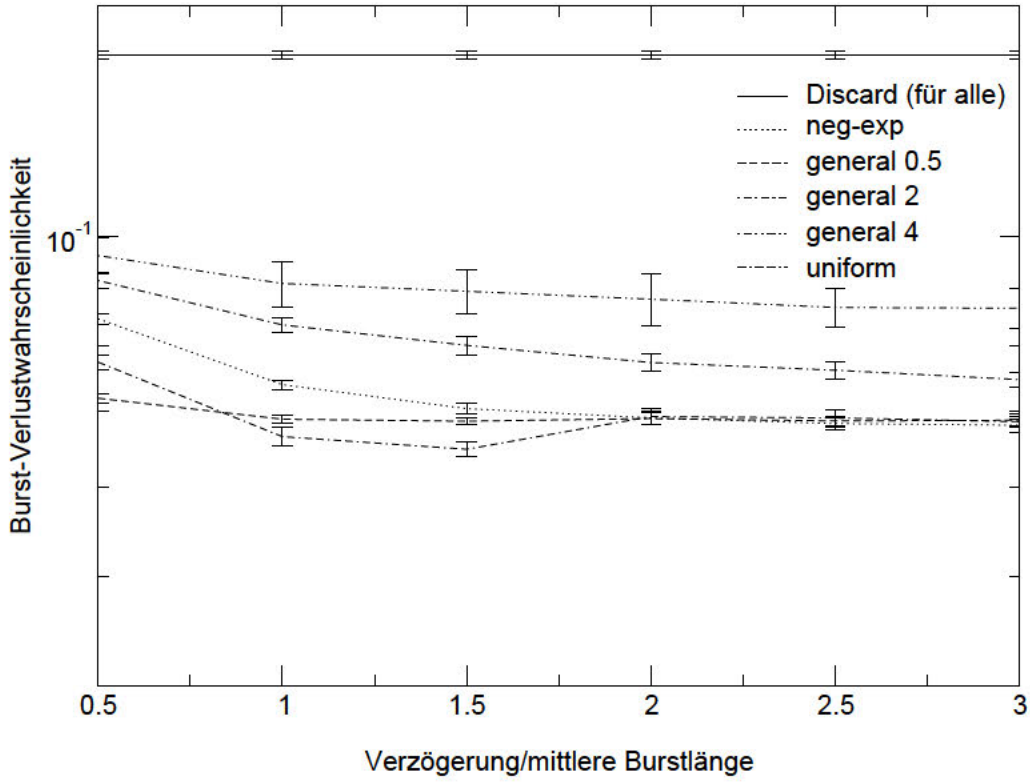

Abb. 4.10: PostRes mit anderen Burstlängenverteilungen (Last 0.8)

Neben anderen Längenverteilungen besteht auch die Möglichkeit, die Eigenschaften des Verkehrs durch andere Ankunftsprozesse zu verändern. Ergebnisse aus [17] deuten darauf hin, dass sich bei der Verwendung von selbstähnlichem Verkehr Phänomene zeigen könnten, die sich nicht durch Poisson-Ankünfte modellieren lassen. Dies sollte daher Gegenstand weiterer Untersuchungen sein.

4.5 Mehrmaliger Pufferdurchlauf

4.5.1 Burstverlustwahrscheinlichkeit

Der erste Schritt in Richtung flexiblerer Pufferstrukturen besteht dahin, von einer *feed forward*-Architektur auf *feed back* überzugehen, also mehrere Pufferdurchläufe zuzulassen. Weiterhin wird ein aus einer einzigen FDL bestehender Puffer („Schleife“) verwendet. Mit einer Maximalzahl an zugelassenen Durchläufen Q wird die Pufferkapazität, d. h. die Anzahl an Wellenlängen n_b im Puffer, ein interessanter Faktor. Darum wird ab jetzt eine FDL mit beschränkter Kapazität in die Betrachtungen mit einbezogen. Als Standardwert wird angenommen, dass im Puffer genauso viele Wellenlängen wie im Switch selbst verwendet werden: $n = n_\mathrm{b} = 8$. Auch die folgenden Messungen beziehen sich auf eine relativ hohe Last von $\rho = 0.8$.

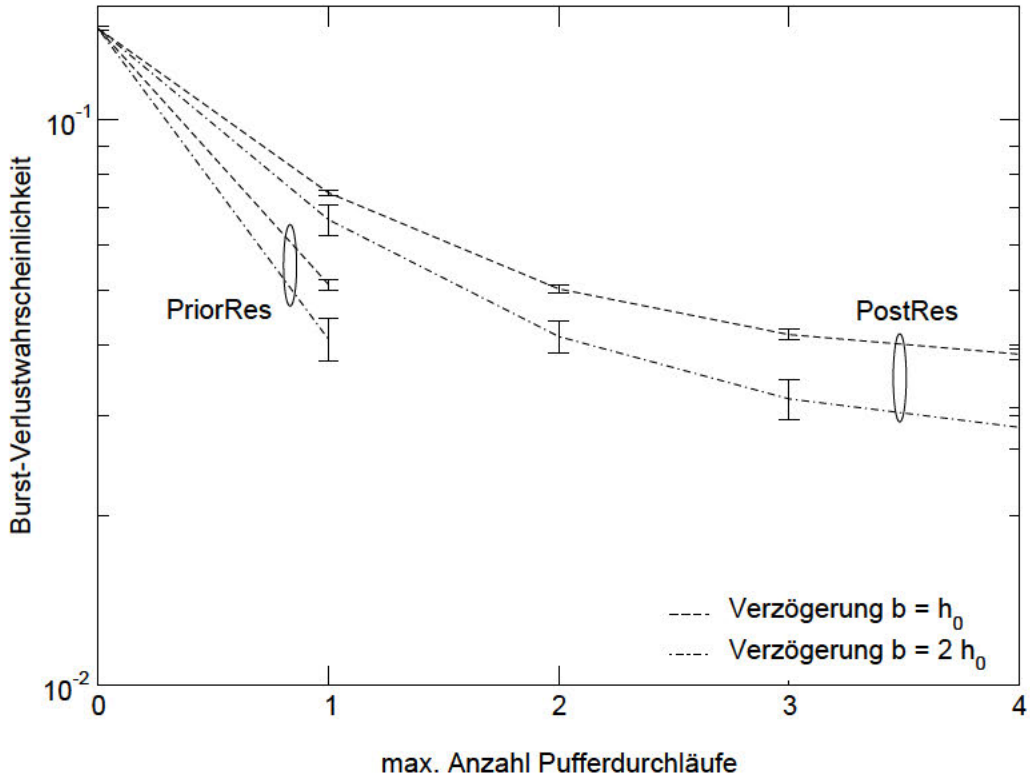

Abb. 4.11: Burstverlustwahrscheinlichkeiten bei mehreren Durchläufen

In Abbildung 4.11 ist dargestellt, welchen Einfluss die Obergrenze Q auf die Burstverlustwahrscheinlichkeit hat ($Q = 0$ entspricht einem Switch ohne Puffer). Wie schon erläutert, ist das PriorRes-Verfahren nicht in der Lage, mit mehreren Pufferdurchläufen umzugehen. Für PostRes dagegen zahlt sich dies aus: Je mehr Durchläufe erlaubt sind, desto mehr sinkt die Verlustwahrscheinlichkeit ab, und zwar unter

die PriorRes-Werte. Bei niedrigerer Last sind durch mehrere Durchläufe auch Verlustwahrscheinlichkeiten erreichbar, die deutlich unter denen von PriorRes liegen. Längere Verzögerungsleitungen sind auch in diesem Fall wieder etwas besser.
Weil bei *feed back* ein Burstverlust sowohl durch Überschreiten von Q als auch durch fehlenden Pufferplatz verursacht werden kann, wird in Abbildung 4.12 der Einfluss der Pufferkapazität untersucht. Aufgetragen ist die sich ergebende Burstverlustwahrscheinlichkeit bezogen auf den pufferlosen Fall über der normierten Kapazität n_b/n der FDL. Eine Kapazität von 2 bedeutet also, dass im Puffer doppelt so viele Wellenlängen wie auf einer Ausgangsleitung vorhanden sind.

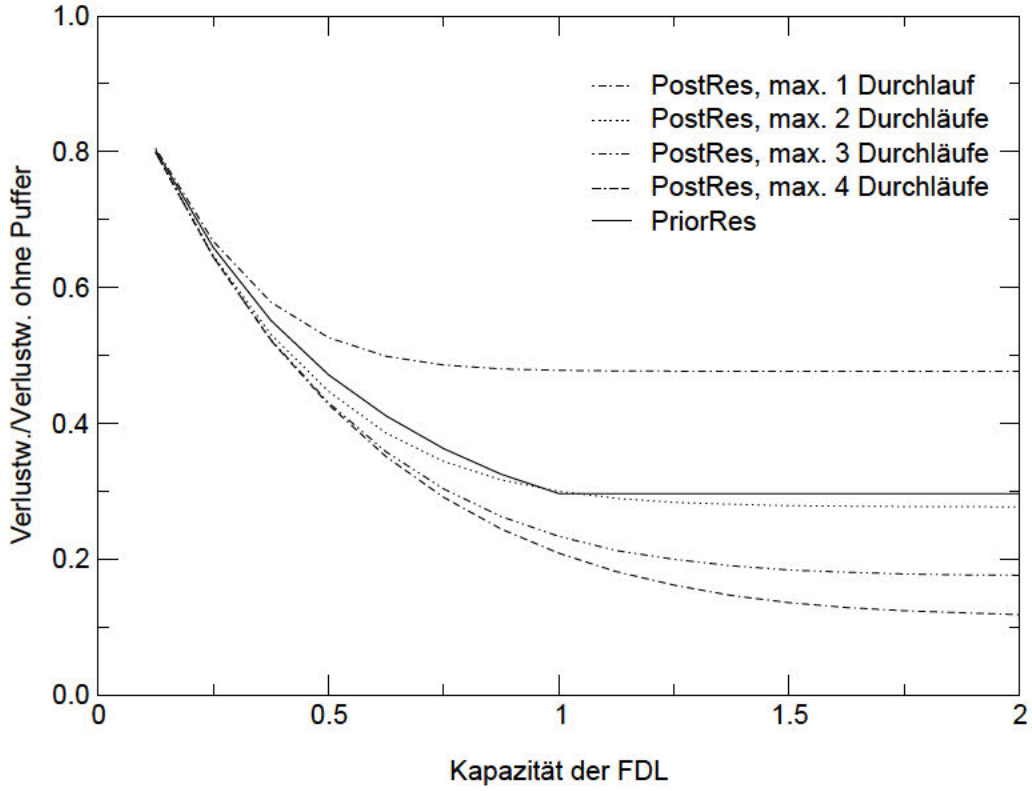

Abb. 4.12: Normierte Burstverlustwahrscheinlichkeit in Abhängigkeit der Pufferkapazität

Bei der Betrachtung von PriorRes fällt auf, dass ein Puffer mit $n_\mathrm{b} > n$ Wellenlängen zu keiner Verbesserung der Burstverlustwahrscheinlichkeit führt und bei $n_\mathrm{b} = n$ ein Knick auftritt. Dies ist dadurch zu erklären, dass schliesslich alle gepufferten Bursts nach dem Puffer auf eine Leitung mit n Wellenlängen passen müssen. Ein größerer Puffer dazwischen spielt dabei keine Rolle. Die Verlustwahrscheinlichkeit steigt aber deutlich an, wenn die Kapazität auf Werte kleiner als 1 reduziert wird. Hier zeigt sich wieder die Tatsache, dass bei PriorRes – gerade bei der hier verwendeten hohen

Last – sehr viele Bursts den Puffer passieren und sich Beschränkungen sofort negativ auswirken.

Im PostRes-Betrieb tritt für $Q = 1$ der Knick bei einer wesentlich geringeren Kapazität auf. Im dargestellten Beispiel würden eine Pufferkapazität von ungefähr 0.5 (entsprechend 4 oder 5 Wellenlängen im Puffer) völlig ausreichen, ohne dass sich die Verluste wesentlich vergrößern würden. Wird Q erhöht, so wird der vorhandene Puffer besser ausgenutzt und es kann, wie schon vorher festgestellt, PriorRes übertroffen werden. Dann macht es aber auch Sinn, mehr Wellenlängen im Puffer zur Verfügung zu stellen.

Um den Einfluss darzustellen, den die Länge der FDL auf die Verlustwahrscheinlichkeit hat, sind in Abbildung 4.13 eine Auswahl von Kurven für unterschiedliche Längen b eingezeichnet. Auch hier zeigt sich wieder, dass sich durch längere Verzögerungen niedrigere Verlustwahrscheinlichkeiten erreichen lassen, der prinzipielle Kurvenverlauf aber erhalten bleibt.

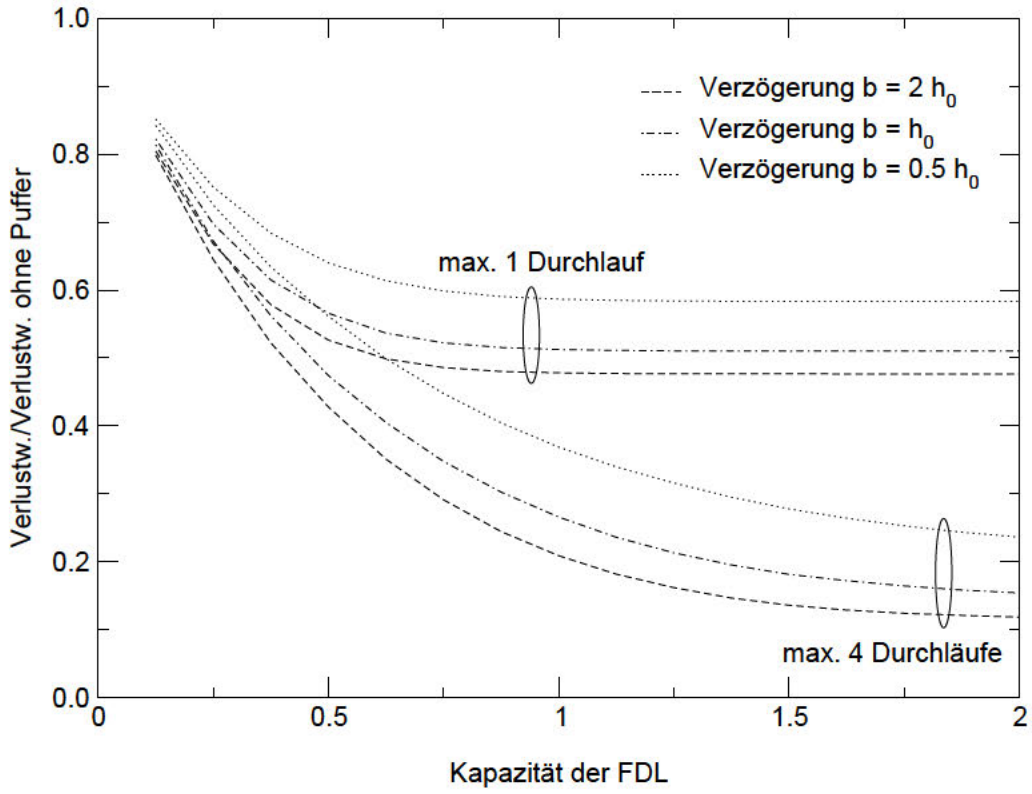

Abb. 4.13: PostRes-Verlustw. in Abhängigkeit der Pufferkapazität

4.5.2 Durchlaufzeit

Mehrere Pufferdurchläufe haben natürlich den Nachteil, dass sich die mehrfach gepufferten Bursts relativ lange im Knoten aufhalten. Welche Auswirkung dies hat, kann Abbildung 4.14 entnommen werden.

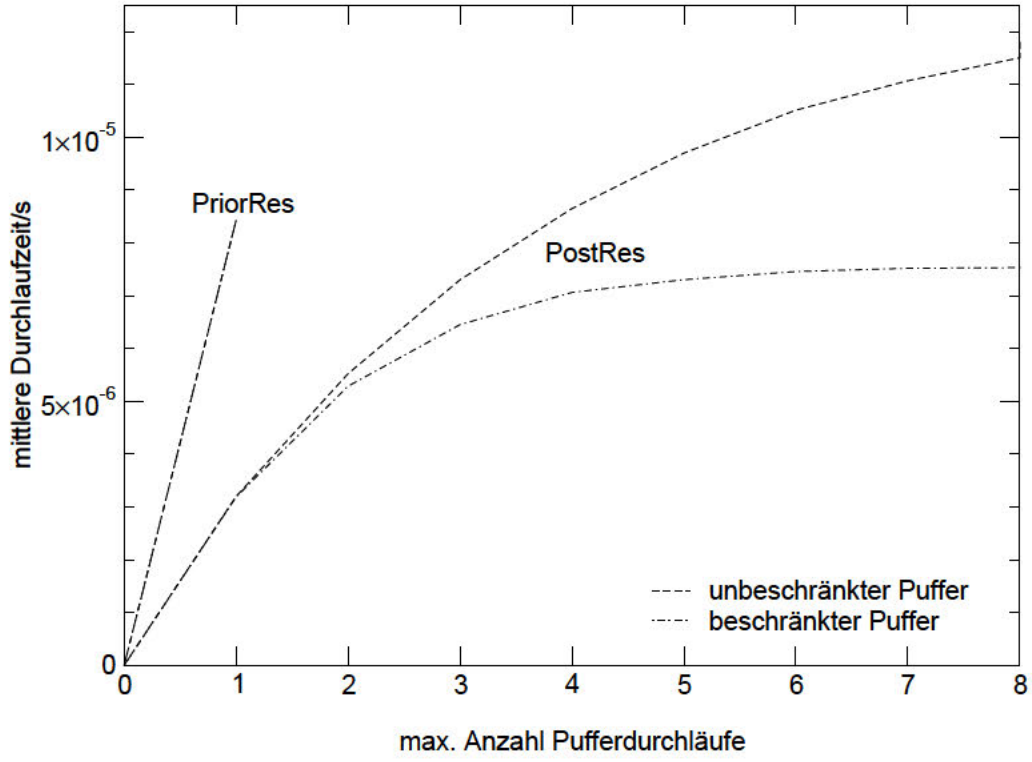

Abb. 4.14: Mittlere Durchlaufzeit bei mehreren Durchläufen

Mit zunehmendem Q nimmt die Transferzeit stark zu auf ca. $10\,\mu$s (bei einer FDL-Verzögerung von $b = 2\,h_0 = 20\,\mu$s). $10\,\mu$s ist die mittlere Übertragungszeit eines Bursts und ungefähr auch die Durchlaufzeit von PriorRes. Neben einem auf $n_\mathrm{b} = 8$ beschränkten Puffer wird auch ein Szenario mit unbeschränktem Puffer angenommen, das die Grenze des Machbaren darstellt und bei dem die Durchlaufzeit noch größer ist. Um dies zu erklären, sind in Abbildung 4.15 die komplementären Verteilungsfunktionen der Durchlaufzeiten für unterschiedliche Q eingezeichnet.
Der stufenförmige Verlauf der Funktion wird durch die deterministische Länge der FDL verursacht. Die Sprünge treten daher bei den ganzzahligen Vielfachen von $b = 2\,h_0$ auf. Nur sehr wenige Bursts nutzen die volle Anzahl möglicher Pufferdurchläufe aus, weil ein Großteil der gepufferten Bursts schon nach wenigen Durchläufen entweder übertragen werden kann oder aber durch den beschränkten Pufferplatz verworfen werden muss. Dies erklärt auch, warum eine sehr große An-

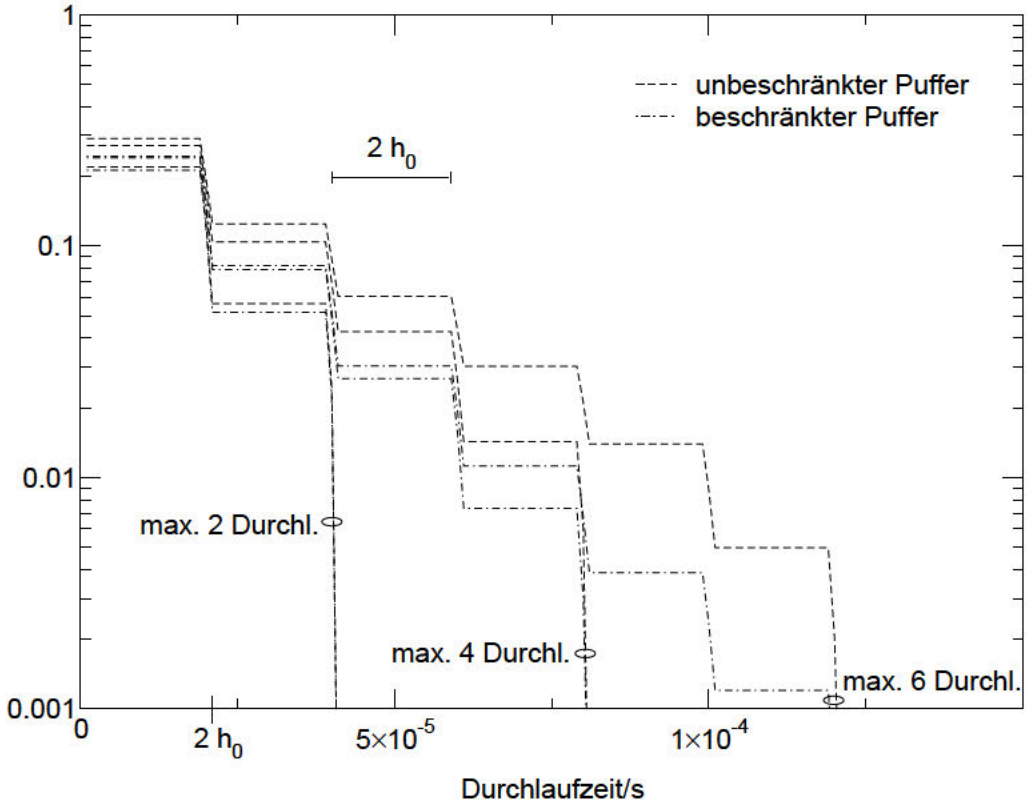

Abb. 4.15: Komplementäre Verteilungsfunktion der Transferzeiten

zahl zulässiger Wiederholungen bei kleinem Puffer keine wesentliche Verbesserung bringt und eine Sättigung eintritt (siehe Abbildung 4.14). Dennoch tritt bei größeren Werten von Q ein Effekt auf: Der Verlauf der komplementären Verteilungsfunktion liegt über denjenigen für kleinere Q. Durch die niedrigere Verlustwahrscheinlichkeit und die damit größere Leitungsbelegung treten nämlich mehr blockierte Bursts auf, die die Pufferbelegung erhöhen. Noch deutlicher tritt dieses Phänomen bei der Verwendung unbeschränkter Kapazität auf: Bei beliebig großem Puffer ist die Wahrscheinlichkeit, länger im Puffer zu bleiben, deutlich größer, weil Bursts nur durch das Erreichen der maximalen Durchlaufzahl Q verloren gehen.

4.5.3 Bewertung

Durch das Zulassen mehrerer Pufferdurchläufe kann für PostRes die Verlustwahrscheinlichkeit deutlich reduziert werden. Schon wenige Durchläufe reichen aus, um bei gleicher Puffergröße besser als die PriorRes-Strategie zu sein, und dies ohne die Nebeneffekte, die mit PriorRes einhergehen. Ein *feed back*-Puffer, bestehend aus einer einzigen FDL und betrieben im PostRes-Verfahren, ist daher eine einfache und effiziente Lösung zur Reduzierung der Burstverlustwahrscheinlichkeiten. Dies gelingt für den untersuchten Fall mit hoher Last ($\rho = 0.8$) um ungefähr eine Größenordung, für niedrigere Last ist aber nach Kapitel 4.3 eine erheblich stärkere Absenkung zu erwarten.

4.6 Puffer mit mehreren Verzögerungsleitungen

Eine andere Möglichkeit, Bursts zu verschiedenen Zeiten auf der Leitung einfügen zu können, besteht darin, mehrere unterschiedlich lange Verzögerungsleitungen zu verwenden. Dadurch wird eine neue Fragestellung aufgeworfen: In welche FDL wird ein Bursts eingeschrieben?[5] Zu diesem Zweck wurden in Kapitel 3.3.2 drei Verfahren eingeführt: PriorRes, PostRes und PostResRandom. Ziel der nachfolgenden Abschnitte ist es, diese zu vergleichen.

Die Untersuchungen beschränken sich auf „treppenförmige“ Puffer aus festen Verzögerungsleitungen, deren Längen ganzzahlige Vielfache von h_0 sind und (linear) zunehmen. Dies ist der Fall, der meist in der Literatur angenommen wird. Zur Minimierung der Durchlaufzeit wird bei PriorRes und PostRes zunächst eine Pufferung in der kürzesten Leitung versucht. Um eine hinreichende „Auffächerung“ zu erhalten, werden bis zu $N = 5$ Puffer verwendet. Als kürzeste Verzögerung wird eine mittlere Burstlänge verwendet ($b = h_0$), die maximale Verzögerung ist damit $B = N\,b \leq 5\,b$.

4.6.1 Burstverlustwahrscheinlichkeit

Die Effizienz der Verfahren hängt von der Dimensionierung der einzelnen Verzögerungsleitungen ab, d. h. deren Länge und deren Kapazität. Aus praktischen Erwägungen sind für die Zahl der Wellenlängen je Verzögerungsleitung zunächst zwei Fälle relevant:

Erstens kann in jeder Verzögerungsleitung die gleiche Anzahl an Wellenlängen wie auf den Ausgangsleitungen ($n = n_{\mathrm{b}i}$) verwendet werden. Dies bedeutet natürlich, dass für $N > 1$ die gesamte Pufferkapazität zunimmt. Für diese Dimensionierung ergeben sich in Abbildung 4.16 die ersten 3 Kurven („FDL-Kap. 1“). Bei PriorRes kann dadurch die Burstverlustwahrscheinlichkeit stark reduziert werden, im dargestellten Bereich ungefähr zwei Größenordnungen gegenüber dem pufferlosen Fall und mehr als eine Größenordnung gegenüber einem einfachen Puffer. PostRes bringt in diesem Fall gar keine und PostResRandom nur geringe Verbesserungen, wenn N vergrößert wird.

Um z. B. die Anzahl der notwendigen Wellenlängenkonverter zu beschränken, wäre ein anderer anzustrebender Fall, im Puffer *insgesamt* nur n Wellenlängen zur Verfügung zu stellen, d. h. hinsichtlich der Gesamtzahl an Wellenlängen einen gleich

[5] Bei einem *broadcast-and-select*-Switch wird ggf. in mehreren Verzögerungsleitungen parallel verzögert, und dann eine passende FDL ausgewählt. Dieser Spezialfall sei hier nicht weiter betrachtet.

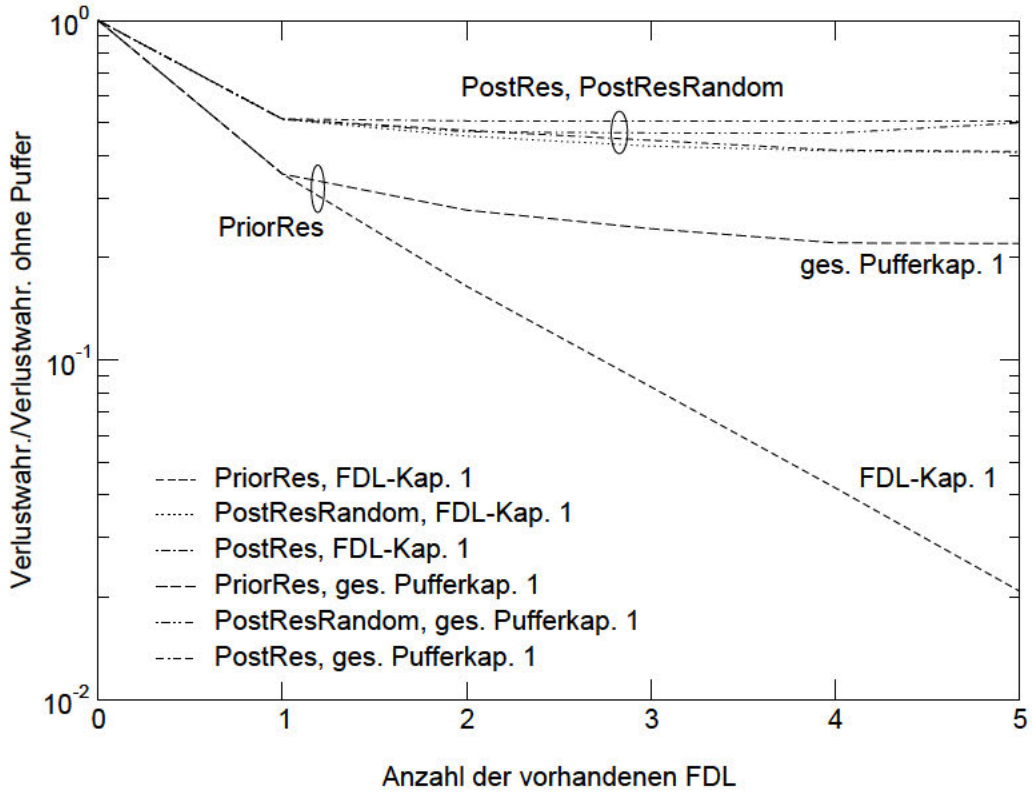

Abb. 4.16: Burstverlustwahrscheinlichkeiten für mehrere FDL

großen Puffer zu verwenden, der nur anders strukturiert ist. Dann sollte folgende Bedingung gelten:

$$\sum_{i=1}^{N} n_{\mathrm{b}i} = n \tag{4.9}$$

Wird für alle Puffer die gleiche Kapazität vorgesehen, dann folgt daraus:

$$n_{\mathrm{b}i} = n_{\mathrm{b}} = \frac{n}{N} \tag{4.10}$$

Dies ist der zweite betrachtete Ansatz, für den sich die in Abbildung 4.16 ebenfalls dargestellten Verläufe ergeben („ges. Pufferkap. 1“). Einige der Messwerte wurden dabei durch Interpolation berechnet. In allen drei Verfahren stellt sich schon bei 2 bis 3 FDLs eine Sättigung ein. Hier ist also – ganz im Gegensatz zum obigen Szenario – eine Bereitstellung von mehr als 3 FDLs wenig sinnvoll. Da PriorRes immer eine relativ große Pufferkapazität benötigt, ist es klar, dass eine derartige Einschränkung einen starken Einfluss auf die Verlustwahrscheinlichkeit hat. Auffällig ist aber, dass nun PostRes geringfügig besser als PostResRandom abschneidet, und auch für eine zunehmende Anzahl an FDLs leichte Verbesserungen auftreten. Dies hängt damit

zusammen, dass hier auch bei PostRes die längeren FDLs genutzt werden.
Um den ersten Ansatz noch etwas weiterzuverfolgen, sind in Abbildung 4.17 die Verlustwahrscheinlichkeiten für verschiedene Anzahlen an Verzögerungsleitungen dargestellt. Auf der Abszisse ist die – nun variable – Kapazität *einer* der verwendeten Verzögerungsleitungen aufgetragen, die Ordinate ist wieder auf den pufferfreien Fall normiert.

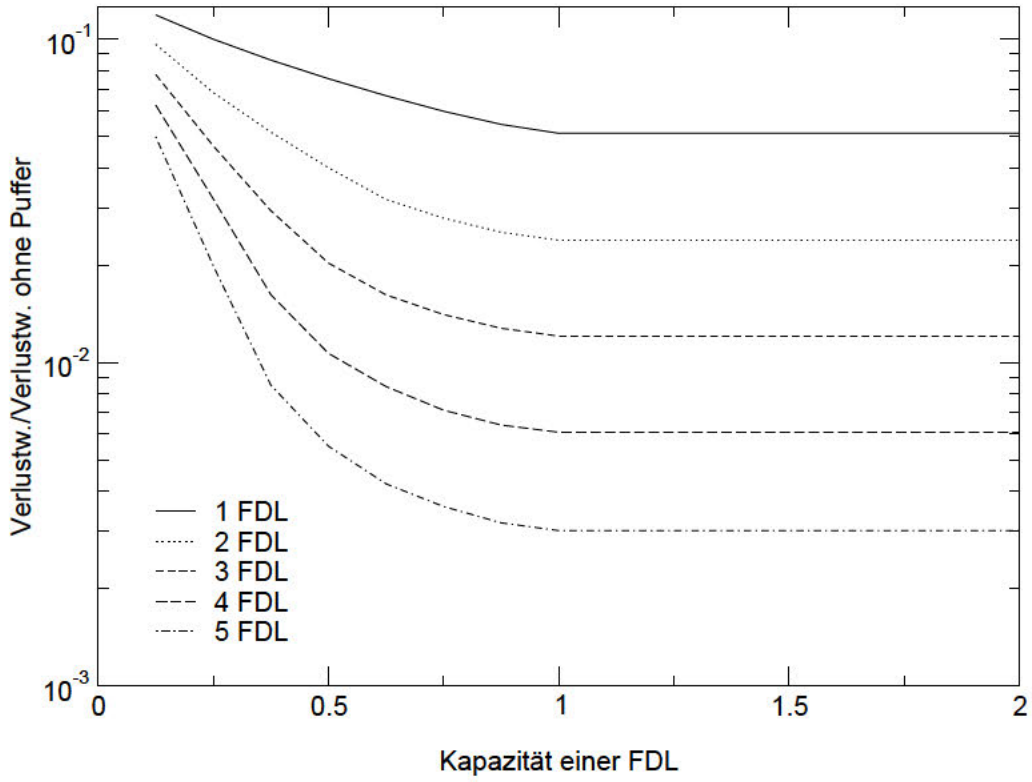

Abb. 4.17: PriorRes-Burstverlustwahrscheinlichkeiten für mehrere FDL

Durch die Verwendung mehrerer Verzögerungsleitungen fällt die Verlustwahrscheinlichkeit sehr stark ab, beispielsweise bei $N = 5$ FDL um ungefähr den Faktor 300. Die Vergrößerung des Puffers zahlt sich also deutlich aus. Wie schon bei einer FDL gilt auch hier der Grundsatz, dass bei PriorRes im Puffer maximal so viele Wellenlängen wie auf der Ausgangsleitung vorgesehen werden sollten – darüber hinausgehende Kapazität bleibt ohne Wirkung.
Abbildung 4.18 enthält die Ergebnisse, die sich bei genau der gleichen Pufferkonfiguration in Verbindung mit dem PostRes-Verfahren ergeben – allerdings in einem anderen Maßstab dargestellt.
Die Bereitstellung mehrerer Verzögerungsleitungen hat hier nur dann einen Einfluss, wenn die einzelne FDL nur über sehr wenige Wellenlängen verfügt ($n_{\mathrm{bi}} < n$).

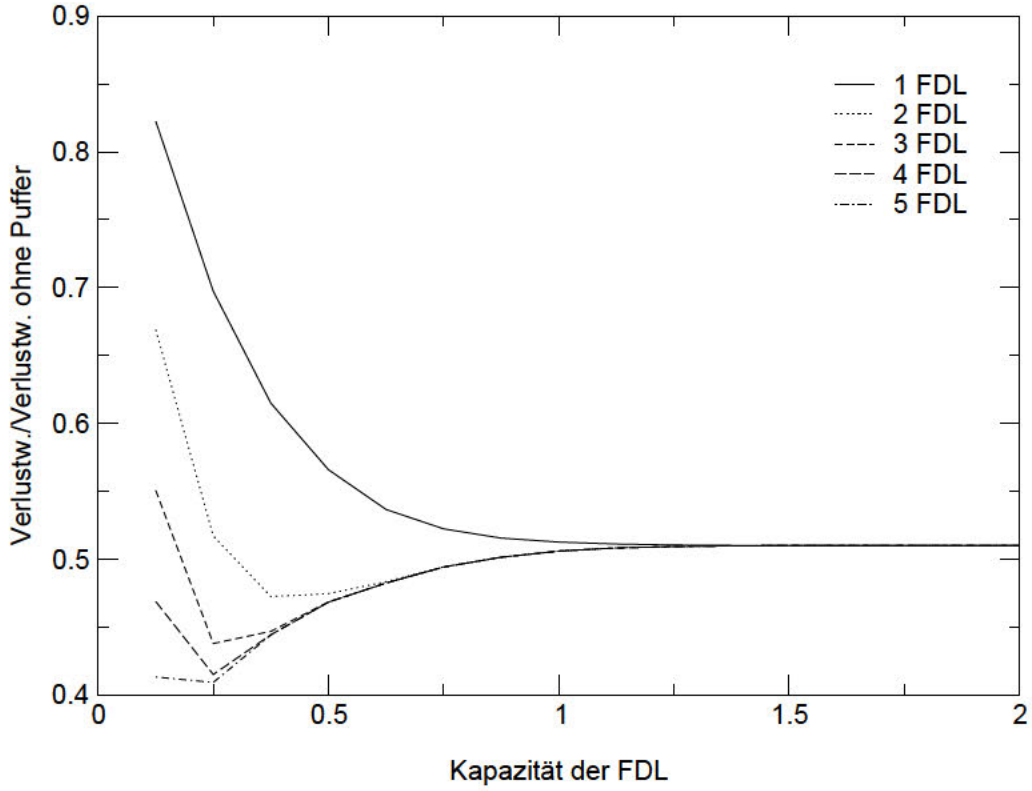

Abb. 4.18: PostRes-Burstverlustwahrscheinlichkeiten für mehrere FDL

Dies hängt mit der schon erläuterten Tatsache zusammen, dass PostRes ohne weitere Vorkehrungen stets in die erste verfügbare FDL puffert, solange diese nicht belegt ist. Verfügt die kürzeste Verzögerungsleitung über ausreichend Pufferplatz, werden weitere möglicherweise vorhandene nicht genutzt. Eine Nutzung weiterer Verzögerungsleitungen geschieht erst dann, wenn die erste FDL wegen nur weniger Wellenlängen belegt ist. In diesem Fall führen die unterschiedlichen Längen zu einer besseren Entzerrung, da blockierte Bursts unterschiedlich lang verzögert werden und auch eine längere Überbrückung möglich ist. Daraus resultieren niedrigere Verlustwahrscheinlichkeiten. Die Verbesserung durch unterschiedliche FDL ist allerdings so gering, dass sie den Aufwand dafür kaum rechtfertigt.

Um die unterschiedlichen Leitungen besser auszunutzen, kann man die schon eingeführte Erweiterung PostResRandom benutzen, die eine FDL zufällig zuteilt. In diesem Fall ergibt sich das in Abbildung 4.19 dargestellte Resultat: Werden Bursts nach einer statistischen Verteilung unterschiedlich lang gepuffert, so bewirkt dies eine geringfügige Absenkung der Verlustwahrscheinlichkeit. Auch hier sind unterschiedlich lange FDLs also etwas günstiger. Unter Berücksichtigung der Tatsache, dass aber – wie im nachfolgenden Abschnitt gezeigt wird – die Verzögerung erheblich

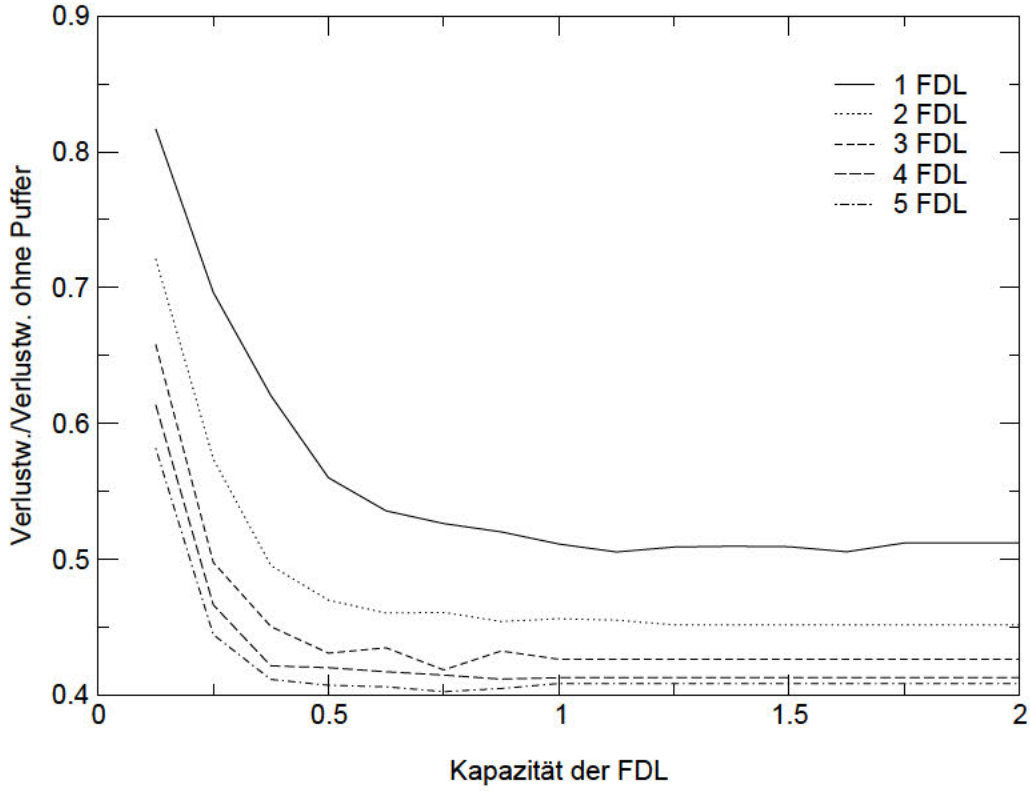

Abb. 4.19: PostResRandom-Burstverlustwahrscheinlichkeiten für mehrerer FDL

zunimmt, dürften diese Verbesserungen kaum relevant sein.
Um die Effekte, die sich durch die unterschiedlichen Zuweisungsstrategien ergeben, nochmals zu veranschaulichen, ist in Abbildung 4.20 die Belegung der einzelnen Verzögerungsleitungen dargestellt (für insgesamt 4 FDL im Puffer).
Zunächst erkennt man, dass die Belegung für die zweite oben genannte Möglichkeit („ges. Pufferkap. 1“) in allen Fällen größer ist. Dies hängt natürlich damit zusammen, dass hier der Puffer insgesamt wesentlich kleiner ist. Für beide Ansätze hat PriorRes jeweils die höchsten Belegungen, wobei die erste (kürzeste) FDL stärker genutzt wird als die weiteren im Puffer existierenden. PriorRes tendiert also auch in diesem Szenario dazu, sehr viele Bursts durch den Puffer zu schicken. Bei PostRes ist dies prinzipiell genauso. Nur dass für eine hinreichend dimensionierte FDL Nr. 1 (was bei „FDL-Kap. 1“ offensichtlich der Fall ist) die anderen Leitungen praktisch gar nicht genutzt werden. Bei PostResRandom, das die Bursts gleichmäßig auf alle FDL verteilt, ist die Belegung erwartungsgemäß gleich groß.

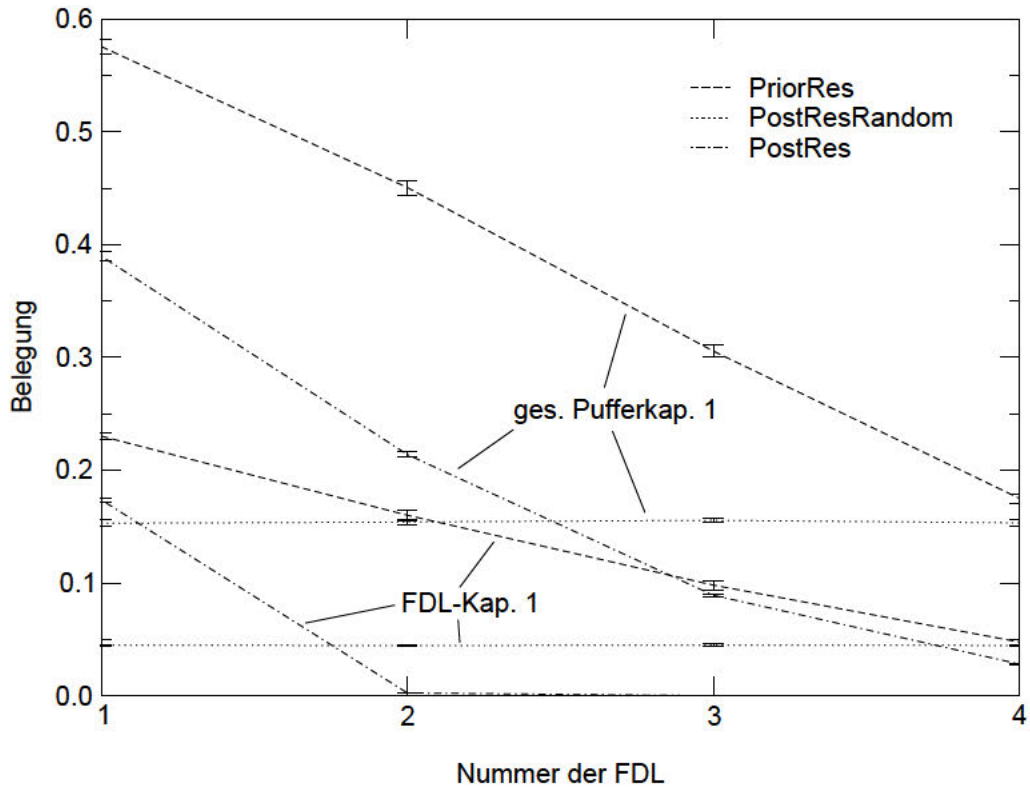

Abb. 4.20: Belegung der einzelnen Verzögerungsleitungen

4.6.2 Durchlaufzeit

Wie in den vorherigen Untersuchungen ist es auch hier sinnvoll, die Auswirkungen auf die Durchlaufzeit zu betrachten.

Abbildung 4.21 zeigt, dass die Durchlaufzeit bei PriorRes auch bei mehrstufigen Puffern größer als diejenige bei PostRes bzw. PostResRandom ist. Bei PostResRandom nimmt die mittlere Durchlaufzeit aufgrund der speziellen Wahl der Leitungslängen mit der Anzahl der vorhandenen Verzögerungsleitungen linear zu. Da für die verwendete Dimensionierung $n_{\mathrm{b}i} = n$ PostRes weitere FDL fast nicht nutzt, besteht natürlich für die mittlere Durchlaufzeit keine Abhängigkeit. Die Abflachung der PriorRes-Kurve kann durch den niedrigeren Einfluss erklärt werden, den die längeren, weniger ausgelasteten FDLs haben.

Eine Aussage darüber, wie die jeweiligen Verzögerungsleitungen genutzt werden, kann auch aus der komplementären Verteilfunktion der Transferzeiten gewonnen werden (siehe Abbildung 4.22). Wie zu erwarten, treten für diese Pufferkonfiguration bei PostRes praktisch keine Bursts auf, die um mehr als $b = h_0 = 10\ \mu$s verzögert werden. Die Verteilung bei PostResRandom nimmt linear ab, dies wird allerdings durch den logarithmischen Maßstab verzerrt dargestellt. Bei PriorRes wer-

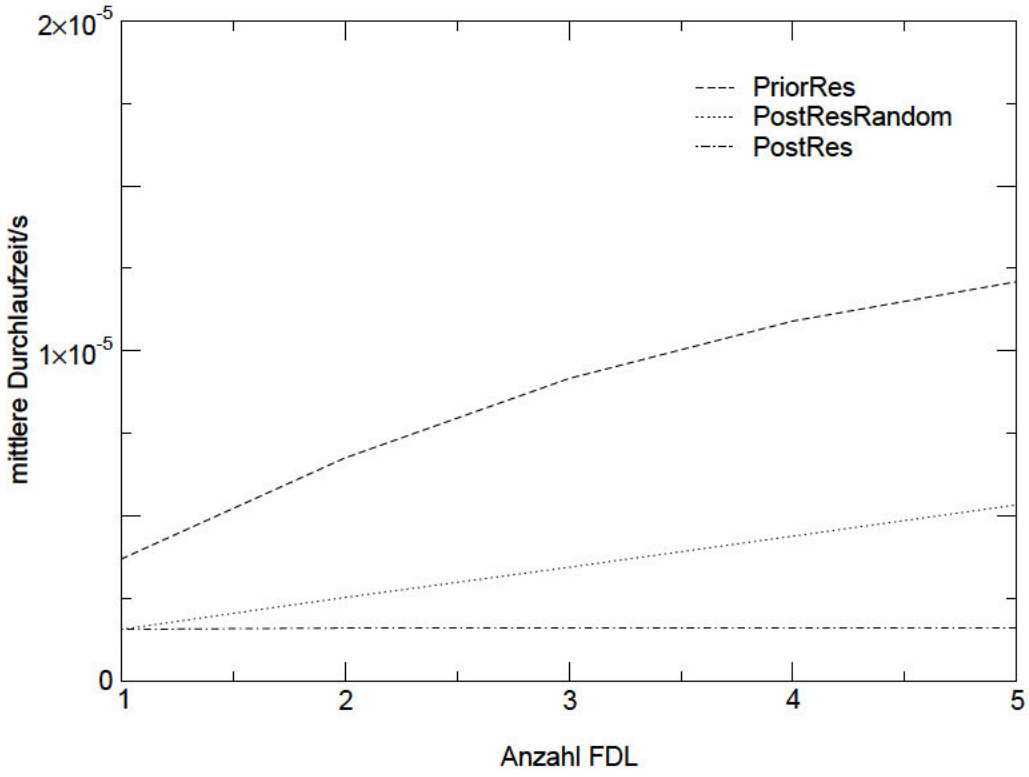

Abb. 4.21: Mittlere Durchlaufzeiten für mehrere FDL

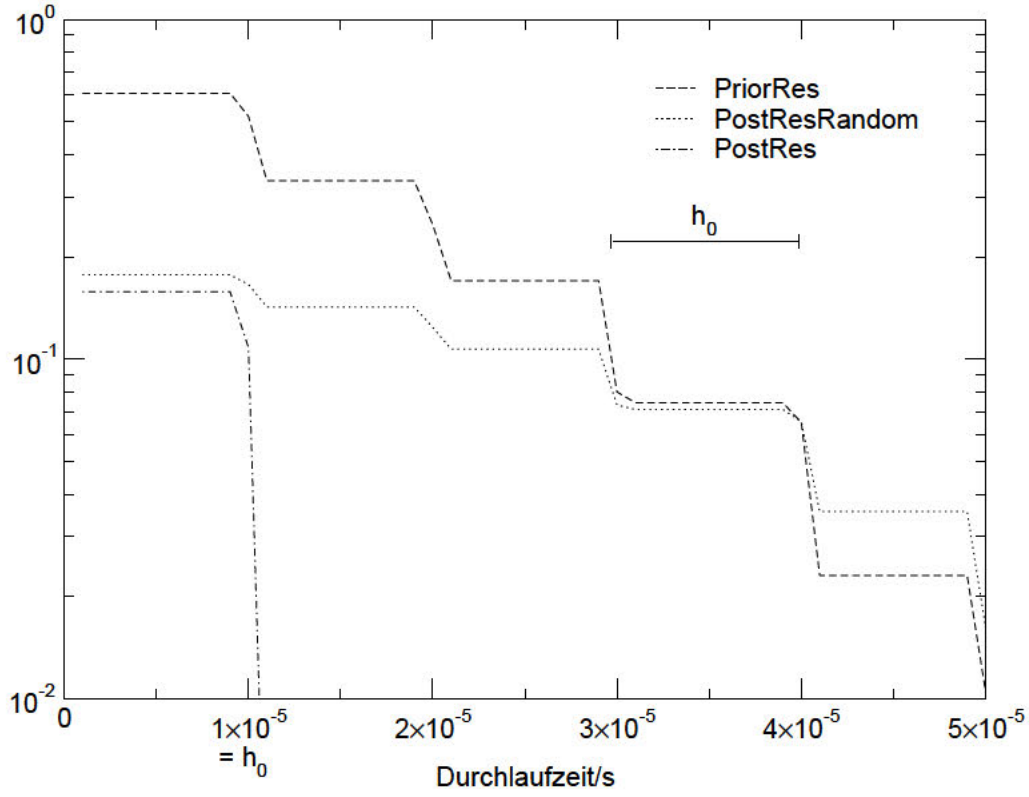

Abb. 4.22: Komplementäre Verteilungsfunktion der Durchlaufzeiten (bei 5 FDL)

den tatsächlich ein gewisser Anteil an Bursts relativ lang gepuffert, wenn auch weniger als in den kürzeren FDL. Die längeren FDLs sind also eine Ausweichmöglichkeit für Bursts, für die die kurzen Leitungen nicht ausreichen.

4.6.3 Dimensionierung der Verzögerungsleitungen

Angesichts der unterschiedlichen Auslastung der verschiedenen Verzögerungsleitungen stellt sich die Frage, ob der bislang verfolgte Weg, nämlich stets in allen Verzögerungsleitungen gleich viele Wellenlängen vorzusehen, die beste Lösung darstellt. Eine denkbare Alternative wäre, die sich insbesondere für das PriorRes-Verfahren anbieten könnte, mehr Platz in den kurzen Verzögerungsleitungen zur Verfügung zu stellen. Schließlich ist der Fall, dass eine zukünftige Leitungsreservierung zwar möglich, die entsprechende FDL aber belegt ist, bei kurzen FDL zu vermeiden. Dies würde entweder zu einer länger als notwendigen Verzögerung bzw. gar einem Burstverlust führen.

In Abbildung 4.23 und 4.24 ist die PriorResBurstverlustwahrscheinlichkeit über der mittleren Durchlaufzeit aufgetragen, und zwar für die verschiedenen Wellenlängenkonfigurationen, die in einem Puffer mit 4 Verzögerungsleitungen möglich sind (unter Berücksichtigung der in Gleichung (3.3) gegebenen Relation). Die Gesamtzahl an Wellenlängen im Puffer wird stets so gewählt, dass sie derjenigen auf der Ausgangsleitung ($n = 8$) entspricht. Beide Abbildungen unterscheiden sich in den Längen der verwendeten FDL. Im linken Fall ist der bisher betrachtete Fall einer linearen Zunahme betrachtet, während für die rechte Abbildungen die Längen h_0, $2\,h_0$, $4\,h_0$ und $8\,h_0$ gewählt wurden.

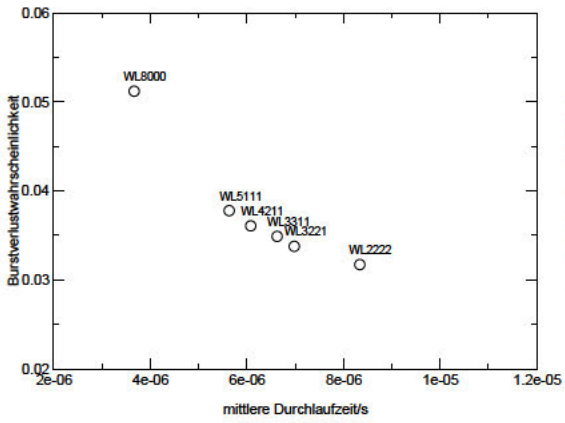

Abb. 4.23: Zusammenhang Verlustw. und Durchlaufzeit (Längen 1,2,3,4)

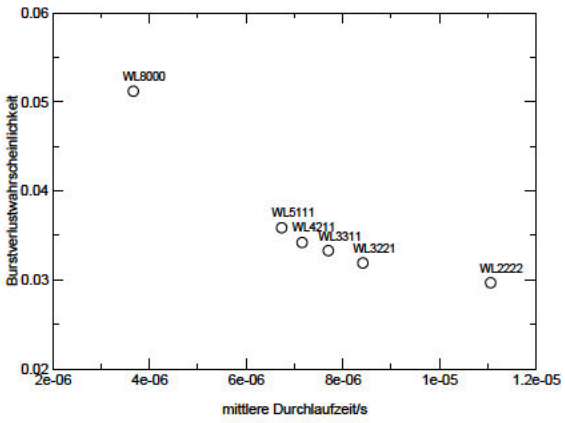

Abb. 4.24: Zusammenhang Verlustw. und Durchlaufzeit (Längen 1,2,4,8)

„WL2222“ bedeutet beispielsweise, dass in der ersten FDL $n_{b1} = 2$ Wellenlängen vorgesehen sind, ebenso in der zweiten, usw. Dies ist der schon im vorherigen Abschnitt untersuchte Fall. Zum Vergleich ist auch der Fall eingetragen, dass nur eine einzige FDL verwendet wird („WL8000“).
Der Vergleich der verschiedenen Szenarien zeigt den schon mehrfach beobachteten Sachverhalt, dass eine Absenkung der Verlustwahrscheinlichkeit stets durch eine Zunahme der Durchlaufzeit begleitet wird. Die kürzeste Durchlaufzeit wird natürlich für eine einzige FDL (mit der Länge $b_1 = h_0$) erreicht, die niedrigste Verlustwahrscheinlichkeit ergibt sich dann, wenn die längste FDL relativ groß dimensioniert ist („WL2222“). Die anderen möglichen Kombinationen liegen zwischen beiden Extremen.
Aus dem Verlauf kann geschlossen werden, dass die längeren Leitung bei PriorRes eine nicht unerhebliche Bedeutung hat. Auch wenn für viele Bursts eine kürzere FDL ausreicht, um einen passenden Zeitschlitz in der Zukunft zu reservieren, bietet die längste FDL die größte Priorisierung gegenüber anderen Bursts. Selbst längeranhaltende Verstopfungen können damit überwunden werden. Außerdem hat die längste Verzögerungsleitung natürlich auch den Vorteil, dass zu dem mit ihr erreichbaren Reservierungszeitpunkt noch keine fragmentierte Wellenlängenbelegung vorliegt. Deshalb kann die Burstverlustwahrscheinlichkeit auch noch weiter reduziert werden, wenn – wie in Abbildung 4.24 – von der bisher verwendeten linearen Längenverteilung abgewichen und eine geometrische verwendet wird. Die längste Leitung kann hier um die Zeit $b_4 = 8\,h_0$ verzögern. Dies bewirkt eine weitere, leichte Absenkung der Verluste bei gleichzeitig zunehmender Durchlaufzeit. Die Veränderung ist aber so marginal, dass kaum von der linearen „Treppe“ an Verzögerungsleitungen abgewichen werden muss.

4.6.4 Bewertung

Puffer bestehend aus mehreren, unterschiedlich langen Verzögerungsleitungen führen bei PriorRes zu deutlich geringeren Verlusten, allerdings wird das Problem der verhältnismäßig großen Durchlaufzeit dadurch noch verstärkt.
Auch andere Reservierungsverfahren schneiden etwas besser ab, dennoch bieten diese für Puffer mit mehreren FDLs kaum Vorteile: PostResRandom bewirkt keine signifikante Senkung der Verlustwahrscheinlichkeit gegenüber einer einzelnen FDL, hat aber erheblich größere Durchlaufzeiten. Eine reine PostRes-Strategie ist nur sinnvoll, wenn die einzelnen Verzögerungsleitungen eine sehr kleine Kapazität haben. Andernfalls droht die Gefahr, dass Hardware verschwendet wird.

Prinzipiell besteht die Möglichkeit, PostRes mit einer intelligenteren Auswahlstrategie zu kombinieren, die z. B. die Leitungsbelegungen berücksichtigt und die Dauer von Verstopfungssituationen abschätzt, trotzdem aber vor dem Eintritt in den Puffer keine Reservierung vornimmt. Damit wäre es sicher möglich, die Leistung von PostResRandom zu überbieten. Solange derartige Algorithmen nicht angewendet werden, macht nur das PriorRes-Verfahren im Zusammenhang mit Puffern aus mehreren Verzögerungsleitungen Sinn.
Aus diesem Grund kann an dieser Stelle auch darauf verzichtet werden, derartige Strukturen in Kombination mit *feed back*-Architekturen zu untersuchen, da auch dafür keine wesentliche Verbesserung gegenüber einem aus einer einzigen FDL bestehenden Puffer zu erwarten sind. Eine derartige Struktur verfügt auch über sehr viele Freiheitsgrade.

4.7 Pufferung bei differenzierter Behandlung von Verkehrsklassen

4.7.1 Prinzip der Klassentrennung

Der Offset δ eines Bursts bestimmt den Abstand zwischen dem Kontrollpaket und den eigentlichen Daten. Wird in einem Knoten ein Kontrollpaket empfangen, so versucht dieser, einen passenden Zeitschlitz zu reservieren. Je größer der Offset δ ist, desto früher findet diese Reservierung statt. Bei einer weit in die Zukunft reichenden Reservierung ist die Übertragung der allermeisten Bursts bereits beendet, so dass mit großer Wahrscheinlichkeit eine freie Wellenlänge vorhanden ist, die belegt werden kann. Dementsprechend ist die Burstverlustwahrscheinlichkeit niedrig.
Qiao und Yoo schlagen daher vor, durch die Verwendung eines zusätzlichen QoS-Offsets δ_{QoS} bestimmte Bursts bei der Reservierung von Ressourcen zu bevorzugen [13]. Je größer δ_{QoS} ist, desto geringer ist die Verlustwahrscheinlichkeit. Dies ermöglicht die Einführung von verschiedenen Verkehrsklassen, die sich nur durch den verwendeten Offset unterscheiden und die bei geeigneter Wahl auch fast vollständig voneinander isoliert werden können. In Optical Burst Switching kann also durch einen sehr einfachen Mechanismus Dienstgüte zur Verfügung gestellt werden. Deterministische Garantien sind allerdings mit diesem Verfahren nicht möglich, und die Klassentrennung setzt die Verwendung des JET-Reservierungsschemas voraus.
Die Klassentrennung in OBS-Netzen ist u.a. in [4, 5] umfassend analytisch und simulativ untersucht worden. Die nachfolgenden Betrachtungen beschränken sich daher auf die Frage, welche Effekte sich durch die Einführung von Puffern in ein OBS-Netz mit differenzierten Verkehrsklassen ergeben.

4.7.2 Zwei Verkehrsklassen

Im Folgenden wird ein Szenario mit zwei verschiedenen Verkehrsklassen angenommen. Aktuelle Diskussionen im Internet tendieren dazu, zwei Klassen („stream“ und „elastic“) als für die meisten Anwendungsfälle völlig ausreichend zu betrachten.
Die hochpriore (hp) Verkehrsklasse zeichnet sich gegenüber der niederprioren durch einen zusätzlichen QoS-Offset aus. Unter Vernachlässigung des Basisoffsets, der bei beiden Klassen gleich ist, hat damit die hochpriore Klassen einen Offset $\delta_{\mathrm{H}} = \delta_{\mathrm{QoS}}$, während für die niederpriore (np) Klasse $\delta_{\mathrm{L}} = 0$ gilt. Der Offset hat einen entscheidenen Einfluss auf die Isolation der hochprioren von der niederprioren Klasse. Soll eine weitreichende Klassentrennung erreicht werden, muss der Offset so groß sein, dass die Übertragung von niederprioren Bursts mit großer Wahrscheinlichkeit vor

dem Beginn eines hochprioren Bursts geendet hat:

$$\delta_{\mathrm{QoS}} > E[h_{\mathrm{L}}] = h_0 \tag{4.11}$$

Tatsächlich sollte der Offset ein Mehrfaches der mittleren Burstlänge betragen: Eine einfache Betrachtung in [20] liefert beispielsweise für $\delta_{\mathrm{QoS}} = 3\,h_0$ eine 95 %-ige Klassentrennung.

Um die Komplexität der Betrachtungen einzuschränken, wird im Folgenden ein *feed forward*-Puffer verwendet, der aus einer einzigen FDL der Länge $b = h_0$ besteht. Der Anteil der hp-Klasse am gesamten Verkehrsaufkommen beträgt 30 % und die übrigen Parameter werden aus den vorhergehenden Messungen übernommen. Beide Verkehrsklassen besitzen also eine gleiche mittlere Burstlänge und in beiden Fällen seien die Längen zunächst negativ-exponenziell verteilt.

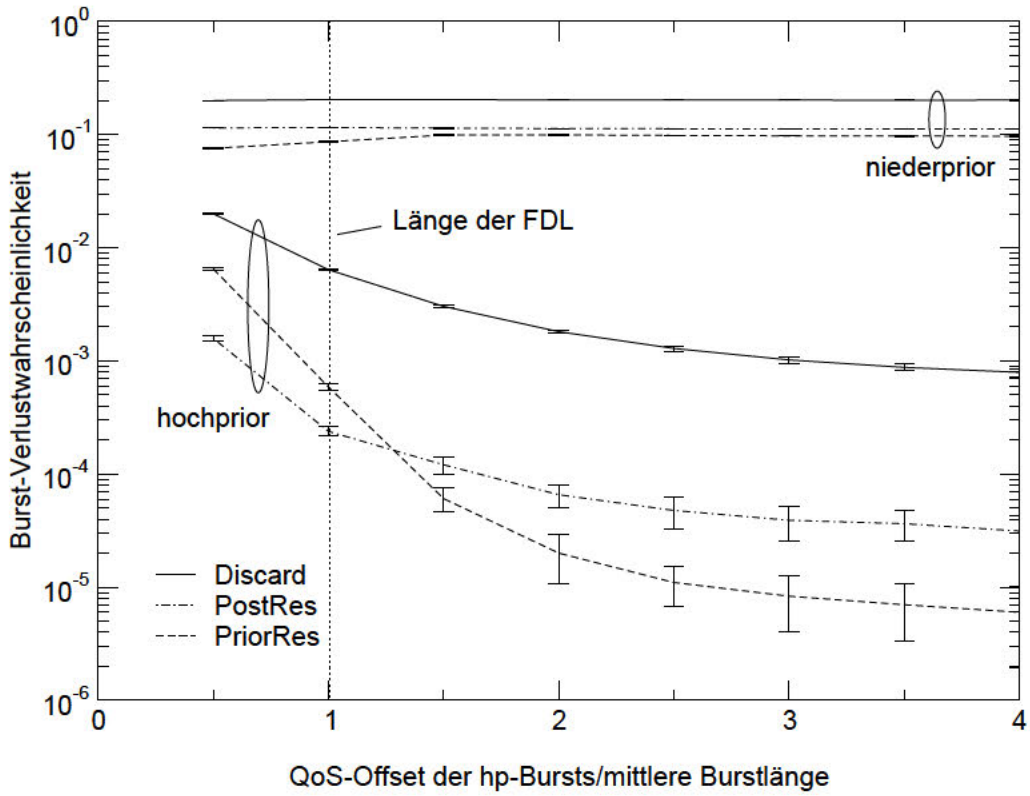

Abb. 4.25: Burstverlustwahrscheinlichkeiten bei zwei Verkehrsklassen

Abbildung 4.25 zeigt die Burstverlustwahrscheinlichkeiten für beide Verkehrsklassen. Für den „Discard"-Fall ist deutlich die zunehmende Trennung der Klassen zu erkennen, die sich in einer niedrigeren Verlustwahrscheinlichkeit für die hochpriore Klasse äußert. Wird ein Puffer in den Knoten eingebaut, so können beide Verkehrsklassen profitieren:

Arten von Bursts	**Offset Δ**
hochpriore, gepufferte Bursts	$\delta_{\mathrm{H}} + b = \delta_{\mathrm{QoS}} + b$
hochpriore, direkt übertragene Bursts	$\delta_{\mathrm{H}} = \delta_{\mathrm{Qos}}$
niederpriore, gepufferte Bursts	$\delta_{\mathrm{L}} + b = b$
niederpriore, direkt übertragene Bursts	$\delta_{\mathrm{L}} = 0$

Tabelle 4.2: Auswirkung von Puffern bei PriorRes

Im Zusammenhang mit PostRes sinken beide Verlustwahrscheinlichkeiten deutlich, bei den hp-Bursts ist die Verbesserung allerdings wesentlich größer. Dies hängt damit zusammen, dass hochpriore Bursts zweimal bevorzugt werden: Erstens ermöglicht ihnen der QoS-Offset δ_{QoS} eine frühere Reservierung des Puffers, falls einmal eine Blockierung auftreten sollte, und zweitens haben sie auch bei dem Wiedereintritt nach dem Puffervorgang eine höhere Erfolgswahrscheinlichkeit. Der prinzipielle Kurvenverlauf bleibt also bei PostRes erhalten, nur sind die Verlustwahrscheinlichkeiten kleiner. Dies hat zur Konsequenz, dass der QoS-Offset δ_{QoS} und die Verzögerung b unabhängig voneinander gewählt werden können: Mit ersterem wird das Verhältnis der Verlustwahrscheinlichkeiten von hochpriorer und niederpriorer Verkehrsklasse eingestellt, während durch $b \geq h_0$ eine ausreichende Überbrückung von Verstopfungen erreicht werden kann.
Wird dagegen das PriorRes-Verfahren verwendet, sind Offset und Verzögerung nicht mehr völlig unabhängig: In Abbildung 4.25 erkennt man dies daran, dass für einen QoS-Offset $\delta_{\mathrm{QoS}} < b \ (= h_0)$ die Verlustwahrscheinlichkeit der niederprioren Klasse abfällt, während die der hp-Bursts relativ hoch bleibt. Zwar ist für beide Klassen die Verlustwahrscheinlichkeit geringer als im pufferlosen Fall, aber dennoch ist hier die Klassentrennung teilweise aufgehoben. Die Ursache dafür ist in der Tatsache zu suchen, dass – wie schon mehrfach erwähnt – bei PriorRes gepufferte Bursts bei der Reservierung bevorzugt werden, da sie ähnlich wie die hp-Bursts voraus reservieren. Durch die Verwendung eines Puffers gibt es praktisch nicht mehr zwei, sondern vier Verkehrsklassen, die sich jeweils durch einen unterschiedlichen Offset Δ zwischen Reservierung und Datenübertragung auszeichnen (siehe Tabelle 4.2).
Ist bei gegebener FDL-Länge b der QoS-Offset δ_{QoS} zu klein, so kann es sein, dass blockierte niederpriore Bursts durch den Puffervorgang genauso weit oder gar noch weiter in die Zukunft reservieren können wie hochpriore Bursts. Obwohl hochpriore Bursts immer noch auf den Puffer ausweichen können, wirkt sich die Konkurrenz auf die Verlustwahrscheinlichkeiten aus. Um eine weitreichende Klassenisolation zu er-

reichen, müssen die hochprioren von den gepufferten niederprioren getrennt werden, d. h. δ_{QoS} so groß gewählt werden, dass gilt:

$$\delta_{\mathrm{QoS}} > b + E[h_{\mathrm{L}}] = b + h_0 \tag{4.12}$$

Sinnvoll ist auch hier die Wahl eines Vielfaches der Burstlänge, z. B. $\delta_{\mathrm{QoS}} = b + 3\,h_0$. In diesem Punkt ist sowohl die Verlustwahrscheinlichkeit der niederprioren wie auch der hochprioren Verkehrsklasse niedriger als im PostRes-Fall.
Die Tatsache, dass die Verlustwahrscheinlichkeit vom Zusammenhang zwischen QoS-Offset und Verzögerung abhängt, muss aber nicht als Nachteil aufgefasst werden: Der Mechanismus der teilweisen Aufhebung der Klassentrennung kann auch dazu genutzt werden, gezielt die Verlustwahrscheinlichkeiten von hochpriorer und niederpriorer Verkehrsklasse einzustellen.
In [6] wird gezeigt, dass die Verlustwahrscheinlichkeit der hp-Bursts von der Längen-Verteilung der niederprioren Bursts abhängt. Dies lässt sich analog auf den Fall mit Puffern übertragen. In den Abbildungen 4.26 und 4.27 wird daher für die niederprioren Bursts andere Verteilungsfunktionen verwendet. Die hochpriore Klasse besitzt weiterhin eine negativ-exponenzielle Burstlängenverteilung. Die verwendeten Bezeichnungen entsprechen denen aus Kapitel 4.4.4.

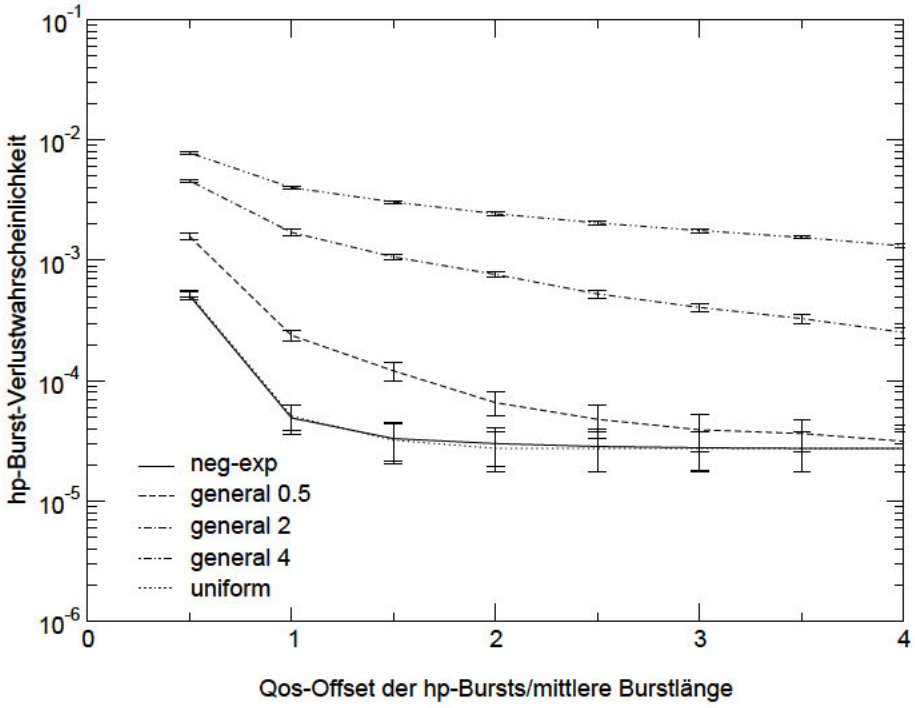

Abb. 4.26: Burstverlustwahrscheinlichkeiten der hochprioren Verkehrsklasse bei PostRes und verschiedenen Längenverteilungen der lp-Bursts

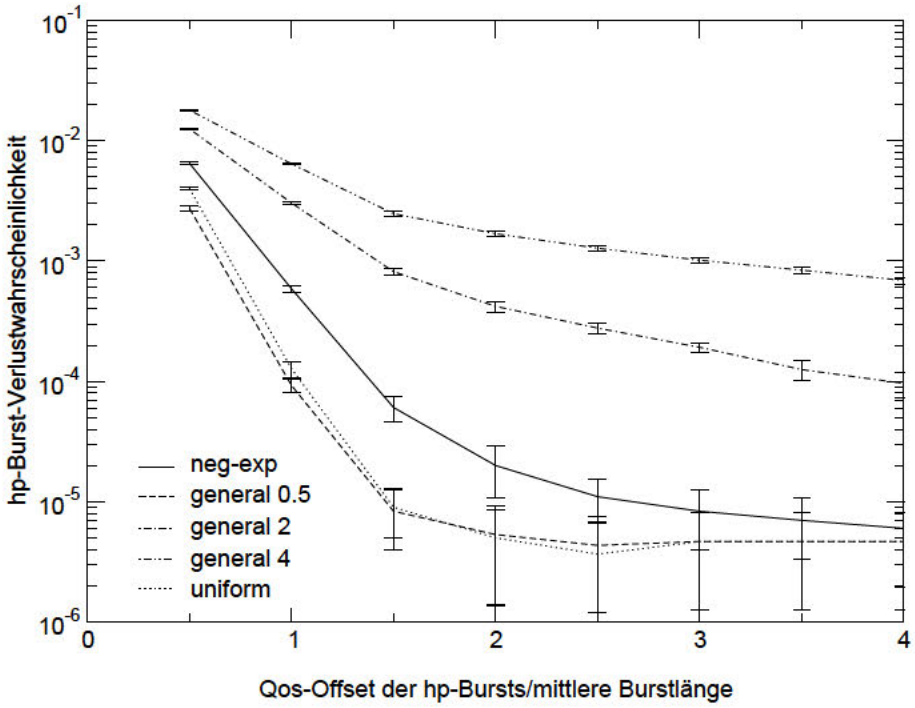

Abb. 4.27: Burstverlustwahrscheinlichkeiten der hochprioren Verkehrsklasse bei PriorRes und verschiedenen Längenverteilungen der lp-Bursts

In beiden Verfahren ist für Verteilungen, bei denen mit großer Wahrscheinlichkeit längere Bursts vorkommen (d. h. der Variationskoeffizient c_T groß ist) die Verlustwahrscheinlichkeit der hochprioren Bursts erheblich größer und es ist ein wesentlich höherer QoS-Offset δ_{QoS} notwendig, um diese zu senken. Dieses Verhalten entspricht demjenigen für den pufferlosen Fall in [6]. Dies zeigt, dass auch bei Verwendung von Puffern auf eine günstige Verkehrscharakteristik geachtet werden muss. Insbesondere ist darauf zu achten, dass nicht zu viele zu langen niederprioren Bursts auftreten.

4.7.3 Verallgemeinerung

Die obigen Ergebnisse lassen sich auf mehr als 2 Verkehrsklassen und kompliziertere Pufferstrukturen ausdehnen: Bei mehr als zwei Verkehrsklassen wird dann eine Trennung der höherprioren von den Klassen mit einer geringeren Priorität erreicht, wenn sich der QoS-Offset jeweils um einen Betrag δ_{diff} unterscheidet. Ohne Puffer kann z. B. $\delta_{diff} = 3h_0$ gewählt werden. Bei Verwendung von Puffern mit dem PostRes-Verfahren reicht dieser Wert weiterhin aus, die Klassentrennung bleibt beim Puffern erhalten. Die jeweilig höchstpriore Klasse wird von den Puffern dann am meisten profitieren. Bei PriorRes dagegen müssen die Offsets so gewählt werden, dass ein „Abstand“ zwischen gepufferten Bursts einer Klasse und direkt übertra-

genen Bursts der nächsthöheren besteht. Nach [20] ist dies erfüllt, wenn bei einer maximal möglichen Verzögerung B eines Puffers gilt: $\delta_{\text{diff}} = 3\,h_0 + B$. Aus dieser Beziehung folgt aber, dass bei relativ vielen Verkehrsklassen und/oder komplizierten Pufferstrukturen mit großem B relativ große Offsetwerte entstehen und damit die Ende-zu-Ende-Verzögerung der hochprioren Verkehrsklassen stark zunimmt. Bei PostRes oder relativ einfachen Verzögerungschleifen mit niedrigem B tritt dieses Problem in wesentlich geringerem Umfang auf.

4.7.4 Bewertung

Werden in einem OBS-Netz verschiedene Verkehrsklassen mit unterschiedlichen QoS-Offsets verwendet, so profitieren diese von der Einführung von Puffern. Die Verlustwahrscheinlichkeit aller Klassen sinkt, allerdings für hochpriore Klassen wesentlich stärker.
Die beiden Reservierungsverfahren PostRes und PriorRes verhalten sich unterschiedlich: Während bei PostRes die Klassentrennung stets gewahrt bleibt, sollte der Offset bei PriorRes unter Berücksichtigung der Pufferverzögerung gewählt werden. Dies macht die Verwendung von PriorRes mit mehreren Verkehrsklassen komplizierter.
Durch die unterschiedlichen Offsets werden höherpriore Verkehrsklassen auch beim Zugriff auf den Puffer bevorzugt. Eine gezielte Senkung der niederprioren Verlustwahrscheinlichkeit könnte evtl. durch die Bereitstellung von eigenen Puffern für verschiedene Klassen erreicht werden. Umgekehrt ist es auch denkbar, nur bestimmte (hochpriore) Klassen zu puffern. Durch derartige Verfahren könnte dynamisch die Verlustwahrscheinlichkeit beeinflusst werden – allerdings würde dies den Verwaltungsaufwand für die Puffer weiter vergrößern.

5 Zusammenfassung und Ausblick

Optical Burst Switching ist eine sehr flexible Architektur, die spezifische Eigenschaften von WDM-Netzen gezielt ausnutzt. Für die sich abzeichnende Entwicklung hin zu photonischen Netzen stellt OBS daher mittel- bis langfristig einen sehr vielversprechenden Ansatz dar.

Im Rahmen dieser Arbeit wurden Puffermechanismen für OBS-Systeme entworfen und zwei prinzipielle Strategien für den Einsatz von Puffern entwickelt: PostRes und PriorRes. Beide Strategien wurden in Kombination mit unterschiedlichen Pufferarchitekturen durch umfangreiche Simulationen untersucht und bewertet. PostRes ist das robustere Verfahren, das auch für mehrere Verkehrsklassen völlig unkritisch funktioniert. Es eignet sich insbesondere für einfache, ggf. mehrfach durchlaufene Puffer. Mit PriorRes können unter ähnlichen Rahmenbedingungen niedrigere Burstverlustwahrscheinlichkeiten erreicht werden, und dieses Verfahren kann auch Puffer aus unterschiedlich langen Verzögerungsleitungen nutzen. Allerdings verursacht es beispielsweise deutlich größere Ende-zu-Ende-Verzögerungen. Somit haben beide Verfahren spezifische Vor- und Nachteile, und eine abschließende Bewertung ist ausgehend von den einfachen betrachteten Szenarien schwierig.

Beide Pufferstrategien sind gleichermaßen geeignet, kurzzeitige Verstopfungssituationen in den Knoten zu überwinden. Schon mit einfachen Puffern kann die Burst-Verlustwahrscheinlichkeit um mehrere Größenordnungen reduziert werden. Aufwendige Pufferstrukturen erlauben noch erheblich stärkere Verbesserungen, bedeuten aber auch einen höheren technologischen Realisierungsaufwand.

Kurzzeitige Lastspitzen können in einem OBS-Netz auch durch geeignete Aggregationsstrategien am Rand ausgeglichen werden. In den Randknoten, in denen eine elektrische Verarbeitung der Packete/Bursts vorgenommen wird, sind Puffermechanismen einfach zu realisieren, weil leistungsfähige elektrische Speichermöglichkeiten genutzt werden können. Eine Hauptaufgabe dieser Router wird deshalb sein, für eine möglichst gleichmäßige Verkehrscharakteristik im photonischen Netz zu sorgen. Gelingt dies, so sind sehr aufwendige Puffer in den Kernnetzknoten nicht notwendig. Einfache Pufferstrukturen, bestehend aus wenigen Verzögerungsleitungen, sind in vielen vorgeschlagenen Switch-Architekturen leicht zu implementieren. Deren Bereitstellung ist daher zur Gewährleistung einer hohen Dienstgüte sinnvoll.

Optical Burst Switching befindet sich momentan im Forschungsstadium. Weitere Untersuchungen sind daher auf den verschiedensten Gebieten notwendig: Sinnvoll ist insbesondere eine Betrachtung von Verkehrsszenarien, wie sie typischerweise im

Internet anzutreffen sind, d. h. mit selbstähnlichem Verkehr. Dies könnte auch weitere Erkenntnisse über die notwendige Dimensionierung von Puffern liefern.
Die Architektur der Knoten hängt stark von technologischen Randbedingungen ab. Bei der bislang vorliegenden Modellierung des Knotens und auch der Puffer wurden einige Parameter vernachlässigt, die angesichts des frühen Entwicklungsstadiums der optischen Komponenten nicht oder nur größenordnungsmäßig bekannt sind, beispielsweise Einstellzeiten. Eine verfeinerte Modellierung der Knoten ist hier sinnvoll und notwendig. Damit wäre auch eine Untersuchung weitere Puffermechanismen, z. B. mehrstufige Anordnungen, möglich.
Besonders interessante Fragestellungen ergeben sich aber bei der Verkehrsaggregation in den Randknoten, zu der bislang sehr wenige Untersuchungen vorliegen. Da eine Vielzahl an Parametern bei Optical Burst Switching entscheidend von der Verkehrscharakteristik abhängen, besteht hier noch ein erheblicher Forschungsbedarf.

A Simulationsprogramm

A.1 Überblick

Am Institut für Nachrichtenvermittlung und Datenverarbeitung existiert ein Tool zur Simulation von Optical Burst Switching-Systemen [5], mit dem bereits zahlreiche Leistungsuntersuchungen durchgeführt wurden [3, 4, 6]. Es können einzelne OBS-Knoten wie auch einfache Netze simuliert und eine Vielzahl an Parametern variiert werden, wie z. B. das Reservierungsverfahren und die verwendeten Verkehrscharakteristika. Im Rahmen dieser Studienarbeit wurde dieses Simulationstool um optische Puffermechanismen erweitert und intensive Messungen mit verschiedenen Pufferkonfigurationen durchgeführt.
Während Abschnitt A.2 näher auf die angewendete Modellierung und Architektur eingeht, wird in A.3 detailliert die Implementierung und in A.4 die Verwendung des Tools in Simulationen erläutert.
Das Simulationstool basiert auf der am IND entwickelten Simulationsbibliothek SimLib (Version 2.2). Mit Hilfe dieser objektorientierten C++-Bibliothek können ereignisgesteuerte Simulationen von Kommunikationssystemen durchgeführt werden. Die grundlegenden Konzepte und die bereitgestellten Komponenten sind in [1] dokumentiert, ergänzt durch eine Beispielanwendung in [2]. Sowohl die SimLib selbst wie auch das Tool benutzen die Component Library (Version 2.2) und die Utility Library (Version 2.2), die ebenfalls am IND entstanden sind.
Das Programm wurde auf dem Linux-Cluster des IND mit Standard-Unix-Werkzeugen entwickelt und übersetzt (g++ Compiler 2.95.3). Auf dem Linux-Cluster wurde es auch getestet und zu umfangreichen Simulationen eingesetzt.

A.2 Architektur

Zur Simulation muss für die Bestandteile eines OBS-Netzes eine geeignete Modellierung gefunden werden, die einerseits eine Abbildung auf die Konzepte der SimLib erlaubt, andererseits aber alle wesentlichen Eigenschaften von tatsächlichen Realisierungen beinhaltet. Da Verbindungsleitungen zwischen den Knoten im wesentlichen nur eine (konstante) Verzögerung verursachen und genaue Untersuchungen zur Verkehrsaggregation in Randknoten bislang nicht durchgeführt wurden, wird ein OBS-Netz im vorliegenden Tool auf eine Kette von OBS-Knoten abstrahiert. Die prinzipielle Struktur eines derartigen Knotens ist in Abbildung A.1 dargestellt.
Der Knoten besteht aus einer Vermittlungseinrichtung (Switch) und verschiedenen

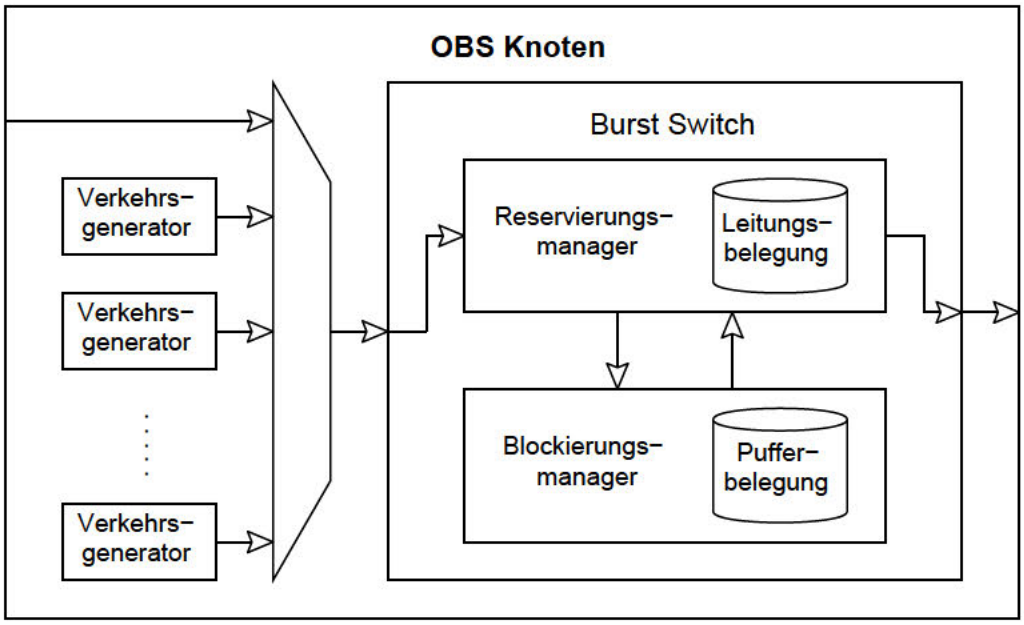

Abb. A.1: Architektur eines OBS Knotens

Verkehrsgeneratoren, die eine Simulation von extern ankommendem bzw. lokal entstehendem Verkehr ermöglichen. Nach außen existieren Schnittstellen, die auf dem Portkonzept der SimLib basieren und die eine lineare Verkettung von Knoten bzw. eine Verbindung mit einem Referenzgenerator bzw. einer Referenzsenke erlauben.
Der zentrale Burst-Switch ist untergliedert in einen Reservierungsmanager und einen Blockierungsmanager. Erreicht ein Kontrollpaket den Burst-Switch, wird zunächst die enthaltene Adress-Information ausgewertet. Da in dem linearen Modell kein echtes Routing notwendig ist, beschränkt sich der Switch darauf, Bursts, die nicht für ihn selbst bestimmt sind, an den Reservierungsmanager weiterzuleiten. Dieser verwaltet die Belegung der Ausgangsleitungen. Ist zu dem gewünschten Sendezeitpunkt ein Übertragungskanal (d. h. eine Wellenlänge) verfügbar, so wird dieser reserviert und das Kontrollpaket an den Ausgangsport und damit ggf. an einen nachfolgenden Knoten weitergeleitet.
Ist dagegen zu dem geforderten Zeitpunkt keine Leitung frei, so wird ein Burst an den Blockierungsmanager übergeben. Je nach Realisierungsform und verfügbarem Speicherplatz puffert dieser den Burst und übergibt ihn zu einem späteren Zeitpunkt wieder an den Reservierungsmanager, oder er verwirft ihn. Der Blockierungsmanager verwaltet den im Switch vorhandenen Puffer. Mit Rücksicht auf die technologische Realisierbarkeit sind dafür bislang nur Verzögerungsleitungen einer deterministischen Länge vorgesehen, deren Belegung analog zum Reservierungsmanager in einer

Datenbank gespeichert werden kann.
Es sind zwei verschiedene Typen von Verzögerungsschleifen implementiert: FDL mit unbeschränkter Kapazität und solche, die durch eine vorgegebene Anzahl an Wellenlängen nur eine beschränkte Anzahl an Bursts gleichzeitig aufnehmen kann. Die Belegung dieser beschränkten Verzögerungsleitungen erfolgt mit dem auch im Reservierungsmanager anwendbaren JET-Verfahren. Prinzipiell sind auch die anderen in [5] untersuchten Reservierungsverfahren (Just-In-Time, Horizon) auf Puffer anwendbar. Diese nutzen zwar den Puffer weniger optimal aus, sind dafür aber erheblich weniger komplex. Die Auswirkungen sind völlig analog zu den für die Leitungen im Switch bekannten. Es ist daher zu erwarten, dass mit diesen anderen Verfahren möglicherweise mehr Puffer zur Verfügung gestellt werden muss, um die gleiche Burstverlustwahrscheinlichkeit zu erreichen. JET wird – wie im Reservierungsmanager – mit Hilfe verketteter Zeitschlitz-Listen implementiert. Prinzipiell sind hier aber auch andere Verfahren denkbar, beispielsweise die in [16] vorgeschlagene Verwendung von Bäumen.
Diese Modellierung erlaubt es, die prinzipiellen Eigenschaften von Switch- und Pufferarchitekturen zu untersuchen. Eine detaillierte Behandlung von konkreten (technologienahen) Realisierungsvorschlägen für OBS-Switche (z. B. Beneš-Netzwerke in [16]) ist allerdings nicht möglich. Einschränkungen beispielsweise hinsichtlich Architekturen mit mehreren Vermittlungseinrichtungen oder der Verwendung sogenannter „hybrider Puffer“ mit variablen Verzögerungszeiten könnten durch Ergänzungen überwunden werden.
Abschließend sei bemerkt, dass zwischen einer Ausgangsleitung im Switch und einer Verzögerungsleitung im Puffer viele Gemeinsamkeiten bestehen. Praktisch handelt es sich in beiden Fällen um Lichtwellenleiter mit einer deterministischen Länge und einer bestimmten Anzahl an Wellenlängen. Aufgrund der im Tool vorgegebenen Strukturen, deren Modifikation den Rahmen dieser Arbeit gesprengt hätte, existiert für den Reservierungsmanager und die Verzögerungsleitungen im Blockierungsmanager eine getrennte, aber ähnliche Implementierung. Eine elegante Alternative wäre die Einführung einer gemeinsamen Basisklasse für reservierbare Objekte, die in der Lage ist, Belegungsinformationen zu verwalten (also z. B. das JET-Verfahren beherrscht) und von der Klassen für Ausgangs- bzw. Verzögerungsleitungen abgeleitet werden können. Dies würde auch die Erstellung von Statistiken über die Anzahl übertragener Bursts, die Wellenlängenbelegung usw. vereinfachen, die bisher für beide Fälle getrennt implementiert sind.

A.3 Implementierung

Die im Tool existierenden Klassen und deren Funktionsweise wird nachfolgend mit Hilfe der *Unified Modeling Language* (UML) vorgestellt. Dabei wird nur auf die Klassen eingegangen, die neu erstellt bzw. erheblich gegenüber dem Tool aus der Vorgängerarbeit modifiziert wurden. Diese beschränken sich im Wesentlichen auf den Reservierungsmanager. Für eine Beschreibung des übrigen Tools sei auf [5] verwiesen. Die zum Simulationsprogramm gehörenden Klassen beginnen stets mit dem Anfangsbuchstaben `C`, während ein `T` für Komponenten aus der SimLib steht.

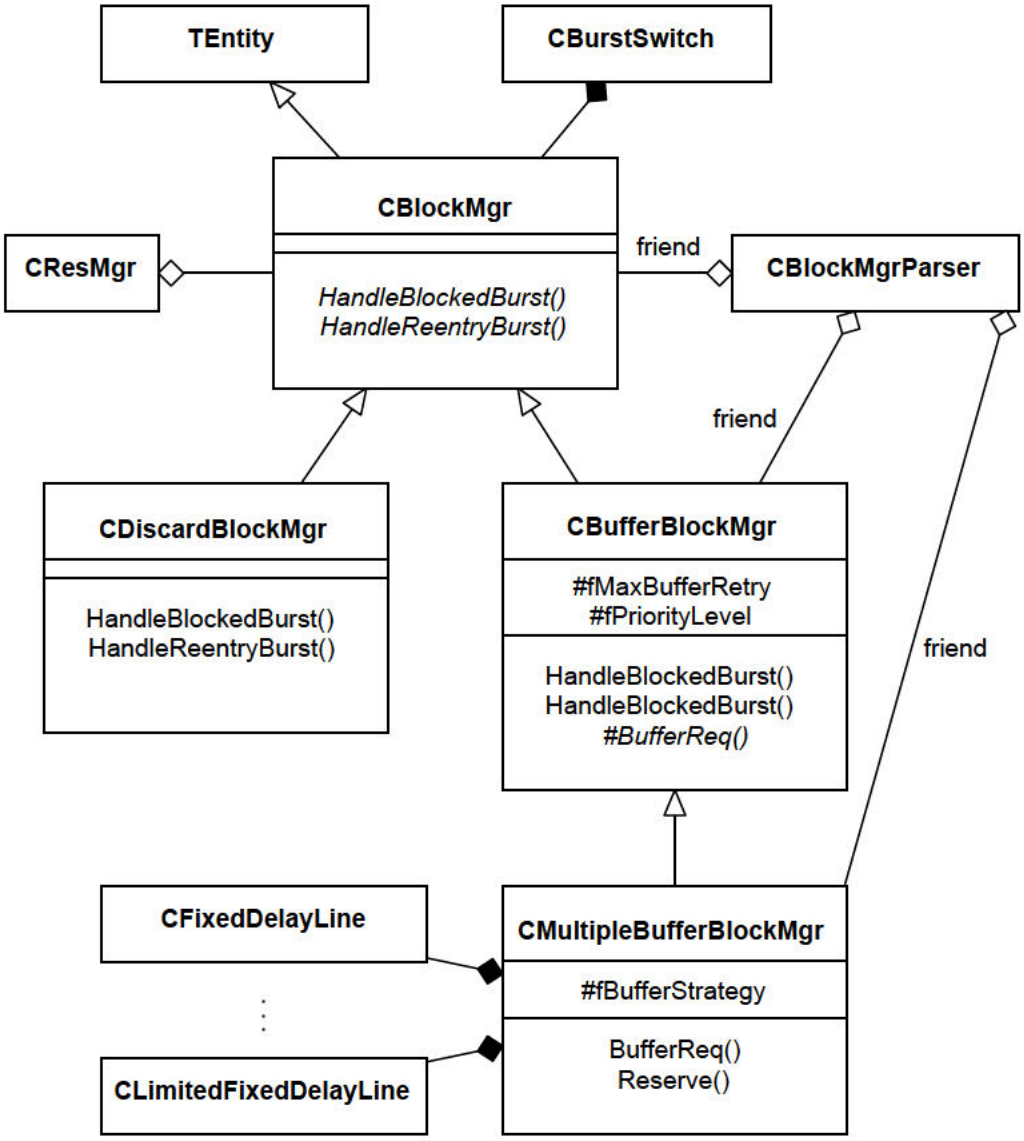

Abb. A.2: Klassendiagramm für `CBlockMgr`

Für den Blockierungsmanager existiert eine abstrakte Basisklasse `CBlockMgr`, die für die Schnittstelle zum Reservierungsmanager `CResMgr` die beiden rein virtuellen Me-

thoden `HandleBlockedBurst` und `HandleReentryBurst` vorsieht (vgl. Abbildung A.2). Funktionsfähige Blockierungsmanager sind Kindklassen von `CBlockMgr` und implementieren diese beiden Funktionen. Die konkrete Klasse `CDiscardBlockMgr` ist für den pufferlosen Fall vorgesehen und verwirft alle Bursts, die zur Behandlung übergeben werden („Discard“). In `CBufferBlockMgr` ist die Schnittstelle für den Fall *mit* Puffern implementiert und es sind auch weitere Eigenschaften eines Puffers verwirklicht, die unabhängig von dessen konkreter Strukur sind, wie z. B. die Möglichkeit, den Puffer mehr als einmal zu durchlaufen (*feed back*) oder nur bestimmte (hochpriore) Klassen zu puffern. Konkret realisiert wird ein Puffer im Tool durch `CMultipleBufferBlockMgr`. Dieser enthält eine linear verkettete Liste aus Verzögerungsleitungen und verwaltet diese mit den Reservierungsverfahren PriorRes, PostRes oder PostResRandom, die bereits in Abschnitt 3.3.2 vorgestellt wurden. Durch die Möglichkeit, theoretisch beliebig viele FDLs zu verwenden, sind mit diesem relativ „einfachen“ Puffermanager sehr viele Strukturen behandelbar. Er wurde in allen Pufferuntersuchungen verwendet. Wird `CMultipleBufferBlockMgr` ohne Verzögerungsleitung verwendet, verhält er sich wie `CDiscardBlockMgr`. Letzterer ist aber sinnvoll, um das Simulationstool abwärtskompatibel zu halten.
Die Einführung der Zwischenklasse `CBufferBlockMgr` ermöglicht es aber auch, im Zuge von Erweiterungen des Tools noch andere Blockierungsmanager zu implementieren, die z. B. andere Pufferstrukturen oder noch komplexere Pufferstrategien verwenden könnten.
Das Einlesen der Einträge in der Parameterdatei, die den Blockierungsmanager betreffen, erledigt `CBlockMgrParser`, der eine friend-Klasse von `CBlockMgr` und einiger abgeleiteter Klassen ist. Das Bereitstellen eines gemeinsamen Parsers stellt einen gewissen Kompromiss dar: Einerseits erlaubt dieser, an zentraler Stelle alle Einträge aus der Parameterdatei auszulesen, andererseits sind dadurch Typ-Überprüfungen notwendig. Diese wurden durch eine Vergabe von Typ-IDs gelöst.
Das UML-Klassendiagramm für Verzögerungsleitungen ist in Abbildung A.3 dargestellt. Auch hier existiert eine abstrakte Basisklasse `CDelayLine`. Neben der Schnittstelle enthält diese die Anzahl der Wellenlängen in dieser FDL. Von `CDelayLine` abgeleitet ist eine Klasse mit unbegrenzter Pufferkapazität und konstanter Länge (`CFixedDelayLine`). Diese kümmert sich nicht um die Pufferbelegung, sondern ist in der Lage, beliebig viele Bursts aufzunehmen und zu verzögern. Für diesen Fall ist die Anzahl der enthaltenen Wellenlängen nur für die Erstellung von Statistiken relevant. In `CLimitedDelayLine` werden dagegen einzelne Wellenlängen reserviert. Dafür wird das auch in `CResMgr` verwendete JET-Verfahren verwendet. Andere Ver-

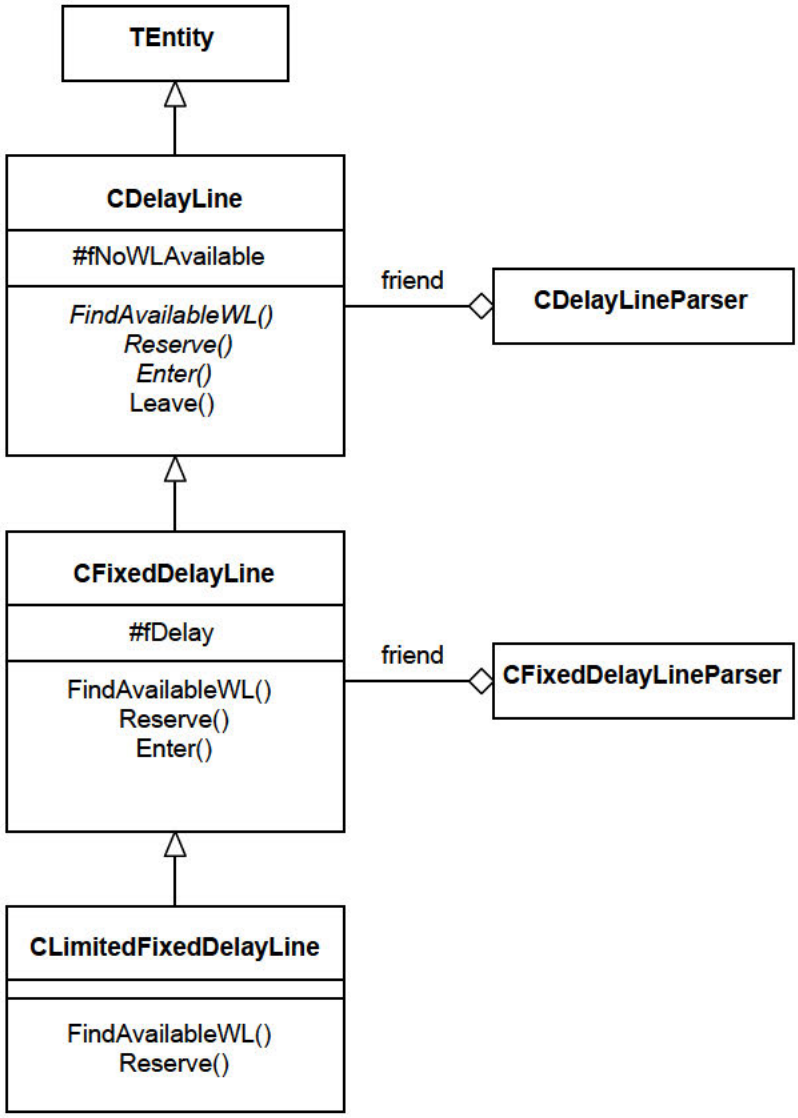

Abb. A.3: Klassendiagramm für `CDelayLine`

fahren könnten – wie schon erwähnt – auch implementiert werden, würden aber sicherlich keine wesentlich neuen Erkenntisse liefern.
Die Klassenhierarchie ist so gewählt, dass es leicht möglich ist, das Tool um weitere FDL-Typen zu erweitern. Daher wurden hier auch – trotz des größeren Aufwands – getrennte Parserklassen programmiert. Insbesondere kann durch weitere Kindklassen von `CDelayLine` und durch die Verwendung von Verteilungsfunktionen aus der SimLib die Beschränkung auf eine konstante Zeitverzögerung überwunden werden. Für einen einzelnen Lichtwellenleiter ist dies technologisch kaum zu rechtfertigen, trotzdem wäre es durchaus interessant: Einerseits erhielte man für eine negativ-exponenziell verteilte Verzögerung ein gedächnisfreies System, dessen mathematische Analyse möglicherweise einfacher wäre. Andererseits wäre dies eine denkbare Möglichkeit, sehr komplexe Pufferstrukturen mit unterschiedlichen Verzögerungszeiten, zu der z. B. auch die „hybriden“ Puffer zählen, zu modellieren. Dies könnte daher ein Ansatzpunkt für weitere Untersuchungen sein.

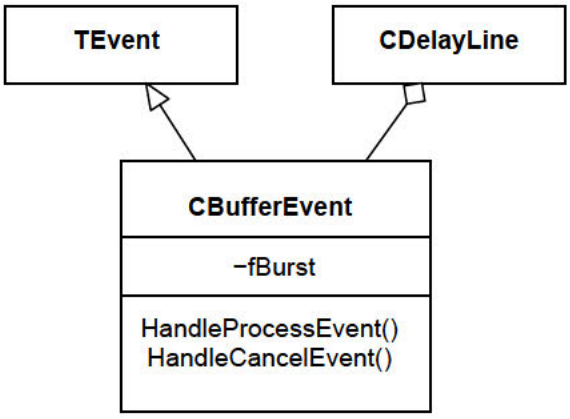

Abb. A.4: Klassendiagramm für `CBufferEvent`

Die eigentliche Pufferung der Bursts wird durch eine von `TEvent` abgeleitete Klasse `CBufferEvent` übernommen (siehe Abbildung A.4). Der in der Simulation vorhandene Kalender ruft die `HandleProcessEvent()`-Methode dieses Ereignisses zu einem bestimmten Zeitpunkt auf. Dieser wird beim Eintragen so gewählt, dass sich die Verzögerung b gibt. Nachdem der Burst durch `HandleProcessEvent()` weitergegeben wurde, löscht sich das Ereignis selbst.
Abgesehen von den hier dokumentierten Klassen wurden auch geringfügige Modifikationen in `CResMgr` und `CBurstSwitch` vorgenommen, die sich aber fast ausschließlich auf die Schnittstelle zum Blockierungsmanager konzentrieren.
Der konkrete Ablauf einer (erfolgreichen) Pufferung ist im Sequenzdiagramm A.5

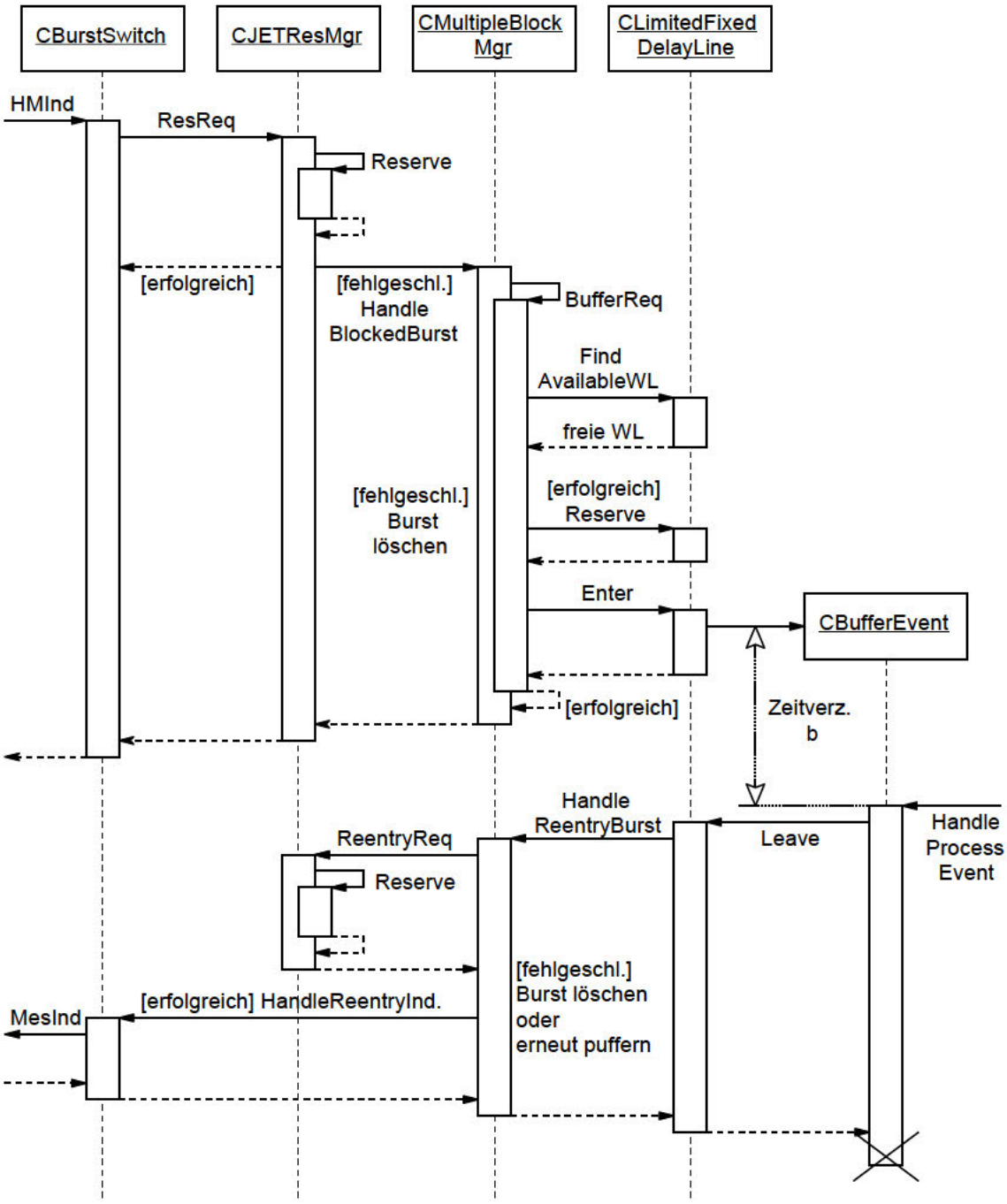

Abb. A.5: Sequenz-Diagramm (PostRes-Fall)

veranschaulicht: Ein im Burst-Switch ankommender Burst (genauer gesagt: das zugehörige Kontrollpaket) verursacht eine Reservierungsanforderung `ResReq` an den Reservierungsmanager. Schlägt diese fehl, wird über `HandleBlockedBurst` der Blockierungsmanager aufgerufen. Dessen Vorgehensweise hängt vom Reservierungsverfahren ab. Für den dargestellten PostRes-Fall wird untersucht, ob auf einer der Verzögerungsleitungen Platz zur Verfügung steht und falls ja, wird der Burst an diese mit `Enter` übergeben. Andernfalls wird der Burst verworfen. Für PriorRes wird in einer zusätzlichen Abfrage die Ausgangsleitung reserviert.
Die Verzögerungsleitung verwendet zur Verzögerung des Bursts ein Ereignis des Typs `CBufferEvent`. Dieses ruft beim Eintreten des Ereignisses wiederum deren `Leave`-Methode auf, die den Burst an den Blockierungsmanager weiterleitet. Dieser versucht, einen freien Zeitschlitz auf dem Ausgangslink zu reservieren (nicht bei PriorRes). Im Fall eines *feed back*-Puffers wird ggf. auch eine weitere Pufferung ermöglicht.
Insgesamt ist das Design so gewählt, dass die Funktionen des Reservierungs- und Blockierungsmanagers völlig von der konkreten Aufgabe der Zeitverzögerung durch die Verzögerungsleitung getrennt sind.

A.4 Bedienung

Das Simulationstool wird von der Kommandozeile gestartet. Dabei müssen die Parameter- (Standardeinstellung: `obs.par`), Format- (`obs.fmt`), Ergebnis- und Parserdatei angegeben werden. An die Formateinstellungen werden keine besonderen Anforderungen gestellt, im wesentlichen kann die in [2] verwendete Datei übernommen werden. Sinnvoll für eine automatisierte Auswertung ist allerdings, die globalen Namen der Statistiken innerhalb der PrintServer-Hierarchie auszugeben. Auf den folgenden Seiten ist eine dokumentierte Parameterdatei abgedruckt, die fast alle verfügbaren Optionen ausnutzt:

```
1 {
2   WLperLink = 8; // Anzahl der Wellenlaengen je Ausgangsleitung im Switch
3   LineRateBps = 1250000000; // Datenrate auf einer WL (in bytes/s), hier 10 Gbit/s
4   GuardTime = 0.0; // Guard Time (in s)
5
6   // Parameter fuer Statistiken in Abhaengigkeit der Burstlaenge (optional)
7   DiffStatMinLength = 0;
8   DiffStatMaxLength = 87500;
9   DiffStatNoOfBuckets = 10;
```

```
10  DiffStatMeanLength = 25000;
```

Globale Parameter sind die Anzahl der Wellenlängen je Ausgangsleitung und die Datenübertragungsrate. Diese Rate wird auch für die Verzögerungsleitungen verwendet, was sicher keine signifikante Einschränkung darstellt. Eine sinnvolle Wahl der Statistikparameter hängt stark von den übrigen Parameterwerten ab. Neben der angegebenen Statistik existieren weitere, die Eingaben einlesen können; diese wurden aber bei Simulationen nie verwendet.

```
11  Node {
12    Batches = 10; // Anzahl der Teiltests
13    Calls = 100000; // Anzahl der Bursts je Teiltest
14    NoOfTrGen = 1; // Anzahl der Verkehrsgeneratoren in diesem Knoten
15
16    TrafficGenerator {
17      OBSSubTrafficGenerator { // hochpriore Klasse (class 0)
18        NoOfSources = 30; // 30 Einzelgeneratoren, d.h., 30% Anteil
19        FirstSourceId = 1;
20        Group = 0; // Nr. der Verkehrsklasse
21
22        // Startwerte fuer die generierten Bursts
23        Destination = 1;
24        Type = 0;
25        PrevWL = 0;
26
27        // QoS-Offset (in s)
28        //   hier 2* mittlere Burstlaenge:
29        OffsetTime = 2e-05;
30
31        // Laengenverteilung der Bursts (in byte)
32        //   hier negativ-exponenziell mit Mittelw. 100 kbit = 12.5 kbyte = 1e-5s
33        LengthDist { BaseDist {NegExp { Mean = 12500; }; }; }
34
35        // Verteilung der Zwischenankunftszeiten (in s)
36        //   hier fuer Last 0.8 und insgesamt 100 Einzelgeneratoren
37        GenDist { NegExp { Mean = 0.00015625; }; }
38      }
39
40      OBSSubTrafficGenerator { // niederpriore Klasse (class 1)
41        NoOfSources = 70; // 70 Einzelgeneratoren, d.h., 70% Anteil
42        FirstSourceId = 31;
43        Group = 1;
44        Destination = 1;
```

```
45          Type = 0;
46          PrevWL = 0;
47          OffsetTime = 0; // niederpriore Klasse ohne Offset
48          LengthDist { BaseDist { NegExp { Mean = 12500; }; }; }
49          GenDist { NegExp { Mean = 0.00015625; }; }
50        }
51      }
```

Im hier angenommenen Verkehrsszenario existiert ein OBS-Knoten mit 100 lokalen Verkehrsgeneratoren. 30 % des erzeugten Verkehrsanteil entfallen auf eine hochpriore Verkehrsklasse, die gegenüber der niederprioren über einen zusätzlichen QoS-Offset $\delta_{\mathrm{H}} = 2\,h$ verfügt. Die Verteilung der Burstlängen ist jeweils negativ-exponenziell und es werden Poisson-Ankünfte bei einer Last von $\rho = 0.8$ angenommen. Eine detailliertere Beschreibung der Parameter ist in [5] zu finden.

```
52      BurstSwitch {
53        NoOfClasses = 2; // Anzahl an Klassen
54        ProcDelay = 0.0; // Verarbeitungsdauer des Kontrollpakets im Knoten (optional)
55        ResMgrType = "JET"; // Reservierungsverfahren im Switch
56        BlockMgrType = "MultipleBuffer"; // Typ des Blockierungsmanagers
57
58        ResMgr {
59          InitialWLType = "Zero"; // Beginn der Linkabsuche (optional)
60
61          // Parameter fuer Verteilstatistiken ueber nicht belegte Laengen auf
62          // den Wellenlaengen (optional)
63          LengthDistrSlots = 20;
64          LengthDistrLower = 0;
65          LengthDistrUpper = 2e-05;
66        }
```

In Zeile 56 wird der zu verwendende Blockierungsmanager angegeben. Es sind die beiden in Tabelle A.1 angegebenen Parameterwerte möglich. Im Fall eines `Discard`-Blockiermanagers kann der `BlockMgr`-Block leer bleiben, ansonsten sollten ggf. die nachfolgenden Angaben gemacht werden:

```
67        BlockMgr {
68          // maximale Group-Nr., die noch gepuffert werden darf (optional)
69          //   Standardeinstellung: Alle Klassen puffern
70          //   hier: nur Klasse 0 (hochprior) wird gepuffert
71          PriorityLevel=0;
72
```

```
73          // maximale Anzahl an zusaetzlichen Pufferdurchlaeufen (optional)
74          //     Standardeinstellung: 0 (feed forward)
75          //     hier: max. 3 zulaessige Pufferdurchlaeufe
76          MaxBufferRetry=2;
77
78          // verwendete Reservierungsstrategie (optional)
79          //      Standardeinstellung: PostRes
80          BufferStrategy="PostRes";
81
82          // Liste der vorhandenen Verzoegerungsleitungen (optional)
83          LimitedFixedDelayLine { // beschraenkte Kapazitaet
84            WL=2; // 2 Wellenlaengen vorhanden
85            Delay=1e-05; // Verzoegerung um eine mittlere Burstlaenge
86          }
87          FixedDelayLine { // unbeschraenkte Kapazitaet
88            WL=4; // 4 Wellenlaengen (nur zur Berechnung der Auslastung benoetigt)
89            Delay=2e-05; // Verzoegerung um zwei mittlere Burstlaengen
90          }
91        }
92      }
93    }
94  }
```

`PriorityLevel` bezieht sich auf die im Verkehrsgenerator angegebene `Group`-Nummer. Nur Bursts, deren Kennung kleiner oder gleich dieses Schwellwerts ist, werden gepuffert, andere Bursts werden sofort nach Eintritt in den Blockierungsmanager verworfen. Damit besteht eine weitere Möglichkeit, bestimmte Bursts gezielt zu bevorzugen. Die Auswirkungen dieses Parameters wurden aber bisher nicht untersucht. Hier wären auch noch weitere Ansätze denkbar, z. B. für jede Verkehrsklasse eigene Puffer vorzusehen. Dies würde aber einen erheblichen Mehraufwand bedeuten.

Standardmäßig wird ein *feed forward*-Puffer angenommen (`MaxBufferRetry`=0). Eine *feed back*-Struktur kann durch Erhöhung dieses Wertes erreicht werden.

In `BufferStrategy` wird anhand der in Tabelle A.2 angegebenen Werte ausgewählt, welche Pufferstrategie verwendet wird. Im Zusammenhang mit `PriorRes` spielt der Wert von `MaxBufferRetry` keine Rolle, da – wie erläutert – PriorRes nur für eine einmalige Pufferung definiert wurde.

Die im Blockierungsmanager-Block angegebenen Verzögerungsleitungen werden in der Reihenfolge der Nennung verwaltet. Bei Verwendung von verschieden langen Leitungen ist zur Minimierung der Verzögerungszeit daher i. Allg. eine Sortierung

nach zunehmender Länge sinnvoll.

Parameterwert	Beschreibung
`Discard`	alle Bursts verwerfen
`MultiBuffer`	puffern (je nach vorhandenem Speicher)

Tabelle A.1: Parameterwerte für `BlockMgrType`

Parameterwert	Beschreibung
`PriorRes`	PriorRes-Verfahren
`PostRes`	PostRes mit sequenzieller Absuche der FDL
`PostResRandom`	PostRes mit zufälliger Auswahl einer FDL

Tabelle A.2: Parameter für `BufferStrategy`

Typ	Beschreibung
`FixedDelayLine`	unbeschränkte Kapazität
`LimitedFixedDelayLine`	JET-Reservierungsverfahren

Tabelle A.3: Übersicht über implementierte FDL-Typen

B Optimiertes JET-Verfahren

Das in der Simulation verwendete Standard-JET-Verfahren beruht auf dem Prinzip, nach einem verfügbaren Wellenlängenkanal zu suchen und den ersten gefundenen auszuwählen. Weitere Wellenlängen werden dann nicht getestet. Im nachfolgenden Abschnitt wird ein JET-Verfahren betrachtet, das im Gegensatz dazu auch die Reservierungssituation der übrigen Wellenlängen berücksichtigt und versucht, die Belegung zu optimieren.

B.1 Theoretische Überlegungen

Bei Verwendung von Bursts mit verschiedenen Offsets lässt es sich nicht vermeiden, dass der Wellenlängenkanal „zerstückelt“ wird und Zwischenräume entstehen, die zu klein sind, um noch einen Burst aufzunehmen. Ein denkbarer Ansatz zur Verbesserung der Auslastung des Kanals und damit zur Senkung der Burstverlustwahrscheinlichkeit ist, diese Lücken möglichst klein zu machen. In [17] wird gezeigt, dass unter bestimmten Voraussetzungen (dort: selbstähnlicher Verkehr) eine Verbesserung erreicht werden kann, wenn zunächst alle freien Kanäle betrachtet werden und dann derjenige Kanal ausgewählt wird, bei welchem die geringste Lücke entsteht.
Zur Minimierung der Burstzwischenräume sind verschiedene Strategien denkbar. Der naheliegenste Ansatz ist, die Distanz zu dem jeweiligen Vorgängerburst auf einer Wellenlänge zu messen und dann diejenige Wellenlänge auszuwählen, bei der diese minimal ist. Bei starker Fragmentierung des Wellenlängenkanals sind aber auch kompliziertere Verfahren denkbar, z. B. auch die Berücksichtigung der Entfernungen zu bereits reservierten späteren Bursts. In der nachfolgenden Untersuchung wurde nur der erste Ansatz verfolgt, da dieser sich am leichtesten realisieren lässt. D. h. die Distanz d zwischen dem Übertragungsende des vorherigen Bursts und dem Beginn des einzufügenden Bursts wird minimiert.
Für Bursts mit gleich langem Offset spielt die Distanz d keine Rolle. Stehen einem Burst mehr als eine mögliche Wellenlänge zur Verfügung, so kann ein nachfolgendes Kontrollpaket stets die andere reservieren, unabhängig davon, welche der erste gewählt hat. Schließlich können keine Reservierungen vorliegen, die weiter in die Zukunft reichen und die Wellenlängen fragmentieren. Existieren aber Bursts mit unterschiedlichen Offsets, kann sich das Reservierungsverfahren auf nachfolgende Bursts auswirken. Ein solches Szenario ist in Abbildung B.1 dargestellt.
Vorausgesetzt wird der Fall, dass eine hochpriore und eine niederpriore Klasse mit Offsets δ_{H} bzw. δ_{L} ($\delta_{\mathrm{H}} > \delta_{\mathrm{L}}$) vorliegt. Einem eintreffenden Kontrollpaket $\mathrm{req}_{\mathrm{H}}$ stehen

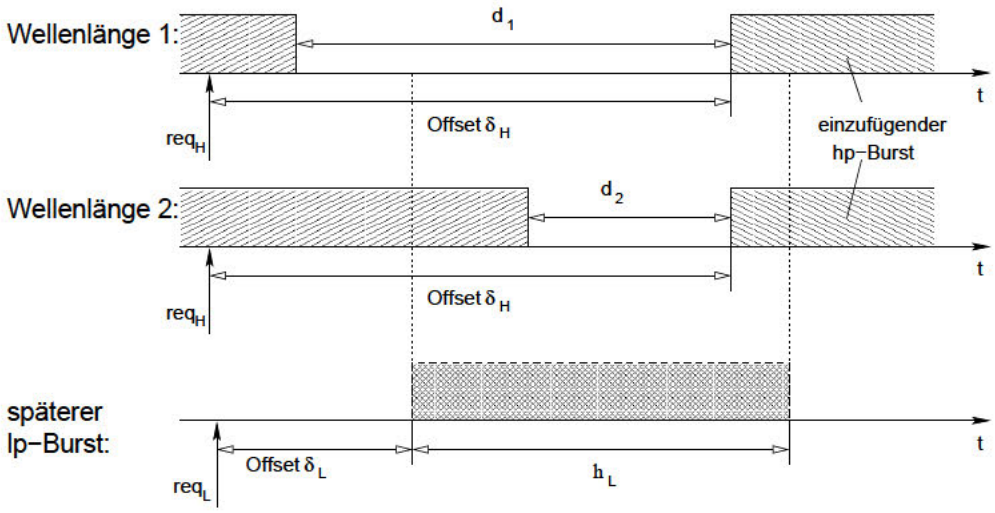

Abb. B.1: Szenario für den Einsatz von JETOpt

zwei mögliche Wellenlängenkanäle 1 und 2 mit der Distanzen $d_1 > d_2$ zum jeweiligen Vorgängerburst zur Verfügung. Das Standard-JET-Verfahren würde in diesem Fall willkürlich eine der beiden Wellenlängen belegen, je nach dem, welche zuerst getestet wird, während im optimierten Verfahren Wellenlänge 2 gewählt würde.
Je nach Wahl der Wellenlänge findet ein nachfolgender Burst eine unterschiedliche Belegungssituation vor. Interessant ist der Fall, wenn nur bei einer der beiden möglichen Belegungen ein weiterer (niederpriorer) Burst mit der Länge h_L einen freien Zeitschlitz findet. Dieser Fall ist in Abbildung B.1 dargestellt. Eine infinitesimal kurz nach $\mathrm{req_H}$ folgende Reservierungsanforderung $\mathrm{req_L}$ passt nicht auf die durch $\mathrm{req_H}$ reservierte Wellenlänge, falls gilt:

$$\delta_\mathrm{L} + h_\mathrm{L} > \delta_\mathrm{H} \tag{B.1}$$

In diesem Fall überschneidet sich das Ende des niederprioren Bursts mit dem bereits reservierten hochprioren Burst. Mit der Annahme $\delta_\mathrm{L} = 0$ folgt aus (B.1):

$$h_\mathrm{L} > \delta_\mathrm{H} \tag{B.2}$$

Tritt der in Abbildung B.1 dargestellte Fall ein, dass der einzufügende niederpriore Burst so lange ist und so früh anfängt, dass er nicht in die kleinere der beiden Lücken passt, so wird der niederpriore Burst verworfen, falls der hochpriore Burst Wellenlänge 1 gewählt hat, während er bei einer Reservierung des hochprioren Bursts auf Wellenlänge 2 übertragen werden kann. Würden dagegen durch eine weitere vorherige Reservierungsanforderung eines hochprioren Bursts beide Wellenlängen

in der in Abbildung B.1 dargestellten Art und Weise belegt sein, so müsste der niederpriore Burst verworfen (oder gepuffert) werden, unabhängig von der Frage, ob optimiert wurde oder nicht.
Aus diesen prinzipiellen Überlegungen folgt also, dass insbesondere für Offset-Werte der hp-Bursts, die kleiner als die Burstlänge der lp-Bursts sind, das optimierte JET-Verfahren zu geringeren Verlusten für die niederpriore Verkehrsklasse führen sollte. Die hochprioren Bursts sollten dagegen praktisch unbeeinflusst bleiben.

B.2 Implementierung

Im Simulationstool wird das JET-Verfahren durch den Reservierungsmanager `CJETResMgr` realisiert. Dieser wurde durch Ableitung durch einen leicht modifizierten Typ `CJETResMgrOpt` ergänzt, in dem in der überschriebenen Methode `FindAvailableWL()` das Optimierungsverfahren implementiert ist. Es wird verwendet, wenn in der Parameterdatei der Eintrag `ResMgrType = "JETOpt"` gesetzt wird. Zusätzlich ist innerhalb des Parameterblocks des Reservierungsmanagers ein (optionales) Feld `OptStrategy` vorgesehen, um verschiedene Optimierungsverfahren auszuwählen. Dem angesprochen Verfahren entspricht das Setzen dieses Feldes auf `MinPriorGap`.

B.3 Untersuchungsergebnisse

Das optimierte JET-Verfahren wurde durch Variation verschiedener Verkehrsparameter untersucht, wobei insbesondere die Frage im Vordergrund stand, unter welchen Bedingungen es gegenüber dem Standardverfahren Vorteile bietet.
Abbildung B.2 zeigt die Ergebnisse für den pufferlosen Fall mit den gleichen Parametern wie in Abschnitt 4.7.2, d.h. eine hochpriore (hp) und eine niederpriore (lp) Verkehrsklasse (Anteil des hp-Verkehrs 30 %) mit negativ-exponenziell verteilten Ankunftszeiten, 8 Wellenlängen und einer Auslastung von 0.8. Mit zunehmendem QoS-Offset wird in beiden Fällen eine deutliche Verkehrstrennung erreicht. Für negativ-exponenziell verteilte Burstlängen mit $h_0 = E[h_\mathrm{L}] = E[h_\mathrm{H}]$ (im Bild mit „neg-exp“ gekennzeichnet) tritt ab einem Offset von ca. 1 (bezogen auf h_0) eine deutliche Sättigung ein. Das optimierte Verfahren liefert - wie erwartet - insbesondere für $\delta_\mathrm{H} \approx h_0$ leichte Vorteile für die niederpriore Klasse, während die Verlustwahrscheinlichkeit für die hp-Bursts leicht ansteigt. Letzteres ist durch die gestiegene Belegung des Kanals erklärbar. So können durch die Optimierung u. a. sehr lange lp-Bursts zugelassen werden, die dann hochpriore Bursts blockieren können. Die

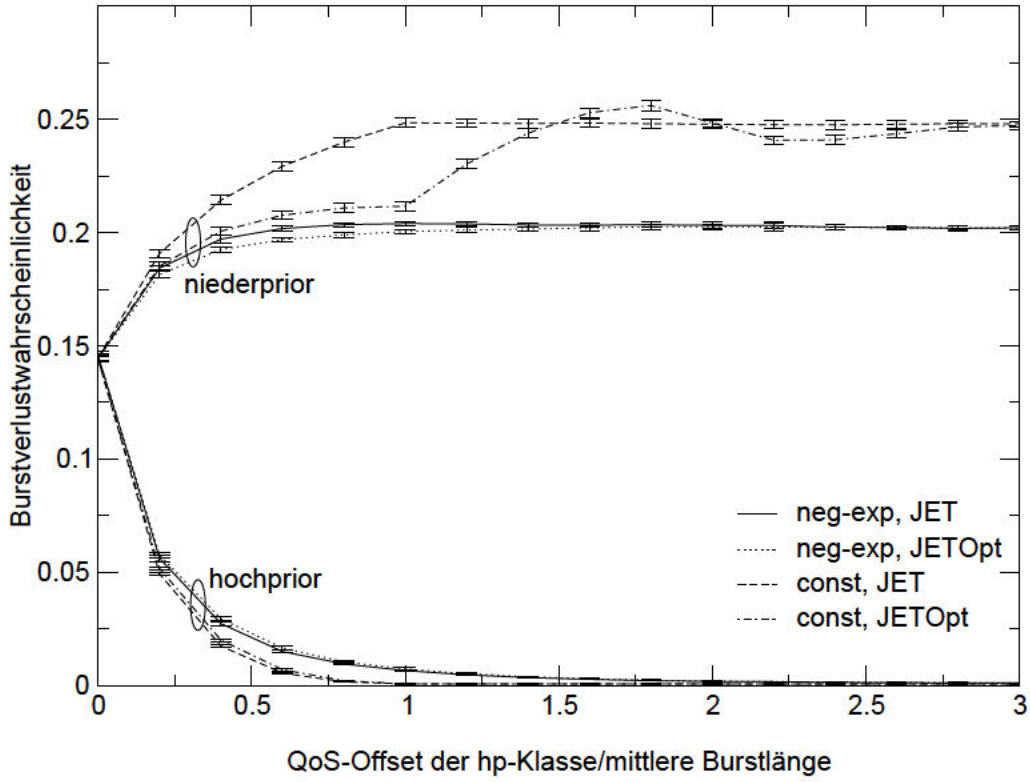

Abb. B.2: Vergleich von JET und JETOpt

Veränderungen sind aber nur sehr gering.

Um ein Szenario zu schaffen, in dem JETOpt deutliche Vorteile aufzeigen sollte, d. h. um die durch Gleichung (B.2) vorgegebene Randbedingung leichter einhalten zu können, wurde in weiteren Simulationen die Länge der niederprioren Bursts auf konstant h_{L} gesetzt („const“ in Abbildung B.2).

Dabei nehmen die Verluste der hochprioren Klasse deutlich ab. Wie in [4] gezeigt wird, hängt die Verlustwahrscheinlichkeit der hp-Klasse von der Verteilung der Burstlänge der niederprioren Klasse ab, nicht aber ihrer eigenen (die daher negativ-exponenziell bleiben kann). Die Verluste der niederprioren Verkehrsklasse sind größer als im negativ-exponenziellen Fall, und die beiden Reservierungsstrategien weichen deutlich voneinander ab: Bis zu einem Offset von ca. 1,5 ist das optimierte Verfahren deutlich besser, für größere Werte tritt aber abschnittsweise sogar das Gegenteil ein. Besonders auffällig ist der deutliche Knick bei einem Offsetwert, der der mittleren Burstlänge entspricht. Dieses Verhalten bestätigt die obigen Überlegungen.

Anzumerken ist, dass bei Untersuchungen mit konstanten Burstlängen diesem Wert eine besondere Bedeutung zukommt. Auch in anderen Messungen zeigen sich hier Effekte, die von den Szenarien mit variablen Burstlängen signifikant abweichen. Ein

derartiges Beispiel, das aber nicht weiter verfolgt wurde, ist eine Pufferverzögerung um genau eine Burstlänge.
Zur Untersuchung von JETOpt wurden weitere Untersuchungen durchgeführt, u. a. auch in Kombination mit Puffern. Das optimierte JET-Verfahren liefert in den meisten untersuchten Fällen geringfügig niedrigere Verlustwahrscheinlichkeiten, die Abweichungen sind aber stets marginal. In [17] wurden erheblich deutlichere Effekte beobachtet, ein direkter Vergleich ist aber nicht möglich, da keine Messungen mit selbstähnlichem Verkehr durchgeführt wurden.

B.4 Bewertung

Die untersuchte optimierte Version des JET-Verfahrens führt im Vergleich zum „normalen“ JET meist zu niedrigeren Verlustwahrscheinlichkeiten, da die Wellenlängen besser ausgenutzt werden. In den Messungen wurde allerdings keine Unterschiede festgestellt, die angesichts des erheblichen Mehraufwandes bei der Implementierung eine Verwendung von JETOpt rechtfertigen würde. Zur abschließenden Bewertung des optimierten JET-Verfahrens sind aber weitere Untersuchungen notwendig – insbesondere mit selbstähnlichem Verkehr.

Literatur

[1] Bodamer, Stefan: *Einführung in die Konzepte der C++-Simulationsbibliothek.* Stuttgart: IND, 1999

[2] Bodamer, Stefan ; Sallinger, Stefan: *Beispiel zur Einführung in die Konzepte der C++-Simulationsbibliothek.* Stuttgart: IND, 1999

[3] Dolzer, Klaus ; Gauger, Christoph. *On Burst Assembly in Optical Burst Switching Networks – A Performance Evaluation of Just-Enough-Time.* Proceedings of the 17th International Teletraffic Congress, Salvander/Brazil. September 2001

[4] Dolzer, Klaus ; Gauger, Christoph ; Späth, Jan ; Bodamer, Stefan: Evaluation of Reservation Mechanisms for Optical Burst Switching. In: *International Journal of Electronics and Communications (AEÜ)* 55 (2001), Nr. 1. – LDB [33135]

[5] Gauger, Christoph: *Untersuchung von Reservierungsverfahren für Optical Burst Switching*, IND, Universität Stuttgart, Diplomarbeit, September 2000

[6] Gauger, Christoph ; Dolzer, Klaus ; Späth, Jan ; Bodamer, Stefan: Service Differentiation in Optical Burst Switching Networks. In: *Beiträge zur 2. ITG Tagung Photonische Netze, Dresden* (2001), März. – LDB [33141]

[7] Hunter, David K. ; Chia, Meow C. ; Andonovic, Ivan: Buffering in Optical Packet Switches. In: *Journal of Lightwave Technology* 16 (1998), Dezember, Nr. 12. – LDB [31191]

[8] Kühn, Paul J. *Communication Networks I.* Vorlesungsskript. Wintersemester 2000/2001

[9] Kühn, Paul J. *Teletraffic Theory and Engineering.* Vorlesungsskript. Wintersemester 2000/2001

[10] Listanti, Marco ; Eramo, Vincenzo ; Sabella, Roberto: Architectural and Technological Issues for Future Optical Internet Networks. In: *IEEE Communications Magazine* (2000), September, S. 82–92. – LDB [32730]

[11] Mathy, Laurent ; Edwards, Christopher ; Hutchison, David: The Internet: A Global Telecommunication Solution. In: *IEEE Network* (2000), Juli/August, S. 46–57. – LDB [32481]

[12] Qiao, C. ; Yoo, M.: Optical Burst Switching (OBS) – A new Paradigm for an Optical Internet. In: *Journal of High Speed Networks* 8 (1999), S. 69–84. – LDB [29795]

[13] Qiao, C. ; Yoo, M.: Choices, Features an Issues in Optical Burst Switching. In: *Optical Networks Magazine* 1 (2000), April, Nr. 2, S. 36–44. – LDB [32599]

[14] Rosen, E. ; Callonn, A. V. *Multiprotocol Label Switching Architecture.* RFC 3031. Januar 2001

[15] Trucker, Rodney S. ; Zhong, Wen D.: Photonic Packet Switching: An Overview. In: *IEICE Transactions on Communications* E82-B (1999), Februar, Nr. 2, S. 254–264. – LDB [30243]

[16] Turner, Jonathan S.: Terabit Burst Switching. In: *Journal of High Speed Networks* 8 (1999). – LDB [29791]

[17] Xiong, Yijun ; Vandenhoute, Marc ; Cankaya, Hakki C.: Control Architecture in Optical Burst-Switched WDM Networks. In: *IEEE Journal on Selected Areas in Communications* 18 (2000), Oktober, Nr. 10. – LDB [33267]

[18] Xu, Lisong ; Perros, Harry ; Rouskas, George: Techniques for Optical Packet Switching and Optical Burst Switching. In: *IEEE Communications Magazine* 39 (2001), Januar, Nr. 1. – LDB [33225]

[19] Yao, Shun ; Mukherjee, Biswanath ; Dixit, Sudhir: Advances in Photonic Packet Switching: An Overview. In: *IEEE Communications Magazine* 38 (2000), Februar, Nr. 2. – LDB [31901]

[20] Yoo, Myungsik ; Qiao, Chunming ; Dixit, Sudhir: QoS Performance of Optical Burst Switching in IP-Over-WDM- Networks. In: *IEEE Journal on Selected Areas in Communications* 18 (2000), Oktober, Nr. 10. – LDB [32723]