System Identification of Computer Networks with Random Service

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte

Dissertation

von

Ralf Lübben, M. Sc.
geboren am 09. Januar 1981 in Emden

2013

Referent : Prof. Dr.-Ing. Markus Fidler
Korreferent : Prof. Dr.-Ing. James Gross
Tag der Promotion : 19. Dezember 2013

ABSTRACT

System theory deals with constructing models of systems to establish a relation between the inputs and the outputs of a system. System identification, as a sub area of system theory, makes this construction of the model possible by the observation of inputs and outputs. Such a construction is beneficial if a derivation of the model by first order principles, as e.g., physical laws, is difficult due to a complex or an unknown system structure.

From the perception of system theory, computer networks are systems in which the data arrivals represent the inputs and the data departures the outputs of a network path. In this thesis, we establish an analytical system identification methodology for computer networks with random service by the measurement of data arrivals and data departures. To apply the developed methodology to computer networks, we implement it into practical procedures.

We establish our system identification methodology in the framework of the stochastic network calculus, a system theory for computer networks. Thereby, the identification applies to linear systems that feature properties, which are characteristic for computer networks, such as multi-hop paths, various scheduling disciplines, and random service due to cross traffic, channel characteristics, or protocol behavior. This universality for linear systems is achieved by the use of a black-box model, where the system model is determined by measurements and no specific assumptions have to be made beforehand on the internal system structure. In the framework of the network calculus, the so-called service curve, which gives a complete description of a linear system, represents the system model. It specifies the coherence between time and data, thereby it provides a description of the system on arbitrary time scales.

We illustrate that the area of available bandwidth estimation also belongs to system identification. On the contrary to our system identification methodology, available bandwidth estimation typically uses a gray-box model of the system, i.e., a concrete system model is assumed in advance that is parameterized by measurements. Available bandwidth estimation often disregards the variability of the service by the assumption of a deterministic system. Effects that occur due to the discrepancy of this assumption and the characteristics of networks are attenuated by post processing of the measured data e.g., by averaging of multiple measurements. These simplifications also lead to descriptions that usually account for only one time scale. Furthermore, such tools often make assumptions that apply only to networks with specific properties e.g., first-in first-out (FIFO) packet scheduling, single-hop topologies, or constant rate channels, by the use of a gray-box model.

Our description of the systems in the network calculus also provides explanations for a number of fallacies observed by the application of available bandwidth estimation tools. Such explanations are possible due to our description of networks by linear time-variant systems. The description accounts for randomness, which is often disregarded.

We transfer the analytical system identification methodology into practical probing procedures. We design therefore one procedure for networks, in which properties can be assumed to be stationary for short time periods, as paths in production networks, and a second procedure for networks, in which properties are stable for infinite scales, such as dedicated testbeds.

By applying these procedures to networks with the earlier mentioned properties (multi-hop paths, various scheduling disciplines, and random service due to cross traffic, channel characteristics, or protocol behavior), we validate the methodology by comparison to results from well-known available bandwidth estimation tools and to analytical results. In doing so, we also provide system models for computer networks and protocols deployed therein for which only asymptotic results were known before.

Keywords: system identification, performance evaluation, network calculus, TCP, WLAN, random systems, available bandwidth estimation, computer networks, effective service curve

ZUSAMMENFASSUNG

Die Systemtheorie beschäftigt sich mit der Erstellung von Modellen zur Beschreibung der Beziehung zwischen Eingangs- und Ausgangssignalen von Systemen. Die Systemidentifikation, als Teilgebiet der Systemtheorie, ermöglicht diese Erstellung anhand von Beobachtungen der Eingangs- und Ausgangssignale. Dies ist von Vorteil, wenn eine Herleitung des Modells auf Grundlagen, wie z.B. physikalischen Gesetzen, aufgrund von einer komplexen oder unbekannten Struktur des Systems schwierig ist. Auch Rechnernetze sind im Rahmen der Systemtheorie Systeme, in denen die Eingangssignale Datenankünfte und die Ausgangssignale Datenabgänge sind. In dieser Arbeit wird eine analytische Methodik zur Systemidentifikation für Rechnernetze mit zufälligem Dienstangebot hergeleitet, wobei das Systemmodell durch Messung der Datenankünfte und -abgänge eines Netzwerkpfades erstellt wird. Zur Anwendung in Rechnernetzen wird diese Methodik in praktische Verfahren umgesetzt.

Die analytische Methodik zur Systemidentifikation wird im Rahmenwerk des stochastischen Netzwerkkalküls etabliert, welches eine Systemtheorie für Rechnernetze ist. Dies ermöglicht die Anwendung auf lineare Systeme, welche für Rechnernetze charakteristische Eigenschaften besitzen, wie Pfade mit mehreren Knoten, verschiedene Abarbeitungsverfahren von Warteschlangen und zufälligem Dienstangebot durch existierenden Datenverkehr, Kanaleigenschaften oder Protokollverhalten. Diese Allgemeingültigkeit für lineare Systeme wird durch Verwendung eines Black-Box-Modells erreicht, wobei das gesamte Systemmodell durch Messungen bestimmt wird und keine spezifischen Annahmen im Voraus über die interne Struktur des Systems getroffen werden. Die sogenannte Dienstkurve, welche eine vollständige Spezifikation eines linearen Systems darstellt, repräsentiert das Systemmodell. Sie gibt den Zusammenhang zwischen Zeit und Datenmenge an, wodurch sie Systeme auf beliebigen Zeitskalen beschreibt.

Wie in dieser Arbeit gezeigt wird, ist die Schätzung der verfügbaren Bandbreite ebenfalls im Bereich der Systemidentifikation anzuordnen. Im Gegensatz zu der in dieser Arbeit entwickelten Methodik, verwenden Programme zur Schätzung

der verfügbaren Bandbreite üblicherweise ein Gray-Box-Modell, d.h. es wird bereits ein konkretes Systemmodell angenommen, welches noch durch Messungen parametrisiert wird.

Die bisherigen Programme vernachlässigen allerdings meist die Variabilität des Dienstangebots, dadurch dass ein deterministisches System angenommen wird. Auftretende Effekte, die durch die Diskrepanz zwischen diesen Annahmen und Eigenschaften von Netzwerken entstehen, werden durch Nachbearbeitung der gemessen Daten abgeschwächt, z.B. durch Mittelung über mehrere Messungen. Durch diese Vereinfachungen sind oft nur Beschreibungen mit Betrachtungen einer Zeitskala möglich. Zudem gelten Programme, die Gray-Box-Modelle verwenden, nur für Netzwerke mit den jeweils angenommen Eigenschaften, wie z.B. das FIFO Abarbeitungsverfahren von Warteschlangen, Rechnernetze, die nur aus einem Knoten bestehen, oder Übertragungskanäle mit konstanter Rate.

Zur Anwendung der in dieser Arbeit erstellten Methodik wird diese in praktische Verfahren umgesetzt. Hierbei wird ein Verfahren für Netzwerke entwickelt, in denen die Eigenschaften für kurze Zeiträume als stationär angenommen werden können, wie z.B. Netzwerkpfade in produktiven Netzen. Des Weiteren wird ein zweites Verfahren für Netze entwickelt, in denen Eigenschaften für eine unbegrenzte Zeit stabil sind, was z.B. in dedizierten Testumgebungen der Fall ist.

Die Beschreibung der Systeme im Netzwerkkalkül erklärt einige bekannte Irrtümer, die bei der Anwendung von Programmen zur Messung der verfügbaren Bandbreite entstehen. Dies wird dadurch ermöglicht, dass in der erstellten Methodik die bisher oft vernachlässigte Variabilität durch die Beschreibung von Netzwerken als lineare zeit-variante Systeme berücksichtigt wird.

Zur Validierung werden die praktischen Verfahren in Netzwerken mit den bereits genannten Eigenschaften (Pfade mit mehreren Knoten, verschiedenen Abarbeitungsverfahren von Warteschlangen und zufälligem Dienstangebot durch existierenden Datenquerverkehr, Kanaleigenschaften oder Protokollverhalten) eingesetzt und die Ergebnisse mit denen von bekannten Programmen zur Messung der verfügbaren Bandbreite sowie mit analytischen Ergebnissen verglichen. Dabei werden zusätzlich Systemmodelle für Netzwerke und für darin verwendete Protokolle erstellt für die bisher nur asymptotische Ergebnisse bekannt waren.

Stichwörter: Systemidentifikation, Leistungsbewertung, Netzwerkkalkül, TCP, WLAN, zufällige Systeme, Schätzung der verfügbaren Bandbreite, Rechnernetzwerke, effektive Dienstkurve

CONTENTS

Symbols xiii
Acronyms xvi

I DISSERTATION 1

1 INTRODUCTION 2
  1.1 System Identification of Computer Networks 5
  1.2 Contribution 8
  1.3 Thesis Structure 11
2 SYSTEM MODELS FOR COMPUTER NETWORKS 13
  2.1 Conventions for Data Flows 16
  2.2 Deterministic Invariant Systems 17
  2.3 Random Systems 25
3 STATE OF THE ART IN SYSTEM IDENTIFICATION OF COMPUTER NETWORKS 31
  3.1 Available Bandwidth Estimation 32
  3.2 System Identification in the Network Calculus 39
4 PROBLEM STATEMENT 42
5 A SYSTEM IDENTIFICATION METHODOLOGY FOR COMPUTER NETWORKS WITH RANDOM SERVICE 45
  5.1 Connection between the Service Curves for Random Systems 46
  5.2 Estimation of ε-Effective Max-plus Service Curves 49
  5.3 System Identification in the Min-Plus Algebra 56
6 CONNECTION BETWEEN SYSTEM MODELS IN THE NETWORK CALCULUS AND THE AVAILABLE BANDWIDTH 66
  6.1 Single-hop Networks 67
  6.2 Multi-hop Networks 71
7 PROBING PROCEDURES FOR SYSTEM IDENTIFICATION 75
  7.1 Procedure for Networks with Change-free Regions 76
  7.2 Procedure for Networks with Stationary Service 88
8 APPLICATION OF SYSTEM IDENTIFICATION 93
  8.1 Comparison to Available Bandwidth Estimation 96
  8.2 Comparison of Service Curve Estimation Procedures 99
  8.3 Wired Networks with Random Cross Traffic 101
  8.4 IEEE 802.11a Wifi Networks 110
  8.5 Window Flow Control Protocols 113
  8.6 Transmission Control Protocol 118
9 CONCLUSION AND FUTURE WORK 128

II APPENDIX 131

A AUXILIARY MATERIAL AND PROOFS 132
  A.1 System Properties in the Min-plus and Max-plus Algebra 132
  A.2 Proofs 133
  A.3 Correlation of the TCP Congestion Window 137

BIBLIOGRAPHY 140

PUBLICATIONS 152

SCIENTIFIC CAREER 154

LIST OF FIGURES

Figure 1.1 Interpretation of a computer network as a system. 3
Figure 2.1 Comparison of constant rate arrivals in the min-plus algebra (fluid and packetized) and the max-plus algebra. 15
Figure 5.1 Exemplary sample paths and service curve with point wise and sample path bound. 48
Figure 5.2 ε-effective service curve estimate of an On-Off server in the max-plus algebra. 55
Figure 5.3 ε-effective service curve estimate of an On-Off server in the min-plus algebra. 61
Figure 6.1 Impact of timings between service and arrivals. 69
Figure 6.2 Impact of averaging. 71
Figure 6.3 Impact of the timing on the service in multi-hop networks. . 73
Figure 7.1 Flow chart of the probing procedure for networks with change-free regions. 78
Figure 7.2 Multi-hop network with multiple 100 Mbps bottleneck links each with a delay of 10 ms. 79
Figure 7.3 Used train lengths and observed steady state delay quantiles. 83
Figure 7.4 Relative frequency to pass the trend test. 84
Figure 7.5 Evaluation of the robustness of the majority decision using the trend test. 85
Figure 7.6 Comparison of estimated and predicted delay quantiles using the POT method. 88
Figure 7.7 Flow chart of the probing procedure for networks with stationary service. 90
Figure 7.8 Comparison of delay quantiles using iterative and continuous sampling. 92
Figure 8.1 Representation of service curves in the max-plus algebra, in the min-plus algebra, and as available bandwidth. 95
Figure 8.2 Comparison of available bandwidth estimates. 97
Figure 8.3 Comparison of system identification procedures. 101
Figure 8.4 Comparison of service curve estimates for unlimited and limited scales. 103
Figure 8.5 Comparison of service curve estimates for various scheduling disciplines. 105
Figure 8.6 Representation of the ε-effective service curve as ε-effective available bandwidth. 107
Figure 8.7 Service curve estimates for network topologies for one to five bottleneck links. 108
Figure 8.8 Service curve estimates for TCP and UDP cross traffic. 109
Figure 8.9 IEEE 802.11a network topology. 111
Figure 8.10 Available bandwidth in an IEEE 802.11a wireless network. . . 112
Figure 8.11 System model for window flow control. 114
Figure 8.12 Service curves for window flow control. 115
Figure 8.13 Service curve estimate for congestion control. 117

Figure 8.14 Dumbbell network topology for system identification of TCP. 119
Figure 8.15 Delay quantiles of TCP flows. 121
Figure 8.16 Delays series of TCP flows. 122
Figure 8.17 Available bandwidth of TCP for different queue sizes. 123
Figure 8.18 Available bandwidth $\alpha^{\varepsilon}(t)$ of TCP for the congestion control protocols Reno and Cubic. 125
Figure 8.19 Available bandwidth of TCP for active queue management. . 126
Figure A.1 Auto correlation of the CWND. 138

LIST OF TABLES

Table 8.1 Available bandwidth estimates for a multi-hop topology with multiple bottleneck links. 99
Table 8.2 Long-term available bandwidth estimates and delay quantile estimates for elastic cross traffic. 110
Table 8.3 One way delays of network paths. 119
Table A.1 95% delay quantile in milliseconds for a probing rate of 5 Mbps138

SYMBOLS

$\otimes$	convolution operator
$\lfloor x \rfloor$	rounding down
$[x]^+$	$max(0, x)$
$\Pi(\cdot)$	operator that maps inputs to outputs
$1_{x \in y}$	indicator function, which equals 1 if $x \in y$, else it equals 0
$\alpha(\tau, t)$	available bandwidth
$\widetilde{\alpha}(t)$	available bandwidth estimate
$\alpha^{\varepsilon}(t)$	ε-effective available bandwidth
$\widetilde{\alpha}^{\varepsilon}(t)$	estimate of $\alpha^{\varepsilon}(t)$
α^{∞}	long-term available bandwidth
α_h^{∞}	long-term available bandwidth of link h
α_{net}^{∞}	long-term available bandwidth of a network path
$A(t)$	cumulative arrivals up to time t
$A^{-1}(n)$	max-plus pseudo inverse of $A(t)$
$B(t)$	backlog at time t
$B(n)$	backlog of packet n
$B(r, t)$	backlog at time t obtained by probing rate r
$B^{\xi}(r)$	$(1 - \xi)$-backlog quantile
$\widetilde{B}^{\xi}(r)$	$(1 - \xi)$-backlog quantile estimate
B_{max}	maximal backlog bound
C	link capacity
C_h	capacity of link h
δ	time interval
$D(t)$	cumulative departures up to time t
ε	violation probability with respect to a service curve
$\mathsf{E}[x]$	expected value of x
$E(t)$	envelope function for $A(t)$
I	iterations per probing rate
$\underline{\mathfrak{F}}$	concave Fenchel conjugate
$\overline{\mathfrak{F}}$	convex Fenchel conjugate
$\mathcal{F}$	set of univariate non-negative continuous or right-continuous wide-sense increasing functions

F	set of bivariate non-negative continuous or right-continuous wide-sense increasing functions
$\mathcal{G}$	set of univariate non-negative discrete wide-sense increasing functions
$\mathcal{G}^*$	set of univariate positive discrete wide-sense increasing functions
G	set of bivariate non-negative discrete wide-sense increasing functions
g_I	gap between packets at the ingress
g_O	gap between packets at the egress
H	number of hops of a network path
λ	long-term cross traffic rate
λ_h	long-term cross traffic rate on link h
m	interval of packets
n, ν	packet indexes
$\mathbb{N}_0$	natural numbers, including 0
N	train lengths
N_1	initial train length
N_i	train length of the ith run
N_{max}	maximal train length
N_{off}	train length offset, where no sampling is performed
N_i^S	mean train length, where sampling is performed
N_1^S	initial train length for N_i^S
$P^L(x)$	packetizer with length function $L(n)$
$\mathbb{R}^+$	positive real numbers, excluding 0
$\mathbb{R}_0^+$	positive real numbers, including 0
r	data rate
r_1	initial probing rate
r_{acc}	estimation accuracy with respect to the rate
r_I	arrival rate
r_O	departure rate
R	set of probing rates
$\mathcal{S}(t)$	deterministic min-plus service curve
$\mathcal{S}_{net}(t)$	deterministic min-plus service curve of a multi-hop network path
$\mathcal{S}^\varepsilon(t)$	ε-effective min-plus service curve
$\widetilde{\mathcal{S}}(t)$	estimate of $\mathcal{S}(t)$
$\widetilde{\mathcal{S}}^\varepsilon(t)$	estimate of $\mathcal{S}^\varepsilon(t)$

$\mathcal{S}_h(t)$	$\mathcal{S}(t)$ of hop h, with h=$\{1,2,3,...,H\}$
$S(\tau,t)$	time-varying min-plus service curve
$S_h(\tau,t)$	$S(\tau,t)$ of hop h, with h=$\{1,2,3,...,H\}$
$S_l(\tau,t)$	time-varying leftover service curve
$S_{net}(\tau,t)$	time-varying min-plus service curve of a multi-hop network path
t, τ	time indexes
$T_A(n)$	arrival time of packet with index n
$T_A^{-1}(t)$	min-plus pseudo inverse of $T_A(n)$
$T_D(n)$	departure time of packet with index n
$T_E(n)$	envelope function for $T_A(n)$
$T_{\mathcal{S}}(n)$	deterministic max-plus service curve
$T_{\mathcal{S}}^{\varepsilon}(n)$	ε-effective max-plus service curve
$\widetilde{T}_{\mathcal{S}}(n)$	estimate of $T_{\mathcal{S}}(n)$
$\widetilde{T}_{\mathcal{S}}^{\varepsilon}(n)$	estimate of $T_{\mathcal{S}}^{\varepsilon}(n)$
$T_{\mathcal{S}_h}(n)$	$T_{\mathcal{S}}(n)$ of hop h, with h=$\{1,2,3,...,H\}$
$T_{\mathcal{S}_{net}}(t)$	deterministic max-plus service curve of a multi-hop network path
$T_S^{-1}(t)$	min-plus pseudo inverse of $T_{\mathcal{S}}(n)$
$(T_{\mathcal{S}}^{\varepsilon})^{-1}(t)$	min-plus pseudo inverse of $T_{\mathcal{S}}^{\varepsilon}(n)$
$T_S(\nu,n)$	shift-varying max-plus service curve
$T_{S_h}(\tau,t)$	$T_S(\tau,t)$ of hop h, with h=$\{1,2,3,...,H\}$
$T_{S_{net}}(\tau,t)$	shift-varying max-plus service curve of a multi-hop network path
$W(n)$	delay of packet n
$W(t)$	delay at time t
$W(r,n)$	delay of packet n obtained by probing rate r
$W^{\xi}(r)$	$(1-\xi)$-delay quantile
$\widetilde{W}^{\xi}(r)$	$(1-\xi)$-delay quantile estimate
W_{max}	maximal delay bound
ξ	violation probability with respect to quantiles

ACRONYMS

AQM	Active Queue Management
ARQ	Automatic Repeat Request
BDP	Bandwidth Delay Product
CSMA/CA	Carrier Sense Multiple Access and Collision Avoidance
CWND	Congestion Window
DCF	Distributed Coordination Function
ECN	Explicit Congestion Notification
ERS	a unit root test from Elliot, Rothenberg and Stock
FIFO	First-in First-out packet scheduling
MAC	Medium Access Control
Mbps	Megabits per second
POT	Peaks over Threshold
pps	Packets per Second
QoS	Quality of Service
RED	Random Early Detection
RTT	Round Trip Time
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Part I

DISSERTATION

1

INTRODUCTION

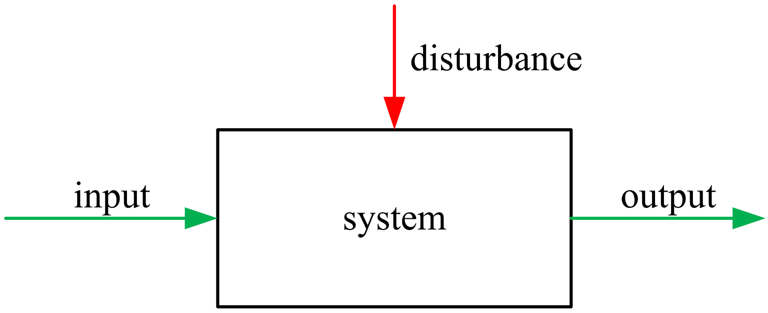
System identification deals with building models of systems by observing signals at the ingress and the egress of the systems. In system theory, a system is defined as [69]:

> “... an object in which different variables interact at all kinds of time and space scales and that produces observable signals“.

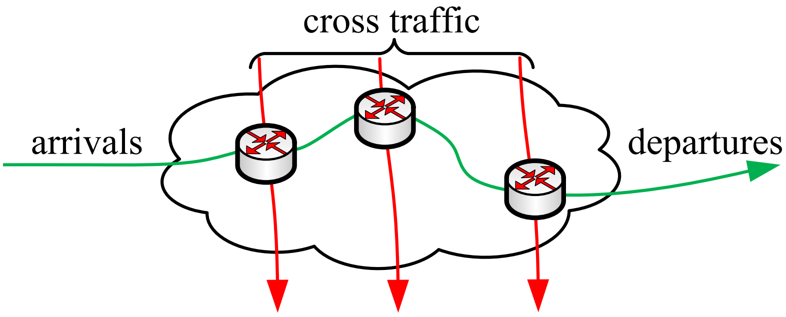
This definition applies for many subject areas as biology, chemistry, economics, engineering, or physics just to name a few [127]. Constructing models of systems from the knowledge of system internals or from measurements supports the comprehension and analysis of these systems. For example, in engineering sciences the stability of systems, which prevails if a system reacts on bounded inputs only with bounded outputs, is often analyzed. We denote the system theory from engineering, such as electrical engineering or mechanical engineering, as the classical system theory in the following.

According to [85], the term system identification was established in [134] in 1956 and deals with the identification of dynamic system in the area of control theory. In the subsequent years, system identification attracted great attention for automatic control [12]. The conjunction of automatic control and system identification led to adaptive control, whereat the controller is adjusted to the system by identification of it. This enables the adjustment of the controller for unknown systems or the parametrization of an assumed system model, e.g., if the system behavior changes slowly over time [13].

Although the term system identification was phrased in the area of control theory, the idea of determining system models from observations exists in many other areas as statistics, time series analysis, machine learning, neural networks, etc., as elaborated in [85]. For further details on system identification in the classical system



(a) general system



(b) computer network as a system

Figure 1.1: Interpretation of a computer network as a system. In computer networks the inputs become the arrivals and the outputs the departures.

theory and on the classical system theory itself, we refer to [84]. In this thesis, we extend system identification to computer networks.

Fig. 1.1a presents a general system with inputs, outputs, and disturbances. In a mathematical notion, the system is an operator that maps inputs to outputs [74]. This mapping is specified by a mathematical operation and a system model. We also refer to it as the system description in a non-mathematical notion. For example in the classical system theory, the mathematical operation is the convolution and the impulse response is the system model. The disturbances further affect the system and prevent an exact relation between the inputs and the outputs.

Fig. 1.1b shows a computer network with data arrivals and departures. The data traverse a network path from the ingress to the egress of that path; we refer to this data flow as the through flow. This through flow interacts with existing traffic on its path, the so-called cross traffic. If the computer network is interpreted as a system, the inputs to the system are the arrivals of the through flow and the outputs are the departures of the flow. The system behavior is defined e.g., by the cross traffic, the employed protocols and scheduling disciplines in the network, and the characteristics of the transport medium as wired or wireless links. For example, in real networks the cross traffic is usually random. Depending on the model class of a system the randomness of the cross traffic can be viewed as disturbances or as a characteristic inherent in the system. Assuming a deterministic system, the randomness of cross traffic disturbs the exact relation between the inputs and outputs, but assuming a system with random service, the randomness codetermines the model. In computer networks, the analysis is often with respect to quality of service (QoS) parameters as e.g., throughput, which is represented as the available bandwidth in this work, and delay. Such parameters are of interest for real-time applications that have specific requirements on these parameters.

A system theory for computer networks is the network calculus [15, 29, 61, 74], which provides a framework for deterministic systems as well as systems that feature randomness. It makes use of the min-plus algebra or the max-plus algebra. Compared to the classical algebra, the plus operator is replaced by the minimum operator or by the maximum operator with respect to the algebra, and the multiplication operator becomes the plus operator. Although the classical system theory and the network calculus deviate in the algebra, many analogies exist, e.g., the convolution operations exists in the min-plus as well as the max-plus algebra and fundamental classes to categorize systems in the classical system theory can also be applied to computer networks e.g., linearity versus non-linearity and time-variance versus time-invariance[1].

The system model in the network calculus is expressed by a service curve. Phrased simply, a service curve in the min-plus algebra specifies the amount of data that a system is able to forward between two instances of time. In the max-plus algebra, a service curve states the time required to forward a specific amount of data.

The foundation of the network calculus was laid in 1991 in [31, 32], in which traffic is described by constraints on its burstiness. This description allows the derivation of performance bounds such as delay and backlog bounds for systems. More precisely, the work is the foundation of the deterministic network calculus since worst-case bounds are used for the description of the data traffic, the systems, and the performance bounds. To this day, descriptions for systems as constant rate links, various scheduling disciplines, delay elements, traffic shapers, and window

---

[1] For details on theses model classes, see Chap. 2 and Appendix A.1

flow control protocols exist just to name a few. Even more, the network calculus offers the analysis of combinations of systems as connections in parallel or series. Comprehensive summaries exist in [29, 43, 74].

For the description of networks with random service, the network calculus provides two different approaches. On the one hand, an approach for linear and non-linear time-varying systems is established in [29, 68, 74], where the system model is represented by a random process. On the other hand, there is the stochastic extension, namely the stochastic network calculus, that accounts for randomness by using probabilistic bounds on the system model. The basis of the stochastic network calculus was laid by the extension of the seminal work [31, 32] in [25, 27, 71]. Comprehensive summaries are given in the monographs [29, 61] and the survey [43]. To this day, stochastic network calculus has allowed the analysis of state of the art Internet traffic models e.g, in [81, 96, 113].

While network calculus is a system theory for computer networks, it also finds its way to other fields such as real time systems [123], information theory [86, 87], sensor networks [118], smart grids [128], and battery lifetime [75, 131].

In the framework of the network calculus, many works engage in finding the system model with the full knowledge of the internals of the systems. Such models are known in the notion of system identification as white-box models [69]. However, with incomplete knowledge or when complexity prevents a direct derivation of a white-box model, system identification leads to models built by measurements. In the classical system theory, many works exist in the field of system identification, see e.g., [69, 84]. Contrary, in the framework of the network calculus only few works are available for system identification.

1.1 SYSTEM IDENTIFICATION OF COMPUTER NETWORKS

In system identification, system models are built from measurements of the inputs and the outputs of systems, in which the internal system structure is only partially available or completely unknown. If the internal structure is partially known and only the parameters of an assumed model are determined by measurements, these models are called gray-box models. Whereas, if no internals are known for a system, we refer to black-box models, at which the system model is completely determined by measurements [69].

To this day, there are only few approaches for system identification in the framework of the network calculus. Gray-box models are used for building deterministic system models of routers in [22, 125] based on measurements. In [80], it is shown that system identification on the basis of black-box models is also feasible in the framework of the deterministic network calculus. This establishes system identification in the network calculus that applies to a broad range of min-plus linear systems since the identification procedure does not rely on any specific system structure. The latter work has its origin in the framework of the network calculus, but it is also related to available bandwidth estimation since the probing approaches originate from the available bandwidth estimation tools presented in [58, 111].

Besides the approaches named above, which are developed in the framework of the network calculus, available bandwidth estimation also belongs to system identification. The available bandwidth specifies the unused capacity of a network path, which is of interest e.g., for applications to test whether required sending rates are achievable or to adapt to the available bandwidth, for congestion control not to overload a path, or for testing the quality of service (QoS) of a path. Many tools developed for available bandwidth estimation assume a deterministic single-hop network with fluid traffic and a first-in first-out (FIFO) scheduling discipline e.g., [54, 58, 59, 94, 111, 122]. The resulting system model, the tools rely on, is a gray-box model. It leads to the description of the available bandwidth by usually a single value and assumes it to be constant over time, notwithstanding in practical networks it varies over time. The available bandwidth is derived from information that is imprinted on packets when they traverse the network path. This information is extracted e.g., by measurements of the end-to-end delay from timestamped probe packets or by an evaluation of the dispersion of packets. Available bandwidth estimation tools that rely on delay measurements are presented in [58, 59], and tools that measure the packet dispersion are described in [54, 94, 122].

Obviously, many assumptions made for the construction of the system model, which is used by these tools, are not met in practical computer networks. The available bandwidth is not deterministic since traffic in the network is random and typically a network path consists of multiple hops. The assumptions of constant rate links and FIFO scheduling are appropriate for wired networks but must be adapted to wireless networks. For example, the medium access procedure in wireless networks that are based on the IEEE 802.11 standard results in a non-FIFO scheduling behavior, see [21, 108]. Additionally, the medium access procedure in these networks leads to non-work-conserving systems since before data is transmitted via the channel, a random latency occurs even if the channel is idle [67].

Existing available bandwidth estimation tools produce acceptable results if the assumptions are mildly violated, e.g., if the assumption of determinism holds for time scales observed by a bandwidth estimation tool. In [20, 60], the impact of the simplified model used by many bandwidth estimation tools is reported. The applicability of the available bandwidth estimation tools [54, 58, 59, 78, 94, 111, 122] to IEEE 802.11 networks is limited. The tools show partially a strong deviation from the expected available bandwidth as demonstrated by measurements [20]. In addition, underestimation of the available bandwidth due to randomness of the cross traffic and multi-hop networks is reported in [60]. On the one hand, this led to available bandwidth estimation tools adapted to IEEE 802.11 networks e.g., [62, 64, 78], on the other hand, the impact of the assumption of a deterministic system with fixed-capacity links and FIFO scheduling is analyzed and relaxed with respect to randomness in [35, 52, 82, 83, 104, 105].

To summarize, few system identification methodologies exist in the network calculus. These methodologies show the applicability of gray-box and black-box models but neglect the randomness of systems. Also, available bandwidth estimation belongs to system identification, in which the system is often described by a single value. The usual system model utilized for available bandwidth estimation ignores the randomness inherent in real computer networks and thereby neglects that the available bandwidth varies on different time scales. Few approaches include randomness in their system description, but these are typically customized to networks for which specific assumptions apply.

1.2 CONTRIBUTION

We contribute an analytical methodology in the framework of the stochastic network calculus to the system identification of computer networks that belong to the class of linear systems. Thereby, we enable the determination of system models for networks with random service without the knowledge of the internal system structure by relying on a black-box model. The implementation of this methodology into practical procedures allows us to validate the procedure and also to determine the system model of various networks by measurements. Utilizing the network calculus for the modeling of systems, brings also new insights into available bandwidth estimation. Below, we state the contributions in detail.

The first main contribution is the derivation of the system identification methodology in the min-plus and the max-plus algebra. For this derivation, we prove the relation between two descriptions for networks with random service in the framework of the network calculus: a description by random processes, which characterizes the randomness by a linear time-variant system operator, and a probabilistic description, which uses a probabilistic bound to account for the randomness. Using the relation between these two descriptions, we solve the inversion problem that is the derivation of the system model from delay or backlog measurements. This leads to our system identification methodology that characterizes the system model by a service curve. We give evidence that probing the network path with constant rate packet trains, which consist of multiple successive equally spaced packets, and that extracting quantiles from the delay distributions or backlog distributions for various probing rates, leads to service curves that conform to the well-known definition of the ε-effective service curve, which is introduced in [25]. Furthermore, we prove conditions when steady state delay and backlog distributions exist, which are required for our system identification methodology. We derive our methodology in the two algebras that are used in the network calculus. The max-plus algebra is advantageous in practical applications for the derivation of the service curve from measurements. The min-plus algebra is widely used in the network calculus, because it simplifies the calculation for multiplexing of traffic, and it provides a more intuitive description. We prove the relation between our methodologies in both algebras by the definition of a pseudo-inverse, which only results in a difference of at most one packet. Using this framework of the network calculus, the system identification methodology holds for a broad range of systems, ranging from simple constant rate links up to complex systems of multi-hop networks with random cross traffic and non-work-conserving systems as IEEE 802.11 networks and also complex protocols.

Second, we contribute new insights to available bandwidth estimation. Since our system description belongs to the class of time-varying systems and it thereby applies for networks with random service, it explains effects where available bandwidth estimation deviates from real-world results. These effects are ascribed to the choice of the system model often used for available bandwidth estimation, which neglects properties inherent in real computer networks. First, we prove that the leftover service curve [29, 42] resembles the definition of the available bandwidth for single-hop networks. Furthermore, we show that the expected value of the available bandwidth leads to an overestimation of the departures. For multi-hop networks, we prove that the definition of the available bandwidth [82, 83] is recovered by a service curve description only in the limit. Additionally, our methodology accounts for the service availability on arbitrary scales and not only in the limit, which explains the underestimation reported for available bandwidth estimation tools in multi-hop networks [60].

The specification of the probing procedures is the third main contribution. The system identification methodology leaves space for the parameter selection for the probing procedure, which is based on constant rate packet trains. The set of probing rates, the number of iterations, and the train length have to be specified for a practical system identification procedure. In theory, the number of iterations and the train length should be infinite. However, in practice we show that these parameters can be reduced to finite values by means of statistical methods, including the specification of the reliability by confidence intervals. For the identification, we require that the probing rates lead to steady state delay or backlog distributions. We follow two different design goals for the probing procedure. The first procedure targets on systems, where stationary path characteristics can only be assumed in the short-term, e.g., as it applies to Internet paths for several hours [135]. For such systems, providing fast results is preferred. Therefore, we introduce a heuristic to decrease the measurement duration by reducing the required train length for the system identification. Furthermore, the rate selection achieves a fast convergence to the maximum rate that observes steady state backlog or delay distributions. This procedure can also be pruned to the task of estimating the available bandwidth as a single value. The second procedure targets on systems, where the assumption of stationarity applies also in the long-term and long measurement durations are acceptable as e.g., in dedicated testbeds arranged for testing network topologies and protocols. Moreover, a fine granular rate selection is used to extract more details from the system.

The fourth main contribution is the experimental validation and evaluation of our system identification procedures in controlled testbed environments [1, 130] and by simulation. We compare our system identification procedures to well-known available bandwidth estimation tools and system identification methodologies from the deterministic network calculus. We show results for various kinds of networks and protocols as single- and multi-hop networks with random traffic, wireless networks with non-work-conserving behavior and protocols such as window flow control, congestion control, and the transmission control protocol (TCP), which is the prevalent transport protocol in the Internet. We validate our estimated system models by comparing it to known analytical results for various networks. The experiments also confirm the findings for multi-hop networks, where service availability only recovers the available bandwidth in the limit. Beyond that, we deliver service curves for networks, where no analytical results exist so far or only fragmentary findings are available e.g., as asymptotic results. These results include service curves for wireless IEEE 802.11 networks and also for the TCP protocol. The service curve estimates include results on the short-term and the long-term behavior and thereby generate a comprehensive analysis of the systems compared to existing asymptotic results.

Especially, the application to TCP and wireless networks shows that our system identification methodology has a broad range of application, which goes beyond the field of available bandwidth estimation approaches. Furthermore, TCP is a challenging protocol since analytical results only exist for simplifying assumptions [24, 93, 100], although it is the standard transport protocol in the Internet [92]. Recently, the delay introduced by TCP is brought into focus in [24, 37, 48] since it is utilized by real-time applications [92] and large delays have a negative impact on the quality of user experience [37]. Many modifications of TCP are justified by simulations and experiments. Examples are active queue management (AQM) approaches as random early detection (RED) [46] and implementations in the TCP stack as e.g., various congestion control protocols as TCP Cubic [51]. Our system identification methodology offers a systematic evaluation of these features and estimates the service of the entire end-to-end path from sender to receiver and not only parts of it. Moreover, the estimates comprise the behavior of the protocol for arbitrary input rates. Thereby, the estimates provide models that are applicable to application traffic, which goes beyond simple bulk transfer. Such models are required for today's application as depicted in [115].

To sum up, our system identification procedure is applicable to linear systems with random service, which is inherent in many computer networks. By choosing an appropriate model class for computer networks, we can overcome weaknesses of deterministic models used for available bandwidth estimation or system identification, so far. The service curve as a system model gives a coherent description of delay and rate. Also, by using the framework of the network calculus, we bring new insights into the field of available bandwidth estimation by explaining effects identified before as fallacies in this area.

1.3 THESIS STRUCTURE

The rest of the thesis is structured as follows. In Chap. 2, we introduce the basics of system models known in the network calculus. We include system models for the classes of deterministic systems and systems with random service in the min-plus as well as in the max-plus algebra and show how performance bounds such as delay and backlog are computed in the network calculus.

We discuss the state of the art of system identification approaches in Chap. 3. This includes system identification approaches derived in the framework of the network calculus and also approaches from available bandwidth estimation, which we present from a system theoretic perspective.

In Chap. 4, we summarize the aspects of the problem, which are dealt with in this thesis.

In the subsequent Chap. 5, we derive our system identification methodology. We interrelate network calculus approaches for systems with random service introduced in Chap. 2 and present our system identification methodology in the max-plus and the min-plus algebra. This chapter also includes a first intuitive example for system identification of an On-Off server.

In Chap. 6, we establish the connection from the network calculus to common definitions of the available bandwidth. We also present analytical results for effects that arise if the definition, which is commonly used in available bandwidth estimation, is applied to random networks.

Chap. 7 provides practical guidelines on the selection of the probing parameters as the probing rates, the train lengths, and the number of iterations. We therefore derive probing procedures for two different design goals. On the one hand, for a fast estimation procedure, preferable in productive networks, and on the other hand, an estimation procedure that gives more detailed estimates by using more probing rates and longer packet trains. This procedure is usually acceptable in dedicated networks for identification purposes.

Chap. 8 presents service curve estimates for various protocols and networks obtained from simulations and experiments. These estimates are used for validation where analytical results exist; if no results are available so far, these estimates contribute to new system models.

At last, Chap. 9 concludes the thesis and gives a perspective of the future work.

2

SYSTEM MODELS FOR COMPUTER NETWORKS

This chapter introduces system models that exist in the framework of the network calculus for the description of deterministic networks and networks with random service. A system[2] is described in the network calculus by the convolution operation and a service curve. This description specifies the mapping from inputs to outputs with the service curve as the system model. It can be defined either in the min-plus or in the max-plus algebra. In the classical system theory, the description is analog to the convolution and the impulse response of a linear time-invariant system. For linear systems, the service curve and the impulse response are complete descriptions of the respective systems.

In the classical system theory, four fundamental classes arise from combinations of properties a system has as linearity versus non-linearity and time-invariance versus time-variance. If a system is time-invariant, the response to an input signal does not depend on the instant of time when the signal is applied to the ingress of the system. In the classical system theory, linearity implies that the output in response to a signal consisting of an addition of two single inputs is equal to the addition of the outputs of the individual signals and that a scaling of the input signal leads to the same scaling of the output signal. In the network calculus, these properties transfer according to the respective algebra as described below for the min-plus and the max-plus algebra[3]. Time-(in)variance applies to the min-plus algebra. Using the max-plus algebra, a system is (in)variant with respect to the amount of data to which is referred as shift-(in)variance.

Besides this classification, systems in the network calculus are often classified as deterministic systems or systems with random service, which follows from the differentiation of the deterministic network calculus and the stochastic network

---

[2] The terms computer network and system are used interchangeably in the following, at which the system is the abstraction of the network.

[3] For the definition of the properties, we refer to Appendix A.1

calculus as e.g., in [30]. We follow this classification in this chapter and present service curves for deterministic systems that are time- or shift-invariant and service curves for networks with random service that are time- or shift-variant or that use probabilistic invariant bounds on the service curve.

Many other similarities exist between the network calculus and the classical system theory [74], e.g., transforms exist that have similar features as the Laplace transform and the Fourier transform used in the classical system theory [44]. Properties of these transforms are advantageous for system identification, which we elaborate in Chap. 3 and Chap. 5.

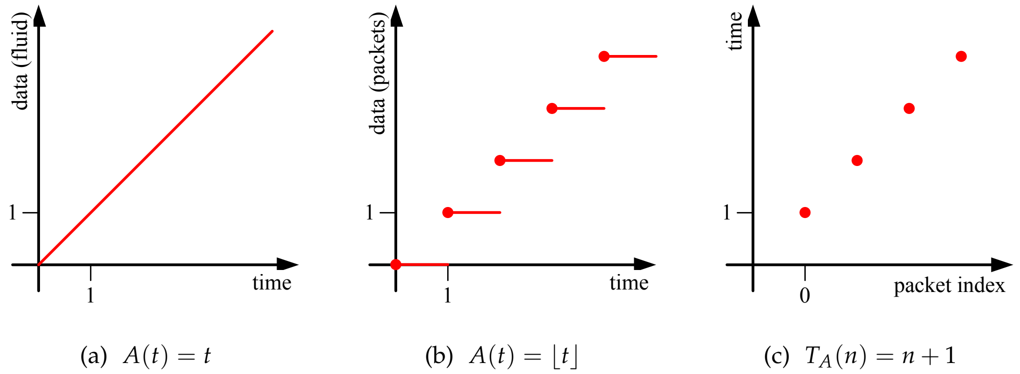
The network calculus makes use of the min-plus algebra and the max-plus algebra. In the min-plus algebra, the plus is replaced by the minimum operation and the multiplication by the plus operation. Accordingly in the max-plus algebra, the plus is exchanged by the maximum operation and the multiplication by the plus operation. Instead of the minimum and the maximum often the infimum and the supremum are used, respectively, as a generalization. This generalization includes the application to sets or functions, for which a minimum or a maximum may not exists. For details on these algebras see [15, 29, 74].

In the network calculus, the inputs to a system are denoted as the arrivals and the outputs as the departures. The outputs of a system follow by the convolution, which is introduced in the next sections, of the arrivals and the service curve of the system. In the min-plus algebra, functions describe the cumulative amount of data in the interval $(0,t]$[4] with the convention that $A(0)=D(0)=0$. We denote the arrivals in the interval by $A(t)$, the departures by $D(t)$, and the service curve by $S(t)$. The functions in the max-plus algebra are functions of data instead of time, i.e., for unit size packets this coincides with the index $n \geq 0$ of a packet. The arrivals are $T_A(n)$ and the departures $T_D(n)$, which specify the arrival time and departure time, respectively, of a packet with index n. $T_S(n)$ denotes the service curve.

We introduce three different kinds of system models: service curves for deterministic systems, service curves for systems with random service described by

---

[4] From the convention of right-continuous functions the interval $(0,t]$ follows. Right-continuous functions arise from the application of a packetizer to fluid flows [29, 74], which is used to transform fluid traffic to packetized traffic throughout this work. Often also left-continuous functions are assumed, for a discussion see [74, Sec. 2.3.2].



(a) $A(t) = t$ (b) $A(t) = \lfloor t \rfloor$ (c) $T_A(n) = n + 1$

Figure 2.1: Graphical representations of arrivals with constant rate one. Fig. (a) shows the arrivals for fluid traffic in the min-plus algebra, Fig. (b) illustrates the packetized arrivals of Fig. (a), and Fig. (c) presents the related arrivals in the max-plus algebra.

stochastic processes, and service curves for systems with random service specified by invariant functions. The invariant functions come along with a violation probability ε that specifies the probability that these invariant functions are violated. We therefore have different labels for the different kinds of service curves and in each case one for the specific algebra. For deterministic systems the service curve is denoted by $\mathcal{S}(t)$ or $T_{\mathcal{S}}(n)$, for random systems described by stochastic processes by $S(\tau, t)$ or $T_S(\nu, n)$, and for random systems specified by probabilistic bounds by $\mathcal{S}^\varepsilon(t)$ or $T^\varepsilon_{\mathcal{S}}(n)$.

We assume fluid traffic in the min-plus algebra. To address packetized traffic in the min-plus algebra, we use the concept of the packetizer, whereas in the max-plus algebra, we only consider unit size packets to reduce the notational complexity, which is sufficient throughout this work. With additional effort, it can be extended to networks with variable packet sizes, see [29, Chap. 6].

As an example for the min-plus and the max-plus algebra, Fig. 2.1 shows constant rate arrivals for a rate of one. Fig. 2.1a presents the arrivals in the min-plus algebra. Since we use fluid traffic, it follows that $A(t) = t$. Fig. 2.1b shows the packetized arrivals of $A(t) = t$ for a unit packet size that are $A(t) = \lfloor t \rfloor$ following from the concept of a packetizer as e.g., defined in [74, Sec. 1.7]. Fig. 2.1c displays the arrivals in the max-plus algebra, where the first packet has the index $n = 0$.

One benefit of the network calculus is the simple derivation of performance bounds i.e., backlog and delay bounds, if the service curve and the arrivals are known. Both algebras allow the computation of delay and backlog bounds, whereas the min-plus algebra is beneficial for the computation of the backlog and the max-plus algebra for the delay, as shown in the subsequent sections.

In the following, we present details on the service curves. We first introduce conventions for data flows assumed throughout this work in Sec. 2.1. In Sec. 2.2, we present the description for deterministic systems that build the foundation for the random systems. For the basics we refer to [29, 74], in [29] a time-discrete traffic model is used in the min-plus algebra and mostly a fluid one in [74]. For the discrete traffic model, the infimum operation can be replaced by a minimum operation [29]. In this work, we use continuously the infimum in the min-plus algebra, also if we refer to [29]. Sec. 2.3 introduces two concepts for the description of random systems. First, we describe the approach from [29, Chap. 5] that extends the univariate functions to bivariate ones, e.g., $\mathcal{S}(t)$ becomes $S(\tau, t)$, to account for variability. Second, in Sec. 2.3.2, we present the framework of the stochastic network calculus, which provides a probabilistic framework for random systems. Here, we describe primarily the approach from [25], for a broad overview on the stochastic network calculus we refer to [15, 29, 30, 43, 61].

2.1 CONVENTIONS FOR DATA FLOWS

Before we present the service curves, we introduce some general conventions valid throughout this work. We assume that data are processed in order and no reordering occurs in the network. Furthermore, we define sets for univariate and bivariate functions in the min-plus algebra and the max-plus algebra, which arise from the variety of model classes. We assume functions are continuous in the min-plus algebra, we also refer to it as fluid data flows, and discrete in the max-plus algebra based on a unit packet size. To establish a connection between the min-plus algebra and the max-plus algebra, we use the concept of a packetizer [29, 74], which results in right-continuous functions in the min-plus algebra.

For the univariate functions, we define the sets $\mathcal{F}$, $\mathcal{G}$, and $\mathcal{G}^*$. In the min-plus algebra, $\mathcal{F}$ is the set of univariate non-negative continuous or right-continuous wide-sense increasing functions, i.e., for any function $f \in \mathcal{F}$ and $\forall 0 \leq \tau \leq t$ it holds that $f(t) \geq 0, f(\tau) \leq f(t), f(0) = 0$ and $f(t), \tau, t \in \mathbb{R}_0^+ = [0, \infty)$.

In the max-plus algebra, the domain of the functions is discrete because of the assumption of unit size packets. Therefore, we specify the set of univariate non-negative discrete wide-sense increasing functions $\mathcal{G}$, i.e., for any function $g \in \mathcal{G}$ and $\forall 0 \leq \nu \leq n$ it holds that $g(\nu) \leq g(n)$, $g(n) \in \mathbb{R}_0^+$, and $n, \nu \in \mathbb{N}_0 = \{0, 1, 2, \ldots\}$.

We further define the set $\mathcal{G}^* \in \mathcal{G}$ with the restriction that $g(n) \in \mathbb{R}^+ = (0, \infty)$[5].

For the bivariate functions, we define the sets F and G. In the min-plus algebra, F is the set of bivariate non-negative continuous or right-continuous wide-sense increasing functions, i.e., for any function $f \in F$ and $\forall 0 \leq \tau \leq t$ it holds that $f(\tau, t) \geq 0, f(\tau, t) \leq f(\tau - \delta, t), f(\tau, t) \leq f(\tau, t + \delta), 0 < \delta, f(t, t) = 0$ and $f(t), \tau, t \in \mathbb{R}_0^+$.

In the max-plus algebra, G is the set of bivariate non-negative discrete wide-sense increasing functions, i.e., for any function $g \in G$ and $\forall 0 \leq \nu \leq n$ it holds that $g(\nu, n) \leq g(\nu - 1, n), g(\nu, n) \leq g(\nu, n + 1)$, $g(\nu, n) \in \mathbb{R}_0^+$, and $\nu, n \in \mathbb{N}_0$.

2.2 DETERMINISTIC INVARIANT SYSTEMS

In the following, we describe the basic principles for deterministic time-invariant and shift-invariant systems. In the min-plus algebra, these systems are described by functions of time t, such that the output $y(t) = \Pi(u(t))$ for any input signal $u(t)$ is equal to the time-shifted output $y(t - \delta) = \Pi(u(t - \delta))$ for any time-shifted input $u(t - \delta)$, where Π is the mapping function from inputs onto outputs and $\delta \geq 0$ a time-shift. In the max-plus algebra, the functions become functions of data instead of time and invariance refers to shift-invariance i.e., invariant in terms of the packet index. We refer to Appendix A.1 for details on the related system properties.

We present the service curve definitions in the min-plus algebra as well as the max-plus algebra, present the derivation of the performance bounds backlog and delay, show the connections between both algebras, and present transformations

---

[5] Note that we define the restriction $g(n) \in \mathbb{R}^+$ only to comply with the set $\mathcal{F}$ for the inversion from the max-plus algebra to the min-plus algebra defined in Sec. 2.2.3, where $f(0) = 0$

in a rate domain, which features useful properties for building service curves and deriving performance bounds.

2.2.1 *System Models in the Min-Plus Algebra*

In the min-plus algebra, the arrival function $A(t) \in \mathcal{F}$ specifies the cumulative amount of data arriving at the system in the time interval $(0, t]$ and the departure function $D(t) \in \mathcal{F}$ is the cumulative amount of data departing from the system in the time interval $(0, t]$. A deterministic time-invariant system has the service curve $\mathcal{S}(t) \in \mathcal{F}$ if

$$D(t) \geq \inf_{\tau \in [0,t]} \{A(\tau) + \mathcal{S}(t - \tau)\} =: A \otimes \mathcal{S}(t) \tag{2.1}$$

holds for all $t \geq 0$, where $\otimes$ is the convolution operator. If the system is linear, an exact service curve exists and the inequality becomes an equality

$$D(t) = \inf_{\tau \in [0,t]} \{A(\tau) + \mathcal{S}(t - \tau)\}. \tag{2.2}$$

If no service curve can be stated, which holds with equality, the maximal non-trivial service curve is typically sought in Eq. (2.1). These service curve definitions are formulated in [7, 28, 34]. The concept of the service curve was previously introduced in [33, 102, 103, 116].

A basic example of a service curve is a constant rate link with capacity C, which has the service curve $S(t) = Ct$. If the link has the delay $\mathcal{T}$, the service curve becomes $\mathcal{S}(t) = [C(t - \mathcal{T})]^+$, where $[x]^+ := \max(0, x)$.

Analog to the concatenation of systems in the classical system theory, the concatenation of single systems results in the network calculus from the convolution of the individual service curves $\mathcal{S}_h(t)$ of the systems in series. The end-to-end service curve $\mathcal{S}_{net}(t)$ of a network path with hops $h = \{1, 2, \ldots, H\}$ becomes

$$\mathcal{S}_{net}(t) = \mathcal{S}_1(t) \otimes \mathcal{S}_2(t) \otimes \cdots \mathcal{S}_H(t).$$

For further examples of systems, we refer to [29, 74].

The use of the service curve of a system enables the derivation of performance bounds i.e., backlog and delay bounds [29, 74]. The performance bounds in the min-plus algebra are as follows.

The backlog $B(t)$ in the system at time $t \geq 0$ is the difference between the cumulative arrivals and the cumulative departures

$$B(t) = A(t) - D(t). \tag{2.3}$$

If $D(t)$ is substituted by Eq. (2.1) and if $A(t)$ is bounded from above by a function $E(t) \in \mathcal{F}$, so that $A(t) - A(\tau) \leq E(t - \tau)$ for all $0 \leq \tau \leq t$ holds, $E(t)$ is an envelope function for $A(t)$, the maximal backlog B_{max} is bounded by

$$B_{max} \leq \sup_{\tau \geq 0} \{E(\tau) - \mathcal{S}(\tau)\}, \tag{2.4}$$

which corresponds to the maximal vertical deviation between $E(t)$ and $\mathcal{S}(t)$.

The delay of data arriving at time t is

$$W(t) = \inf\{w \geq 0 : A(t) - D(t + w) \leq 0\}.$$

With the envelope $E(t)$ the maximal delay bound becomes

$$W_{max} \leq \inf\{w \geq 0 : \sup_{\tau \geq 0} \{E(\tau) - \mathcal{S}(\tau + w)\} \leq 0\}, \tag{2.5}$$

which is the maximal horizontal deviation between $E(t)$ and $\mathcal{S}(t)$.

2.2.2 *System Models in the Max-Plus Algebra*

In the max-plus algebra, the arrivals are given by $T_A(n) \in \mathcal{G}^*$, which specifies the arrival times of packets with index $n = \{0, 1, 2, \cdots\}$. Accordingly, the departure timestamps are denoted as $T_D(n) \in \mathcal{G}^*$. The following description holds for unit size packets. A deterministic shift-invariant system offers the service defined by the service curve $T_S(n) \in \mathcal{G}$ if for all $n \geq 0$ the inequality

$$T_D(n) \leq \max_{\nu \in [0,n]} \{T_A(\nu) + T_{\mathcal{S}}(n-\nu)\} =: T_A \otimes T_{\mathcal{S}}(n) \tag{2.6}$$

holds [29]. For a max-plus linear system an exact service curve exists and Eq. (2.6) becomes an equality:

$$T_D(n) = \max_{\nu \in [0,n]} \{T_A(\nu) + T_{\mathcal{S}}(n-\nu)\}. \tag{2.7}$$

Since we assume discrete functions in the max-plus algebra as in [29], the convolution uses the maximum operation. Equivalent definitions exist for fluid traffic in the max-plus algebra e.g., in [74, Sec. 3.2.], where the maximum is substituted by the supremum.

The concatenation of H systems follows from the successive application of the convolution, i.e., $T_{\mathcal{S}_{net}}(n) = T_{\mathcal{S}_1} \otimes T_{\mathcal{S}_2} \otimes \ldots \otimes T_{\mathcal{S}_H}(n)$ as in the min-plus algebra. For further examples of systems, we refer to [29].

Below, we present the performance bounds for delay and backlog in the max-plus algebra as given in [29].

The delay of a packet with index n is

$$W(n) = T_D(n) - T_A(n).$$

If $T_E(n)$ is an envelope function for $T_A(n)$, so that $T_A(n) - T_A(\nu) \geq T_E(n-\nu)$ for all $0 \leq \nu \leq n$ holds, the maximal delay bound is

$$W_{max} \leq \max_{\nu \geq 0} \{T_{\mathcal{S}}(\nu) - T_E(\nu)\},$$

which is the maximal vertical deviation between $T_{\mathcal{S}}(n)$ and $T_E(n)$.

In [40], the backlog in the system for an arriving packet n is given by

$$B(n) = \min\{b \geq 0 : T_A(n) \geq T_D(n-b)\},$$

with envelope $T_E(n)$ the maximal backlog becomes

$$B_{max} \leq \min\{b \geq 0 : \max_{\nu \geq 0}\{T_{\mathcal{S}}(\nu - b) - T_E(\nu)\} \leq 0\},$$

which is the maximal horizontal deviation between $T_{\mathcal{S}}(n)$ and $T_E(n)$.

The comparison of the equations for backlog and delay calculation in both algebras shows that the derivation of the backlog is simpler in the min-plus algebra and the derivation of the delay is simpler in the max-plus algebra. In both cases it follows by subtraction.

2.2.3 *Connection between the Min-Plus and the Max-Plus Algebra*

The min-plus algebra and the max-plus algebra are two different algebras for the description of systems. In the min-plus algebra the domain of the functions is time and the codomain is the amount of data, whereas in the max-plus algebra it is reversed. In this work, we assume fluid traffic in the min-plus algebra and discrete traffic in the max-plus algebra. To state connections between fluid traffic and discrete traffic, we apply the concept of the packetizer $P^L(x)$ to fluid traffic. In [29, 74], the packetizer is defined by $P^L(x) = \sup_{i \in \mathbb{N}}\{L(i)1_{\{L(i) \leq x\}}\}$ with the cumulative arrivals x to the packetizer, the cumulative packet length function $L(i)$, where i is ith packet, $L(0) = 0$ by definition, and $1_{\{L(i) \leq x\}}$ is the indicator function, i.e., it is one if the expression $\{L(i) \leq x\}$ is true, else it equals zero.[6] The packetizer simplifies to $P^L(x) = \lfloor x \rfloor$ for unit size packets [29, 74], where $\lfloor x \rfloor$ means rounding down. The continuous functions of fluid traffic become right-continuous by the application of the packetizer.

PSEUDO-INVERSION FROM MAX-PLUS TO MIN-PLUS: To obtain the functions in the min-plus algebra, we utilize the cumulative service requirement defined

---

[6] Note that in Sec. 2.2.2 the index n of the first packet is $n = 0$ in the max-plus algebra, which translates into $i = n + 1$ for the definition of the packetizer.

in [29, Lem. 6.2.8], which determines the packets arriving (or departing) at (or from) the system. For unit size packets, the inversion for the arrivals $T_A(n) \in \mathcal{G}^*$ is

$$T_A^{-1}(t) = \sum_{n \geq 0} 1_{\{T_A(n) \leq t\}}. \tag{2.8}$$

For unit size packets an equal inversion is $T_A^{-1}(t) = \sup\{n \geq 0 : T_A(n) \leq t\} + 1$ since the cumulative length up to and including packet n is $n + 1$. Note that for $t < T_A(0)$ no packet has arrived and the sum equals zero. Eq. (2.8) yields a right-continuous function in the min-plus algebra.

For the inversion of a service curve according to Eq. (2.1)

$$T_{\mathcal{S}}^{-1}(t) = \inf\{n \geq 0 : T_{\mathcal{S}}(n) \geq t\}$$

is established in [29, Lem. 6.3.2]. The result is a left-continuous service curve that is deviant to the assumption of right-continuous functions assumed in this work.

PSEUDO-INVERSION FROM MIN-PLUS TO MAX-PLUS: To obtain functions in the max-plus algebra from functions in the min-plus algebra, we use the pseudo-inverse from [74, Sec. 3.1.4]. The inversion holds for the arrivals $A(t)$ as well as the departures $D(t)$. Here, we only show the inversion for the arrivals.

The arrivals in the max-plus algebra follow from arrivals in the min-plus algebra by

$$A^{-1}(n) = \inf\{t > 0 : A(t) \geq n + 1\}.$$

Since the cumulative length for packet index $n = \{0, 1, 2, \ldots\}$ is $n + 1$, the pseudo-inverse is defined for the time where $A(t) \geq n + 1$.

To our knowledge, no inversion of the service curve from the min-plus algebra to the max-plus algebra is established so far.

2.2.4 *Transformations in the Network Calculus*

In the classical system theory, transforms as the Laplace transform and the Fourier transform exist for linear time-invariant systems, which transform functions from the time domain into the frequency domain. For example, the Fourier transform shows the spectral densities in the frequency domain. These transforms are advantageous to solve the convolution operation because this operation becomes a multiplication in the frequency domain, which simplifies the computation for complex systems. The solution in the time domain is obtained by an inverse transformation from the frequency domain.

Corresponding transforms exist in the network calculus; these are the convex and concave Fenchel conjugates, for the definitions see [114]. The application of the Fenchel conjugates in the min-plus algebra and the analogy to the Fourier transform is extensively analyzed in [44]. The application of the Fenchel conjugates to communication networks is also shown in [6, 53, 101]. The domain of the Fenchel conjugates is called the rate domain in [43] since it decomposes a curve to its rate components. Below, we first state the Fenchel conjugates on the basis of service curves. Second, we list an abstract of properties of the transforms, which are beneficial for the following chapters, simplify the concatenation of systems, and are advantageous for the derivation of bounds on the service curve and of performance bounds.

TRANSFORM IN THE MIN-PLUS ALGEBRA: The transform is the convex Fenchel conjugate e.g., for the service curve:

$$\overline{\mathfrak{F}}_{\mathcal{S}}(r) = \sup_{t \in \mathbb{R}_0^+} \{rt - \mathcal{S}(t)\}.$$

If $\mathcal{S}(t)$ is differentiable the transform is called the Legendre transform. In reference to the network calculus, the variable r is the rate component of the function $\mathcal{S}(t)$.

transform in the max-plus algebra: The transform is the concave Fenchel conjugate e.g., for the service curve:

$$\underline{\mathfrak{F}}_{T_\mathcal{S}}(s) = \inf_{n \in \mathbb{N}_0} \{sn - T_\mathcal{S}(n)\}.$$

Strictly speaking, since $T_\mathcal{S}(n) \in \mathcal{G}$, we apply the discrete concave Fenchel conjugate [98]. The variable s corresponds to the reciprocal rate $1/r$ if applied in the framework of the network calculus.

properties of the transforms: Below, we list an abstract of properties of the transforms useful in the framework of the network calculus. For a comprehensive list see [44].

Convolution in the Rate Domain: Analog to the Fourier and the Laplace transform, the Fenchel conjugates simplify the concatenation of systems since the convolution becomes an addition in the rate domain for the min-plus algebra as well as the max-plus algebra, see [44, 53]. For example, in the min-plus algebra $D(t) = A \otimes \mathcal{S}(t)$ becomes $\overline{\mathfrak{F}}_D(r) = \overline{\mathfrak{F}}_A(r) + \overline{\mathfrak{F}}_\mathcal{S}(r)$.

Inverse of Fenchel Conjugates: The convex Fenchel conjugate is its own inverse, for closed convex functions it holds that $\overline{\mathfrak{F}}\{\overline{\mathfrak{F}}_\mathcal{S}(r)\} = \mathcal{S}(t)$. For non-convex functions the conjugate returns the convex closure of $\mathcal{S}(t)$, so that $\overline{\mathfrak{F}}\{\overline{\mathfrak{F}}_\mathcal{S}(r)\} \leq \mathcal{S}(t)$. Equivalently, the concave Fenchel conjugate is its own inverse for closed concave functions, or for non-concave functions it yields the concave closure with $\underline{\mathfrak{F}}\{\underline{\mathfrak{F}}_{T_\mathcal{S}}(r)\} \geq T_\mathcal{S}(n)$.

Performance Bounds from Fenchel Conjugates: If constant rate arrivals are assumed, i.e., $A(t) = rt$ and $T_A(n) = \frac{n+1}{r}$, the maximal backlog results from the convex Fenchel conjugate of the min-plus service curve $B_{max} \leq \overline{\mathfrak{F}}_\mathcal{S}(r) = \sup_{t \in \mathbb{R}_0^+}\{rt - \mathcal{S}(t)\}$ and the delay from the concave Fenchel conjugate of the max-plus service curve $W_{max} \leq -\min_{n \in \mathbb{N}_0}\left\{\frac{n}{r} - T_\mathcal{S}(n)\right\} = \max_{n \in \mathbb{N}_0}\{T_\mathcal{S}(n) - \frac{n}{r}\}$.

2.3 RANDOM SYSTEMS

The previous section presents system models defined in the deterministic network calculus by two algebras, which are closely related. These system models build the basis for this section that presents system models for random system. Even if following equations are similar, the difference is in the details. For the min-plus algebra, we omit the computation of the delay, and for the max-plus algebra, we omit the computation of the backlog. As presented before, in each case the computation is beneficial in the respective other algebra.

We introduce two concepts for the description of random systems, for which we follow the classification of [30], where these two concepts treat service curves as random processes or as non-random functions. In the former case, bivariate functions capture time- or shift-variant characteristics of the systems, where the functions are interpreted as random processes. In the latter case, the random systems are described by invariant functions specifying the variability as a probabilistic bound.

2.3.1 *System Models as Stochastic Processes*

In this section, we present the basics for time- and shift-variant systems in the network calculus, i.e., the assumption of an invariant mapping operator for the system description does not apply. We follow the perception of [30] and assume the following functions represent stochastic processes. Originally, this framework was introduced as a deterministic extension of the network calculus for time-variant systems in the min-plus algebra in [29, Chap. 5]. To account for time-variance the univariate functions of the arrivals, the departures, and the service curves from the deterministic network calculus, which are presented in Sec. 2.2, are extended to bivariate functions. These functions depend on the time instances (τ, t) instead of on the length of the time interval $(t - \tau)$. The extensions for shift-variant systems in the max-plus algebra are presented in [68]. The functions become bivariate functions depending on the packet indexes (ν, n) instead of the amount of data.

This system description by bivariate functions is less intensively studied than the description for deterministic systems presented in Sec. 2.2. Analog to the systems described in Sec. 2.2 performance bounds exist, but in contrast, the connection between the min-plus and the max-plus algebra, see Sec. 2.2.3, and the transforms, see Sec. 2.2.4, are not declared so far. The use of bivariate functions and the lack of corresponding transforms make the declaration hard. For example, for the deterministic systems described in Sec. 2.2 the property of invariance, which is missing for the bivariate functions, is utilized in [29, Lem. 6.2.8] to connect the descriptions in the max-plus and the min-plus algebra.

2.3.1.1 *System Models in the Min-Plus Algebra*

For the description, we abbreviate the arrivals $A(0,t)$ and departures $D(0,t)$ by $A(t)$ and $D(t)$. The functions $A(\tau,t) = A(t) - A(\tau) \in F$ and $D(\tau,t) = D(t) - D(\tau) \in F$ specify the arrivals and departures, respectively, in the time interval $(\tau,t]$. A system offers the service defined by the service curve $S(\tau,t) \in F$ if the inequality

$$D(t) \geq \inf_{\tau\in[0,t]} \{A(\tau) + S(\tau,t)\} =: A \otimes S(t), \tag{2.9}$$

is satisfied for all $t \geq 0$. For linear systems, an exact service curve exists and the inequality becomes an equality:

$$D(t) = \inf_{\tau\in[0,t]} \{A(\tau) + S(\tau,t)\}. \tag{2.10}$$

This extension for discrete time systems is introduced in [29, Chap. 5], where the infimum becomes a minimum.

The service curve of a network path that consists of H hops follows from

$$S_{net}(\tau,t) = S_1 \otimes \cdots \otimes S_H(\tau,t). \tag{2.11}$$

A fundamental difference to the invariant systems is that for the concatenation of systems the commutative property does not hold, i.e., $S_1(t) \otimes S_2(t) \neq S_2(t) \otimes S_1(t)$.

Analog to the deterministic systems, performance bounds exists. The backlog at time $t \geq 0$ is

$$B(t) = A(t) - D(t).$$

from which the backlog bound

$$B(t) \leq \sup_{\tau \in [0,t]} \{A(\tau, t) - S(\tau, t)\}$$

follows with Eq. (2.9). For all t the backlog is bound by [29, Thm.5.5.5]:

$$B_{max} \leq \sup_{t \geq 0} \left\{ \sup_{\tau \in [0,t]} \{A(\tau, t) - S(\tau, t)\} \right\}.$$

Compared to the backlog bound for deterministic system given in Eq. (2.4), this bound has to be evaluated for τ and t and does not simplify due to the usage of bivariate functions.

2.3.1.2 *System Models in the Max-Plus Algebra*

The max-plus algebra for deterministic shift-invariant systems is also extended to shift-variant systems in [68]. The functions $T_A(\nu, n) = T_A(n) - T_A(\nu) \in G$ and $T_D(\nu, n) = T_D(n) - T_D(\nu) \in G$ specify the arrivals and departures in the max-plus algebra, respectively, according to the packet indexes $[\nu, n]$. The service curve $T_S(\nu, n) \in G$ for all $n \geq 0$ is defined by the inequality:

$$T_D(n) \leq \max_{\nu \in [0,n]} \{T_A(\nu) + T_S(\nu, n)\} =: T_A \otimes T_S(n). \tag{2.12}$$

For linear systems, the inequality becomes an equality

$$T_D(n) = \max_{\nu \in [0,n]} \{T_A(\nu) + T_S(\nu, n)\} \tag{2.13}$$

with an exact service curve.

The delay of packet n is

$$W(n) = T_D(n) - T_A(n),$$

and inserting Eq. (2.12), the delay $W(n)$ is bounded by

$$W(n) \leq \max_{\nu \in [0,n]} \{T_S(\nu, n) - T_A(\nu, n)\}.$$

For all n, the bound becomes

$$W_{max} \leq \max_{n \geq 0} \{ \max_{\nu \in [0,n]} \{T_S(\nu, n) - T_A(\nu, n)\}\}.$$

2.3.2 *System Models as Probabilistic Bounds*

The deterministic system theory in Sec. 2.2 derives worst-case performance bounds and requires the existence of deterministic envelopes for arrival traffic. In random systems, non-trivial worst-case bounds potentially do not exist. If the sample space of the applied probabilistic models is infinite, no finite bounds exist for the models e.g., for the exponential distribution used in the $M|M|1$ queueing system [70] and Internet traffic models as [76, 106]. Furthermore, delay sensitive applications allow small violations of performance bounds, e.g., see [41]. Since the service curve in Sec. 2.3.1 assumes random processes, the performance bounds hold for realizations of these processes and are in this sense worst-case performance bounds for the specific realizations. The stochastic network calculus overcomes the drawbacks of the deterministic network calculus and meets the requirements of random systems and applications by describing service and traffic as probabilistic bounds, i.e., the variability is described by invariant functions in combination with a violation probability with respect to these functions, see e.g. [15, 29, 61]. Compared to the approach for random systems described in Sec. 2.3.1, the probabilistic bounds comprise a complete description of the random system and not only of a specific realization. This approach allows the derivation of probabilistic bounds on the arrivals, the service curve, and the departures and with that also on the performance bounds delay and backlog. Here, we limit the description to the ε-effective service curve developed in [25] with related performance bounds in the min-plus algebra and the equivalent description in the max-plus algebra, which is given in [132]. For the Fenchel transform, we show its applicability to systems in the min-plus algebra in [87]. Since the transform is already stated in Sec. 2.2.4, we omit it in this section.

Till this day, the connection between the max-plus and the min-plus algebra has not been stated for these system descriptions since the probabilistic extension makes the connection difficult. We prove a connection in Sec. 5.3.3 that is sufficient for the application of our system identification approach.

2.3.2.1 *System Models in the Min-Plus Algebra*

With the arrivals $A(t) \in \mathcal{F}$ and departures $D(t) \in \mathcal{F}$ a system has an ε-effective service curve $\mathcal{S}^{\varepsilon}(t) \in \mathcal{F}$ if

$$\mathsf{P}\left[D(t) \geq \inf_{\tau \in [0,t]} \{A(\tau) + \mathcal{S}^{\varepsilon}(t-\tau)\}\right] \geq 1 - \varepsilon, \tag{2.14}$$

applies for all $t \geq 0$, where ε is the violation probability [25].

The service curve is defined by a probabilistic bound on the departures. The convolution $A \otimes \mathcal{S}^{\varepsilon}(t)$ is equal to or smaller than $D(t)$ with a probability of $1 - \varepsilon$ at least. In addition to the definition of the service curve, [25] also contains performance bounds for the ε-effective service curve.

With deterministic envelope $E(t) \in \mathcal{F}$ (see Sec. 2.2.1) for the arrivals $A(t)$ the backlog is bounded for all $t \geq 0$ by

$$\mathsf{P}\left[B(t) > \sup_{\tau \geq 0}\{E(\tau) - \mathcal{S}^{\varepsilon}(\tau)\}\right] \leq \varepsilon.$$

2.3.2.2 *System Models in the Max-Plus Algebra*

The equivalent approach to the min-plus ε-effective service curve in the max-plus algebra is introduced in [132]. The definition of the ε-effective service $T^{\varepsilon}_{\mathcal{S}}(n) \in \mathcal{G}$ in the max-plus algebra follows from the relation

$$\mathsf{P}\left[T_D(n) \leq \max_{\nu \in [0,n]} \{T_A(\nu) + T^{\varepsilon}_{\mathcal{S}}(n-\nu)\}\right] \geq 1 - \varepsilon, \tag{2.15}$$

for all $n \geq 0$ with the arrivals $T_A(n) \in \mathcal{G}^*$ and the departures $T_D(n) \in \mathcal{G}^*$. The performance bounds are derived as probabilistic bounds on the service as well as on the arrivals in [132]. Here, we present performance bounds according to [25] for random service but deterministic arrivals that are bounded by $T_A(\nu, n) \geq T_E(n-\nu), \forall 0 \leq \nu \leq n$.

With packet delay $W(n) = T_D(n) - T_A(n)$ and applying Eq. (2.15) for $T_D(n)$, the delay becomes

$$\mathsf{P}\left[W(n) > \max_{\nu \in [0,n]} \{T_{\mathcal{S}}^{\varepsilon}(n-\nu) - T_A(n-\nu)\}\right] \leq \varepsilon.$$

Using the envelope $T_E(n-\nu)$, the delay is bounded for all $n \geq 0$ by

$$\mathsf{P}\left[W(n) > \max_{\nu \geq 0} \{T_{\mathcal{S}}^{\varepsilon}(\nu) - T_E(\nu)\}\right] \leq \varepsilon.$$

This chapter introduces service curves for deterministic networks and networks with random service. The next chapter presents related work to system identification, in which among others the system model of a deterministic system is determined by system identification. The service curves presented for networks with random service are used to establish our system identification methodology in Chap. 5.

3

STATE OF THE ART IN SYSTEM IDENTIFICATION OF COMPUTER NETWORKS

Numerous approaches exist for system identification of computer networks. The approaches differ e.g., in their probing methodology, i.e., which kind of data traffic is sent to probe the network, the assumed model class, which may include classes for deterministic and random systems with assumptions on the scheduling disciplines, or the used framework, as queueing theory or the network calculus. In the following, we present the state of the art in end-to-end active measurement approaches for system identification of computer networks. End-to-end active measurement approaches send actively probing traffic with specific traffic characteristics into the network and observe the packets at the ingress and the egress of the network path to identify the system. On the contrary, passive measurement approaches exist, which rely on capturing production traffic. Here, we focus on active measurement approaches. These are essential for system identification of computer networks, see e.g., [16, 91], because passive approaches are only able to see characteristics that are imprinted in the captured traffic. Active approaches have a greater flexibility for the identification by using specific probing traffic.

We add available bandwidth estimation approaches to the field of system identification because they describe the systems usually by a single value. In terms of the network calculus, the system model of the available bandwidth is given by the service curve $S(t) = \alpha t$, where α is the available bandwidth estimate. We make this intuitive relation between the available bandwidth and the framework of the network calculus explicit in Chap. 6.

System identification in the classical system theory distinguishes between gray-box and black-box models [84]. For gray-box models, the internal system structure is known, which allows the creation of an analytical model with free parameters. The identification process is constrained to the estimation of these free parameters. From this characterization it follows that available bandwidth estimation approaches, which characterize the system by one value, use a gray-box model with one free parameter. For black-box models no internal structure is predetermined. The complete system model is identified by measurements. It is thereby not limited to any specific structure of the system.

In system identification known from the classical system theory, systems are stimulated by, e.g., the Dirac impulse, step functions, and sine waves to measure the response of the systems and subsequently deduce the system model from it [84]. In system identification for computer networks, systems are stimulated by data traffic with different characteristics. In the following, we present the approaches known from the literature and summarize their probing procedures.

We start with the description of available bandwidth estimation approaches in Sec. 3.1. First, we present approaches that assume the network is a deterministic system. Second, we describe approaches that consider networks featuring randomness due to random cross traffic. At last, we show available bandwidth approaches for wireless networks. In the first two cases, the system models base on simplifications for wired networks, i.e., if the models account for randomness, it is because of random cross traffic, and the scheduling is often assumed to be work-conserving and FIFO. These assumptions do not apply to wireless networks with non-work-conserving, non-FIFO scheduling, and non deterministic channels.

In Sec. 3.2, we present existing system identification approaches that utilize the framework of the deterministic network calculus from Chap. 2. Few approaches exist so far that rely on this system theory for computer networks. We introduce approaches, which use a gray-box model for the characterization of routers and black-box models for the identification of end-to-end network paths.

3.1 AVAILABLE BANDWIDTH ESTIMATION

The available bandwidth describes the portion of the capacity of a network path that is unused by existing traffic to which we refer as cross traffic. For a single-hop network with one link h in a time interval $[t, t+\delta]$, the available bandwidth is defined in [82] by

$$\alpha_h(t, t+\delta) = \frac{1}{\delta} \int_t^{t+\delta} C_h(x)(1 - u_h(x))dx, \tag{3.1}$$

where $C_h(t)$ is the capacity of the link and $u_h(t) \in \{0, 1\}$ is its utilization at time t by cross traffic with rate $\frac{1}{\delta} \int_t^{t+\delta} C_h(x) u_h(x) dx$. Further, if the cross traffic has a long-term rate, which is defined as $\lambda_h := \limsup_{t \to \infty} \frac{A(t)}{t}$, where $A(t)$ are the cumulative arrivals up to time t, we refer to $\alpha_h^\infty = \liminf_{\delta \to \infty} \alpha_h(t, t+\delta) = C_h - \lambda_h$ as the *long-term available bandwidth*.

The available bandwidth of a network path that consists of multiple links is defined by the minimum of the available bandwidths of the individual links [59, 60, 83]:

$$\alpha_{net}(t, t+\delta) = \min_{h \in H} \{\alpha_h(t, t+\delta)\}, \tag{3.2}$$

where H is the set of links of the path. For the estimation of the available bandwidth a comprehensive number of approaches exists. These approaches assume different system models to establish the estimation. Below, we introduce these approaches in respect to the assumed system model and the consequential restrictions.

3.1.1 *Single-hop Network with Fluid Constant Rate Traffic and Constant Capacity*

In the following model of a deterministic system, the link capacity C and also the cross traffic rate λ are assumed to be constant over time and the traffic is assumed to be fluid. The available bandwidth at a single-hop network with link capacity C that is utilized by λ follows from $\alpha_h(t) = C - \lambda$. This fluid constant rate network model is widely used for available bandwidth estimation approaches [38, 54, 59, 94, 111, 122]. Obviously, this model does not apply for most practical networks due to non deterministic characteristics of traffic, but most tools are aware of the variability of the available bandwidth. At the end of this section, we present how the tools approach the contradiction between the deterministic system assumption and the variability in practical networks.

For the estimation of the available bandwidth, tools transmit pairs or trains of probe packets with defined gaps between the packets via a network path and measure the gap or the rate of the packets at the egress of the path. The use of constant rate packet pairs or packet trains and the additional assumption of FIFO scheduling in a single-hop network leads to the gap response curve and the rate response curve, respectively. These curves specify the relation between the dispersion of the packets or the rate at the ingress and the egress of a system. The relation of the input gap g_I and the output gap g_O of a packet pair is given in [54, 82] as:

$$g_O = \begin{cases} g_I & , g_I \geq \frac{s}{C-\lambda} \\ \frac{s+g_I\lambda}{C} & , g_I < \frac{s}{C-\lambda} \end{cases}, \tag{3.3}$$

where C is the capacity, λ the cross traffic rate and s the packet size. Accordingly, the rate response curve with arrival rate r_I and departure rate r_O of a packet train is in [82, 95] defined by:

$$r_O = \begin{cases} r_I & , r_I \leq C - \lambda \\ C\frac{r_I}{r_I+\lambda} & , r_I > C - \lambda \end{cases}. \tag{3.4}$$

For both response curves applies that the outputs of the network are equal to the inputs as long as the arrival rate is less or equal to the available bandwidth. If the probing rate exceeds the available bandwidth, the departure rate r_O equals the portion of the arrival rate r_I as part of the total arrival rates $r_I + \lambda$. This relation follows from the assumption of FIFO scheduling.

In [60], available bandwidth approaches are classified by direct probing and iterative probing. For the direct approach, the link capacity must be known or estimated separately. With the known capacity and one probing rate above the available bandwidth the segment for which r_I exceeds the available bandwidth follows from the estimation. The estimate of the available bandwidth results from the intersection with the known segment for which r_I is below the available bandwidth. Thus, for the direct probing approaches, a single probing rate is sufficient. The iterative approach makes use of the two segments of the response curves from Eq. (3.3) and Eq. (3.4) by iteratively adapting the probing rate. If a transition from the first segment of the response curve, where the outputs equal the inputs, to the second segment is detected, the available bandwidth is exceeded. In this way, the tools estimate the available bandwidth. The system models presented in Eq. (3.3) and Eq. (3.4) also imply that the one way delays are constant if the probing rate is below the available bandwidth and increase if it is above due to queueing in the network.

Available bandwidth estimation tools that use a direct approach are e.g., IGI [54], and Spruce [122], and tools that use an iterative approach are, e.g., BART [38], Pathchirp [111], Pathload [58, 59], PTR [54], and TOPP [94]. The tools usually use constant rate packet probes. One exception is Pathchirp, which uses packet trains, at which the gap between successive packets decreases due to a geometric progression. Tools that use an iterative approach and detect the transition between the segments of the response curve are also called congestion inducing techniques [111] since the transition is caused by congestion.

The tools Pathload and Pathchirp do not require FIFO scheduling since they detect the increase of one way delays by iteratively adapting the probing rate as elaborated in [20]. Therefore, they do not rely on one of the response curves, the assumption of work-conserving scheduling is sufficient, i.e., the delays increase only if the arrival rate exceeds the available bandwidth.

In reality, traffic is random, see e.g., [76], and packetized. These effects disturb the clear shape of the two segments of the response curves. To overcome these disturbances, available bandwidth estimation tools use post processing of the measurement results as e.g., averaging over estimates of several packet pairs or packet trains sent at the same rate [54, 58, 59, 79, 111, 122], linear regression to reconstruct the shape of the response curve [94], or a Kalman filter to eliminate the disturbance created by random traffic [38, 119]. Furthermore, the tool Pathload returns the available bandwidth as a range to account for the variability of it.

The approaches presented in this section rely on deterministic systems and may be applicable to networks with low variability. Compared to the system description from Fig. 1.1, these approaches interpret randomness as disturbance. The negative impact on the estimate of the mismatch between the assumption of a deterministic system and practical networks is e.g., presented in [60]. The next section presents approaches that consider randomness due to cross traffic.

3.1.2 *Networks with Random Traffic and Constant Link Capacity*

In Sec. 3.1.1, we discuss a system model that assumes a deterministic system with constant rate fluid traffic. Available bandwidth estimation tools that are based on this model mitigate effects of randomness by post processing. Below, we present approaches that consider the randomness of cross traffic but preserve the assumptions of constant link capacities and FIFO scheduling.

The impact of random and packetized traffic on the rate response curves is analyzed with a queueing theoretic framework for single-hop and multi-hop networks, e.g., in [35, 52, 82, 83, 104]. The gap response curve and the rate response curve from Eqs. (3.3) and (3.4) are a lower bound and an upper bound, respectively, for the relation between inputs and outputs in a single-hop network with FIFO scheduling with random traffic as shown in [82]. The authors of [82] also prove that the estimate approaches these bounds with increasing length of the packet size or the packet trains. In [83], the work is extended to multi-hop networks, in which the authors show that the gap response curve is again a lower bound. This bound can also be approached by increasing the packet size or the length of packet trains.

In [35, 52, 104], the distribution of the output gap is derived under the assumption of a known queueing model as $M|D|1$ in [104], $M|G|1$ in [52], and a known cross traffic model in [35]. In [52], the empirical distribution of the output dispersion is fitted to the analytical model to estimate the available bandwidth.

The estimation of the random cross traffic process from delay measurements of probe packets is defined as an inversion problem in [91]. It is shown that an inversion, which retrieves the complete process, for a single-hop network with FIFO scheduling, cross traffic with stationary independent increments, and a renewal process as probing process is possible. Besides this scenario, limitations of active probing are presented and estimators are defined and evaluated, also for conditions where the before mentioned constraints are not met.

In [124], an approach for available bandwidth estimation based on the definition

$$\alpha := \max\{r_I : \mathsf{P}[r_O > r_I - r_\delta] \geq \gamma\},$$

is presented, where α is the available bandwidth, r_I is the arrival rate, r_O is the departure rate, r_δ defines a tolerance for the difference between the arrival and the departure rate and γ is a bound on the probability. The parameters r_δ and γ are user defined and are based on application requirements. This approach assumes also FIFO scheduling, but it also applies to other scheduling disciplines.

The available bandwidth estimation approaches presented in this section account for random cross traffic in the network. Still, most of them rely on assumptions as specific queueing models and traffic models. One exception is the approach presented in [124], which is more general in terms of the assumptions but is specific in the selection of the free parameters. These are specific for the application and are not embedded in any framework as queueing theory or the network calculus. Thus, it prevents the derivation of general results as backlog and delay bounds. The next section presents available bandwidth estimation approaches for networks to which usually defined assumptions such as links with fixed capacity and FIFO scheduling do not apply, namely wireless IEEE 802.11 networks.

3.1.3 *Wireless Networks with Random Service and non FIFO Scheduling*

A broad range of available bandwidth estimation approaches are developed for wireless IEEE 802.11 networks since these networks are wide-spread and feature challenging characteristics as a time-varying channel and an approximately fair and non-work-conserving scheduling discipline. This stands in contrast to wired networks, where usually constant link capacities and work-conserving FIFO scheduling are assumed[7]. Wireless networks are systems with random service due to random cross traffic and due to time-varying link capacities of the wireless medium caused by interference. Furthermore, IEEE 802.11 networks have a complex medium access

---

[7] For wired networks, we refer to Ethernet based (IEEE 802.3) networks with the prevalent scheduling discipline FIFO at intermediate hops.

control (MAC) in comparison to wired networks. The medium access is controlled by the distributed coordination function (DCF), and an automatic repeat request (ARQ) protocol is used to retransmit packets that are corrupted during transmission. The MAC leads to non-work-conserving and approximately fair scheduling [21, 67], which is a fundamental difference to wired networks where schedulers are typically work-conserving. The non-work-conserving behavior results from the DCF, where a station has to wait for a random time before it tries to access the channel, and due to the ARQ protocol, where the next packet can only be sent if an acknowledgement is received. Recall that, the response curves defined in Eqs. (3.3) and (3.4) hold for work-conserving FIFO scheduling. In the following, we present available bandwidth estimation tools designed for wireless IEEE 802.11 networks.

The available bandwidth estimation tool Wbest [78] considers time-varying link capacities by measuring the capacity using the median of gaps of a sequence of back-to-back probe packets. Nevertheless, the median is not concluded from a system model. On the contrary, the minimum over the gaps is used in wired networks [55], which follows from the assumption of constant rate links. Wbest assumes FIFO scheduling for measuring traffic in the downstream from an access point to a wireless station, thus, it relies on the rate response curve from Eq. (3.4).

Adaptations of the rate response curve, which is given Eq. (3.4), for IEEE 802.11 networks are made in [20, 108]. In [20], the rate response curve is specified under the assumption of fluid constant rate traffic and fair scheduling. The gap response curve, see Eq. (3.3), is extended for fluid constant rate traffic and fair scheduling plus a queue with FIFO scheduling to model competing traffic on the same node and random access delays in [108].

The tools TOPP [94] and BART [38] are extended for wireless IEEE 802.11 networks in [65] and [62], respectively, by taking into account the access delays of the medium access control under the assumption of FIFO scheduling. In [62], the Kalman filter is adapted for wireless networks. It is also employed in [21] for a passive available bandwidth estimation approach for IEEE 802.11 networks utilizing a generalized processor sharing model.

IEEE 802.11 networks have challenging characteristics in the perspective of available bandwidth estimation. Tools address them by customizing their estimation methodology to these characteristics, e.g., by the adaptation of the post processing of results or by the adaptation of the response curves. The next section contains system identification approaches that are aimed at accounting for a broad range of networks by avoiding customized system models, which only apply to specific networks. The approaches, presented in the next section, include also the time of the service availability to the system description. This stands in contrast to available bandwidth estimation tools, which describe the system usually by a single value and thereby disregard the time scale.

3.2 SYSTEM IDENTIFICATION IN THE NETWORK CALCULUS

The network calculus is a system theory for computer networks. So far, only few approaches were developed directly in this framework for system identification. Next, we outline the existing approaches from [6, 22, 53, 80, 125]. In the network calculus, the system model of the network path is a service curve as introduced for systems with various characteristics in Chap. 2. The service curve describes the mapping from the inputs to the outputs of the system.

A system identification approach for router modeling, which relies on a gray-box model, is presented in [22, 125]. Service curves known from the deterministic network calculus (see Sec. 2.2) are assumed as system models, in detail the approach is based on service curves of the guaranteed rate model and the packet scale rate guarantee model for which service curves exist in the network calculus. The estimation approach is presented in [125] and relies on passive measurements. It is extended in [22] for active measurements. Therein, the approach is used to analyze the performance of various routers as hardware and software routers and the impact of virtualization on software routers. The use of service curves for router modeling extends the standard performance metric, i.e, the packet forwarding rate, with timing information such as jitter and delay in a coherent representation.

System identification approaches using a black-box model in the framework of the deterministic network calculus are discussed in [6, 53, 80]. Using such black-box models, makes the identification of a broad range of networks possible without relying on specific models. These approaches include fundamental networks as single-hop networks, network paths with multiple bottleneck links, and various scheduling disciplines, in which the system is described by a service curve, see [80].

In [6, 53], system identification approaches are sketched based on the Fenchel transform. The work in [80] presents an extensive analysis of probing approaches used for available bandwidth estimation, which are on the one hand passive measurement approaches, and on the other hand the approaches implemented by the tools Pathchirp [111] and Pathload [58, 59].

For passive measurements, it is shown in [80] that for non-linear networks it is only possible to achieve a not useful bound on the service curve.

Active measurement approaches have the freedom to control the arrival traffic and thereby can avoid regimes where the network shows a non-linear behavior. For example, such regimes occur in overloaded links with FIFO scheduling.

One active measurement approach for available bandwidth estimation, which is analyzed in [80] using the framework of the network calculus, is based on the probing methodology of the tool Pathchirp [111]. The tool probes the network path by packet trains, in which the gap between packets is decreasing according to a geometric progression. Using the convex Fenchel transform, see Sec. 2.2.4, of the probe traffic arrivals $\overline{\mathfrak{F}}_A$ and the departures $\overline{\mathfrak{F}}_D$, the estimate of the service $\mathcal{S}(t)$ is

$$\widetilde{\mathcal{S}}(t) = \overline{\mathfrak{F}}(\overline{\mathfrak{F}}_D - \overline{\mathfrak{F}}_A).$$

Another active measurement approach analyzed in [80] is based on the tool Pathload [58, 59], which employs packet trains with constant rate to probe the network. The final estimate results from repeated packet trains with several probing rates. It is shown that a service curve $\mathcal{S}(t)$ can be estimated with constant rate arrivals $A(t) = rt$ from the measurement of the related maximal backlog $B_{\max}(r) = \sup_{\tau \geq 0}\{r\tau - D(\tau)\}$ in min-plus linear deterministic systems by

$$\widetilde{\mathcal{S}}(t) = \sup_{r \geq 0}\{rt - B_{\max}(r)\} = \overline{\mathfrak{F}}_{B_{max}}(t).$$

Note that the estimate is also the Fenchel transform of $B_{max}(r)$, thus, also the properties of the Fenchel transforms apply. By controlling the arrival rate and observing the departures, it can be avoided that arrival rates that push the network into a non-linear regime are considered by the estimate. Furthermore, experimental validation in [80] shows that the measuring approach of the tool Pathload is more reliable than the approach of Pathchirp.

The presented system identification approaches, which utilize the deterministic network calculus, demonstrate the applicability of the network calculus to system identification of computer networks. These approaches can account for service availability on various time scales by using the service curve for the system description. Especially, black-box models allow a great freedom since only basic assumptions are made on the underlying system structure. One major restriction made by these presented approaches is that the assumed system is deterministic. Before, we introduced approaches from available bandwidth estimation in this chapter. These approaches include, among others, systems with random service but are often tailored to specific networks and ignore the time scale of the availability. The next chapter presents in detail the aim of this work to establish a system identification approach for networks with random service.

4

PROBLEM STATEMENT

In general, systems map inputs to outputs. In the context of computer networks, inputs are data arrivals and outputs are data departures. In the network calculus, which is a system theory for computer networks (see Chap. 2), service curves describe the system model that specifies the mapping from the inputs to the outputs.

Much research work has been conducted for systems with known properties of the arrivals and the service and their interconnection. For example, such properties are statistical models of network traffic, models of channels, or protocol specifications.

Little is known about system identification of networks with random service, i.e., characterizing the system model from measurements of the inputs and outputs.

The inclusion of available bandwidth estimation approaches to system identification imports a comprehensive set of approaches to system identification. However, approaches for deterministic networks are prevalent in the field of available bandwidth estimation as presented in Sec. 3.1.1. They attempt to account for randomness by post-processing. Approaches for networks with random service also exist, which integrate randomness in their system model. Typically, these are customized to particular networks. Common assumptions are networks with fixed capacity, random cross-traffic, and FIFO scheduling (see Sec. 3.1.2). Another set of assumptions originates from IEEE 802.11 networks that feature random link capacities as well as random cross traffic with non-work-conserving scheduling and non-deterministic channel access (see Sec. 3.1.3). One exception is [124], which presents a definition of the available bandwidth for networks with random service. Since the definition is not embedded in any analytical framework, it is tailored to a specific parametrization, which limits the informative value of estimates relying on this definition.

Furthermore, system identification approaches exist in the network calculus. The network calculus adds the time of availability of the service to the characterization of the system by service curves. They allow for the coherent description of time and rate on arbitrary scales. On the contrary to that, available bandwidth approaches usually describe the system by only a single value. So far, only a subset of the framework, namely the network calculus for deterministic networks, is exploited for system identification as discussed in Sec. 3.2. These approaches show that fundamental information is contained in packet probes, which is usually not considered by available bandwidth estimation approaches. However, the network calculus comprises a broader set of networks than deterministic networks. For this reason, we address the challenge of:

System Identification of Computer Networks with Random Service.

Solving this challenge establishes an inversion from measurements to service curves that account for randomness by the system model. Furthermore, since the service curve is embedded in the framework of the network calculus, this enables the re-utilization of the results in this well-established framework, e.g., for calculating performance bounds.

The challenge of this work is to define a methodology for system identification so that it fulfills requirements from service curve definitions for random systems established in Chap. 2 and to transfer this methodology to practical probing procedures.

Such system identification procedures can provide solutions where analytical derivations fail or are hard to derive. For example, modeling service curves for well-established Internet traffic models in wired networks is challenging [81, 113] but possible. If such characteristics are unknown, an analytical derivation becomes impossible. Moreover, available bandwidth estimation often aims at wired networks. In such networks, the links can be assumed to feature a constant rate. This is caused by the fact that protocols as Ethernet and the Internet Protocol, which add their header to the payload of upper layers, hardly impact the variability of the traffic. Already for wireless networks the medium access control protocol has a strong impact on the variability. Few available bandwidth approaches are adapted to wireless networks by relying on properties specific to such networks. Offering a universal system identification approach, which is not tailored to specific networks or protocols, establishes the application to a comprehensive range of networks and complex protocols without detailed knowledge of the systems being required.

In practical networks, further effects occur due to implementations of protocol stacks, buffer limitations, coarse grained timing of packet schedulers, segmentation of packets, etc. These effects are usually not captured by analytical approaches. A universal system identification approach is able to evaluate systems with unknown parameters, estimate impacts due to practical networks, and comprise all details of network paths.

Using the system models from the network calculus can also lead to a better understanding of bandwidth estimation, which we perceive as a subarea of system identification. Service curves explain bandwidth estimation from a system theoretic point of view and interpret known effects, which are not well understood to this day.

In the next sections, we develop an analytical system identification methodology, implement it into practical probing procedures, evaluate and validate it by extensive experiments, and apply the system identification to various systems to obtain a system model. Furthermore, we establish the connection between system identification in the framework of the network calculus to available bandwidth estimation.

5

A SYSTEM IDENTIFICATION METHODOLOGY FOR COMPUTER NETWORKS WITH RANDOM SERVICE

The formulation of a system identification methodology in the network calculus for linear systems with random service allows the identification for a broad range of practical networks by the use of a black-box model. This comprises networks as single-hop and multi-hop networks with random traffic and also networks that use complex protocols e.g., as the medium access control protocol in IEEE 802.11 networks or TCP. In the following, we derive such a methodology, which is based on the service curves introduced in Sec. 2.3.1 and Sec. 2.3.2.

In Sec. 5.1, we connect two concepts from the network calculus to describe random service, namely the bivariate function $T_S(\nu, n)$, see Sec. 2.3.1, and the ε-effective service curve $T_S^{\varepsilon}(n)$ from the stochastic network calculus, which is introduced in Sec. 2.3.2. The bivariate functions describe the service as a random process, whereas the ε-effective service curves give practical probabilistic bounds on the service. Since the bivariate function represents a random process, we refer to this type of a service curve as a service process for a clear differentiation between these system models. With this connection we state our system identification methodology in Sec. 5.2 that enables the derivation of ε-effective service curves for networks with random service from measurements of end-to-end delays by using packet trains. The methodology of [80] serves as an initial starting point, where an inversion method from backlog to a service curve is elaborated for deterministic networks.

We formulate the system identification methodology in the max-plus algebra of the stochastic network calculus. The formulation in this algebra has practical advantages for probing because the methodology is based on packet delays. It directly utilizes packet timestamps included in probe packets for the calculation of the delays and can express lost packets as packets with infinite delay. Still, the min-plus algebra is often used in the network calculus, which is motivated by the straightforward computation of multiplexing traffic flows. From this it follows that having the result of the identification methodology available in the min-plus algebra is also beneficial due to the existing comprehensive framework. Therefore, we show that the system identification methodology can also be established in the min-plus algebra in Sec. 5.3. Moreover, we make use of the min-plus algebra in Chap. 6 to show the connection between the available bandwidth definition and the service curves from the network calculus. To combine the advantages of both algebras, we further formulate a pseudo-inversion from the max-plus algebra to the min-plus algebra. This inversion combines the advantages existing in both algebras: the system identification methodology relies on the max-plus algebra, and it makes the results available in the min-plus algebra.

The work in this chapter is based on joint work with Markus Fidler and Jörg Liebeherr. It is partially available in [88, 89, 90].

5.1 CONNECTION BETWEEN THE SERVICE CURVES FOR RANDOM SYSTEMS

Sec. 2.3.1 and Sec. 2.3.2 present two different approaches for the description of random systems. The service process described in Sec. 2.3.1 covers the variability by a bivariate function $T_S(\nu, n)$. It is defined by the relation between the arrivals $T_A(n)$ and the departures $T_D(n)$ in Eq. (2.13) for linear networks by

$$T_D(n) = \max_{\nu \in [0,n]} \{T_A(\nu) + T_S(\nu, n)\} .$$

Drawing general conclusions from $T_S(\nu, n)$ is limited since it represents a specific sample path of the stochastic process. A more practical approach is the ε-effective service curve $T_S^\varepsilon(n)$, see Eq. (2.15). It is a shift-invariant probabilistic sample path and allows for the computation of the departure time, which is at most violated by a probability of ε, of a packet with index n. Hence, it describes the service of the system by an univariate function that holds for any interval of length $n \geq 0$ with respect to the violation probability. Furthermore, it makes the derivation of probabilistic performance bounds possible as, e.g., $\mathsf{P}[delay > x] \leq \varepsilon$. In contrast to that, $T_S(\nu, n)$ only allows the derivation of worst-case bounds for a specific sample path, which is non-practical for general results. The ε-effective service curve is defined as a probabilistic bound on the relation between the arrivals and the departures in Eq. (2.15)

$$\mathsf{P}\left[T_D(n) \leq \max_{\nu\in[0,n]}\{T_A(\nu) + T_S^{\varepsilon}(n-\nu)\}\right] \geq 1-\varepsilon.$$

We relate the service process $T_S(\nu, n)$ to the ε-effective service curve $T_S^{\varepsilon}(n)$ in Lem. 5.1. This relation is used in Thm. 5.1 to derive our system identification methodology for networks with service process $T_S(\nu, n)$ described by $T_S^{\varepsilon}(n)$.

Lemma 5.1 ([89]) *Given a system with bivariate service process $T_S(\nu, n)$ as in Eq.* (2.13)*. Any function $T_S^{\varepsilon}(n)$ that satisfies*

$$\mathsf{P}\left[T_S(\nu,n) \leq T_S^{\varepsilon}(n-\nu)\ , \forall \nu \leq n\right] \geq 1-\varepsilon \tag{5.1}$$

for $n \geq 0$ is an ε-effective service curve of the system in the sense of Eq. (2.15)*.*

Proof ([89]) Consider a sample path $T_S^{\omega}(\nu, n)$ of $T_S(\nu, n)$ and fix $n \geq 0$. If it holds that $T_S^{\omega}(\nu, n) \leq T_S^{\varepsilon}(n-\nu)$ for all $\nu \in [0, n]$, it follows from the monotonicity of the max-plus convolution for unit size packets that

$$T_D(n) = T_A \otimes T_S^{\omega}(n) \leq T_A \otimes T_S^{\varepsilon}(n).$$

Since, by assumption, the condition $T_S^{\omega}(\nu, n) \leq T_S^{\varepsilon}(n-\nu)$ holds for all $0 \leq \nu \leq n$ at least with probability $1-\varepsilon$, the claim follows. ∎

Example: We visualize the service process and the ε-effective service curve using a basic example of an On-Off server conducted with Matlab for $\varepsilon = 0.25$. In general, the violation probability is set to smaller values, here it is set to 0.25 for the clearness of the example. In detail, we consider a discrete system, at which the server forwards a packet with probability of $p = 0.1$ in each time slot based on an independent Bernoulli trial. Fig. 5.1 shows four sample paths, the sample path bound $T_S^{\varepsilon}(n-\nu)$, a lower bound, and an upper bound. For this example, the sample path bound is created empirically by generating one million sample paths and choosing $T_S^{\varepsilon}(n)$

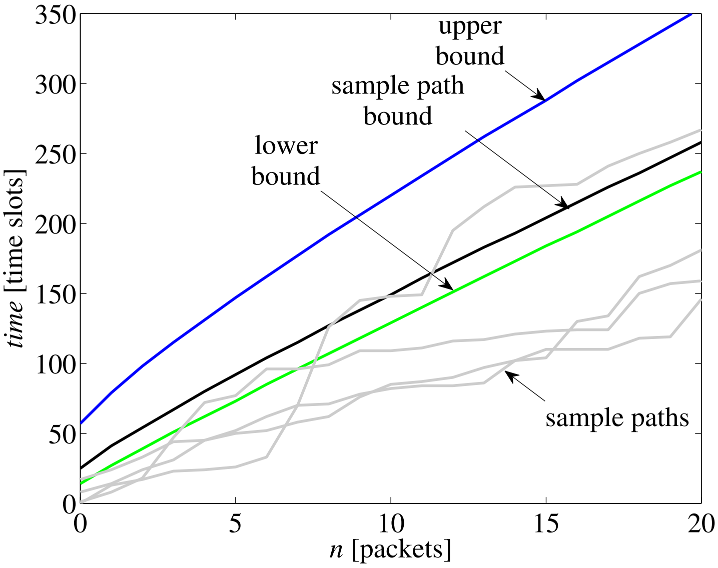


Figure 5.1: Example sample paths of $T_S(\nu, n)$ (gray lines), ε-effective service curve $T^{\varepsilon}_{\mathcal{S}}(n)$, point wise and sample path bound for an On-Off server with $\varepsilon = 0.25$. $T^{\varepsilon}_{\mathcal{S}}(n)$ is a sample path bound that is violated by one of four of the sample paths as defined by $\varepsilon = 0.25$. For almost all n the point wise violation probability is also $\varepsilon = 0.25$, but it is violated by two different sample paths, which is not sufficient for Eq. (5.1).

so that at most a fraction of 0.25 sample paths violates the bound. Additionally, the lower bound is derived from quantiles of the negative binomial distribution, since the number of time slots to forward one packet is geometrically distributed and the sum of a geometrically distributed random variable follows a negative binomial distribution. This lower bound is a point wise bound $T_{\mathcal{S}_p}(n)$ for which $\mathsf{P}\left[T_S(\nu, n) \leq T_{\mathcal{S}_p}(n-\nu)\right] \geq 1 - \varepsilon$ applies with $\varepsilon = 0.25$. The upper bound follows from the application of the union bound, which can be computed for a limited scale. If a limited scale of packets is considered, the violation probability is divided by the size of the scale and the computation of the upper bound follows in a similar manner as for the lower bound. Here, we set the scale to 101 packets.

The figure points out the meaning of $T^{\varepsilon}_{\mathcal{S}}(n)$ that only a fraction of 0.25 of the sample paths violates this bound, which is one of four in this case. The figure also highlights the difference to the lower bound. For almost each individual point n the violation probability is also ε for this point wise bound, but it is violated by two different sample paths. Such a bound is not sufficient to compute performance bounds and does not conform to Eq. (5.1). The sample path bound specifies the probability that a sample path crosses this bound, which is required to compute performance bounds.

5.2 ESTIMATION OF ε-EFFECTIVE MAX-PLUS SERVICE CURVES

For a linear system with service process $T_S(\nu, n)$, we prove that an inversion from delay measurements to an ε-effective service curve exists for constant rate packet train probes. The delay of a packet n is $W(n) = T_D(n) - T_A(n)$, with constant rate packet train probes $T_A(n) = \frac{n+1}{r}$ with rate r, we denote the delay as a function of the packet index and the rate $W(r, n) = T_D(n) - \frac{n+1}{r}$. If $W(r, n)$ has a steady state distribution for $n \to \infty$, we denote this distribution as $W(r)$ since it becomes independent of the packet index. Further, we define the $(1-\xi)$-quantile of the delay distribution by:

$$W^{\xi}(r, n) = \inf\left\{x \geq 0 : \mathsf{P}\left[W(r, n) \leq x\right] \geq 1 - \xi\right\}. \tag{5.2}$$

If a steady state delay distribution for $n \to \infty$ exists, we denote the quantile of it by $W^{\xi}(r)$. We prove in Lem. 5.2 when such a steady state delay distribution exists.

For constant rate packet train probes and the definition of quantiles of the delay distribution, we state the foundation for our estimation methodology in Thm. 5.1.

Theorem 5.1 ([89]) *Given a linear system with bivariate service process $T_S(\nu, n)$ as in Eq.* (2.7). *For a finite set R of probing rates r and $n \geq 0$*

$$T_S^{\varepsilon}(n) = \min_{r \in R}\left\{\frac{n}{r} + W^{\xi}(r)\right\} \tag{5.3}$$

is an ε-effective service curve of the system in the sense of Eq. (2.15) *with violation probability $\varepsilon = \sum_r \xi$.*

Proof ([89]) From the delay of the packet with index n, $W(n) = T_D(n) - T_A(n)$, and Eq. (2.13) it follows that

$$W(n) = \max_{\nu \in [0,n]}\{T_A(\nu) + T_S(\nu, n)\} - T_A(n) = \max_{\nu \in [0,n]}\{T_S(\nu, n) - T_A(\nu, n)\}. \tag{5.4}$$

The maximum in Eq. (5.4) implies that $W(n) \geq T_S(\nu, n) - T_A(\nu, n)$ for all $\nu \in [0, n]$, permitting us to write

$$T_S(\nu, n) \leq T_A(\nu, n) + W(n)\ , \forall \nu \geq 0.$$

Using constant rate arrivals with rate r, $T_A(\nu, n) = \frac{n-\nu}{r}$ and the delay quantile $W^{\xi}(r, n)$, we find that

$$\mathsf{P}\left[T_S(\nu, n) \leq \frac{n-\nu}{r} + W^{\xi}(r, n)\ , \forall \nu \geq 0\right] \geq 1 - \xi. \tag{5.5}$$

If the results are combined for probing rates $r \in R$ by using the complement of Eq. (5.5) and the union bound, it follows that

$$\mathsf{P}\left[\bigcup_{r \in R} T_S(\nu, n) > \frac{n-\nu}{r} + W^{\xi}(r, n)\ , \forall \nu \geq 0\right] \leq \sum_{r \in R} \xi.$$

Since we seek the smallest service curve that complies with Eq. (2.15), we apply the minimum and use the complement again

$$\mathsf{P}\left[T_S(\nu, n) \leq \min_{r \in R}\left\{\frac{n-\nu}{r} + W^{\xi}(r, n)\right\}, \forall \nu \geq 0\right] \geq 1 - \sum_r \xi.$$

With Lem. 5.1 we obtain that $T_{\mathcal{S}}^{\varepsilon}(n-\nu)$ defined as

$$T_{\mathcal{S}}^{\varepsilon}(n-\nu) = \min_{r \in R}\left\{\frac{n-\nu}{r} + W^{\xi}(r, n)\right\} \tag{5.6}$$

for all $\nu \in [0, n]$ is an ε-effective service curve with violation probability $\varepsilon = \sum_r \xi$. Letting $n \to \infty$, inserting the steady state delay $W^{\xi}(r)$, and substituting $n - \nu$ by n yields

$$T_{\mathcal{S}}^{\varepsilon}(n) = \min_{r \in R}\left\{\frac{n}{r} + W^{\xi}(r)\right\}$$

and completes the proof.

∎

Thm. 5.1 proves that an ε-effective service curve can be constructed from quantiles of the delay distributions obtained with constant rate packet train probes at linear systems.

Next, we show useful properties of Thm. 5.1 by taking advantage of properties of the Fenchel conjugates, which are introduced in Sec. 2.2.4. Our ε-effective service curve from Thm. 5.1 owns equivalent features with the transform $\underline{\mathfrak{F}}(s)$. We emphasize this by substituting $r = 1/p$, $W^{\xi}(r) = -V^{\xi}(s)$, and $P = \{1/r_1, 1/r_2, \dots\}$, permitting us to rewrite the result of Thm. 5.1 as

$$T^{\varepsilon}_{\mathcal{S}}(n) = \min_{p \in P}\{pn - V^{\xi}(p)\},$$

which is the concave Fenchel conjugate of $V^{\xi}(s)$. Because of the properties of the Fenchel conjugate, it is known that $T^{\varepsilon}_{\mathcal{S}}(n)$ of Eq. (5.3) is a concave function. Moreover, if $V^{\xi}(s)$ is concave, it holds that

$$V^{\xi}(s) = \min_{n \geq 0}\{sn - T^{\varepsilon}_{\mathcal{S}}(n)\}.$$

This establishes $W^{\xi}(r)$ as a dual characterization for systems, which is equivalent to the characterization by $T^{\varepsilon}_{\mathcal{S}}(n)$. Using our probing methodology is in accordance with applying the Fenchel transform twice: $\underline{\mathfrak{F}}\{\underline{\mathfrak{F}}_{T^{\varepsilon}_{\mathcal{S}}}(r)\}$. Thus, if the system features a concave service curve, the Fenchel transform is its own inverse $T^{\varepsilon}_{\mathcal{S}}(n) = \underline{\mathfrak{F}}\{\underline{\mathfrak{F}}_{T^{\varepsilon}_{\mathcal{S}}}(r)\}$. If the system has a non-concave service curve, we still obtain an useful bound since the Fenchel conjugate returns the concave hull, so that Eq. (5.3) conforms to the definition of the ε-effective service curve from Eq. (2.15).

To obtain an ε-effective service curve $T^{\varepsilon}_{\mathcal{S}}(n)$ by using Thm. 5.1, the network path is probed by constant rate packet trains with a length of N packets that are sent by rate r. The delay distribution of $W(r, n)$ from which the delay quantile is extracted, follows empirically from iterating packet trains I times with the same length and rate. This procedure is repeated for the set of probing rates R. We summarize the required parameters by the triple $\langle R, N, I\rangle$. In theory, the number of probing rates, the train length, and the number of iterations must tend to infinite to achieve a fine granular service curve with exact delay quantiles. In practice, these properties have to be constrained to finite values. We discuss the selection procedure in detail in Chap. 7.

Next, we give some general aspects of the parameters. Each probing rate $r \in R$ contributes a linear segment to the service curve with slope $1/r$ and y-axis intercept $W^{\xi}(r)$. The minimum of these segments gives the tightest bound in relation to the probing rates. The proof also shows that a tradeoff exists between the number of probing rates and the violation probability ε. More probing rates increase the resolution, but make the violation probability increase because of the union bound. The related delay quantile $W^{\xi}(r)$ follows from the empirical distribution by the use of a finite number of iterations and a limited train length, which observes the steady state distribution. We denote the estimate of the delay quantile by $\widetilde{W}^{\xi}(r)$ and refer to the service curve that results from a measurement with finite probing parameters as the ε-effective service curve estimate

$$\widetilde{T}_S^{\varepsilon}(n) = \min_{r \in R} \left\{ \frac{n}{r} + \widetilde{W}^{\xi}(r) \right\}. \tag{5.7}$$

Since the ε-effective service curve from Thm. 5.1 is based on the existence of a steady state delay distribution, we prove in Lem. 5.2 when a steady state delay distribution exists.

Lemma 5.2 ([89]) *Given arrivals $T_A(n)$ at a linear system with service process $T_S(\nu, n)$, satisfying Eq.* (2.13)*, and assuming joint stationarity*[8] *of $T_A(\nu, n)$ and $T_S(\nu, n)$*

1. *The delay is stochastically increasing in n.*
2. *If for all n it holds that*

$$\liminf_{m\to\infty} \frac{T_A(n-m,n)}{m+1} > \limsup_{m\to\infty} \frac{T_S(n-m,n)}{m+1}$$

   almost surely, the delay converges in distribution to a finite random variable W.

Note that we extend the processes $T_A(\nu, n)$ and $T_S(\nu, n)$ from $0 \leq \nu \leq n < \infty$ to $-\infty < \nu \leq n < \infty$. The proof closely follows [29, Lem. 9.1.4] as it is a generalization

---

[8] By stationarity we mean that $\mathsf{P}[T_A(\nu,n) \leq x] = \mathsf{P}[T_A(\nu+m,n+m) \leq x]$ and by joint stationarity $\mathsf{P}[T_A(\nu,n) \leq x, T_S(\nu,n) \leq y] = \mathsf{P}[T_A(\nu+m,n+m) \leq x, T_S(\nu+m,n+m) \leq y]$ for all $\nu \leq n$ and all m. If $T_A(\nu,n)$ and $T_S(\nu,n)$ are independent and stationary, also joint stationarity applies.

for shift-variant service. In [29, Lem. 9.1.4] the min-plus algebra is used for similar results on the backlog distribution at a constant rate server.

Proof ([89]) From Eq. (5.4) it follows for any x that

$$\begin{aligned}
&\mathsf{P}[W(n+1) \geq x] \\
&= \mathsf{P}\left[\max_{\nu\in[0,n+1]}\{T_S(\nu,n+1) - T_A(\nu,n+1)\} \geq x\right] \\
&\geq \mathsf{P}\left[\max_{\nu\in[1,n+1]}\{T_S(\nu,n+1) - T_A(\nu,n+1)\} \geq x\right] \\
&= \mathsf{P}\left[\max_{\nu\in[0,n]}\{T_S(\nu+1,n+1) - T_A(\nu+1,n+1)\} \geq x\right].
\end{aligned}$$

From the assumption of joint stationarity, it follows that the joint distribution of $T_A(\nu+1,n+1)$ and $T_S(\nu+1,n+1)$ is equal in distribution to the joint distribution of $T_A(\nu,n)$ and $T_S(\nu,n)$ for all ν and n. The last line thereby equals $\mathsf{P}[W(n) \geq x]$. Hence, $\mathsf{P}[W(n+1) \geq x] \geq \mathsf{P}[W(n) \geq x]$, which proves that the delay $W(n)$ is stochastically increasing.

From the second assumption of Lem. 5.2, it follows that for any n there exists a finite random variable

$$\mathcal{N} = \max\{m \geq 0 : T_A(n-m,n) \leq T_S(n-m,n)\}.$$

Thus, $T_S(n-m,n) < T_A(n-m,n)$ holds for all $m > \mathcal{N}$. Moreover, since $T_S(n-m,n)$ increases with $m \geq 0$, we have $T_S(n-m,n) \leq T_S(n-\mathcal{N},n)$ for all $0 \leq m \leq \mathcal{N}$. Connecting the two statements and using that $T_A(n-m,n)$ for $m \geq 0$ and $T_S(n-\mathcal{N},n)$ are non-negative yields

$$T_S(n-m,n) - T_A(n-m,n) \leq T_S(n-\mathcal{N},n)$$

for all $m \geq 0$ and hence

$$\max_{m\geq 0}\{T_S(n-m,n) - T_A(n-m,n)\} \leq T_S(n-\mathcal{N},n).$$

For packet indices $-\infty < n - m \leq n < \infty$ the distribution of the delay

$$W(n) = \max_{m \geq 0}\{T_S(n-m,n) - T_A(n-m,n)\}$$

is bounded by

$$\sup_n\{\mathsf{P}[W(n) \geq x]\} \leq \sup_n\{\mathsf{P}[T_S(n-\mathcal{N},n) \geq x]\}.$$

Since $\mathcal{N}$ is finite and $W(n)$ is stochastically increasing in n, there exists a finite random variable W such that

$$\lim_{n\to\infty} \mathsf{P}[W(n) \geq x] = \sup_n\{\mathsf{P}[W(n) \geq x]\} = \mathsf{P}[W \geq x],$$

which completes the proof of the second statement. ∎

Our ε-effective service curve holds for an unlimited scale of $n \to \infty$ packets if a steady state delay distribution exists. The previous lemma proves the existence of the steady state delay distribution as long as the probing rate does not exceed the long-term rate of the service process $\limsup_{n\to\infty} \frac{T_S(\nu,n)}{n-\nu+1}$.

However, already packet trains that do not observe steady state delays deliver service curves that apply to a limited scale of their train length. For systems with stationary service, i.e., $\mathsf{P}\left[T_S(\nu,n) \leq x\right] = \mathsf{P}\left[T_S(\nu+m,n+m) \leq x\right]$ an invariant delay distribution exists for a train length n:

$$\begin{aligned}
&\mathsf{P}\left[W(r,n) \leq x\right] \\
&= \mathsf{P}\left[\max_{\nu\in[0,n]}\left\{T_S(\nu,n) - \frac{n-\nu}{r}\right\} \leq x\right] \\
&= \mathsf{P}\left[\max_{\nu\in[0,n]}\left\{T_S(\nu+m,n+m) - \frac{n-\nu}{r}\right\} \leq x\right].
\end{aligned} \tag{5.8}$$

From Lem. 5.2 it follows that the delay is stochastically increasing, therefore the delay quantile of Eq. (5.6) is replaced by $W^{\xi}(r, N-1)$. The resulting service curve applies for the scale $[0, N-1]$. Such limited scales were introduced in the min-plus algebra in [25] as time scale bounds.

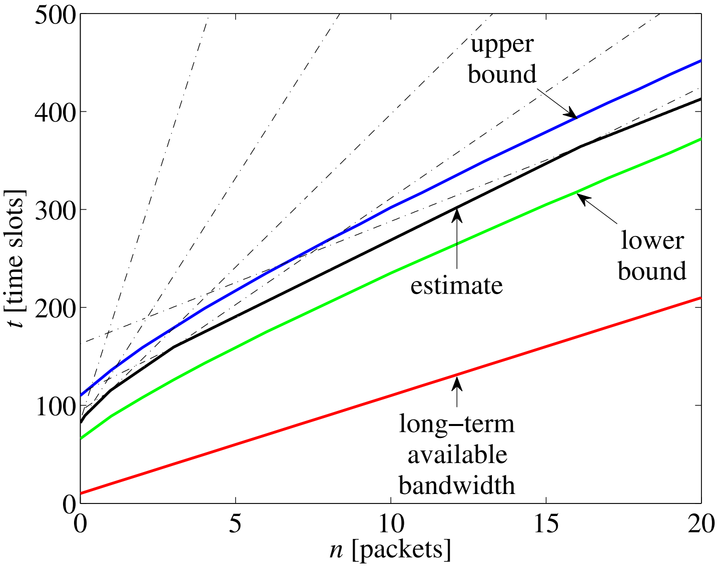


Figure 5.2: The ε-effective service curve estimate compared to analytical upper and lower bounds and the long-term available bandwidth. The ε-effective service curve is composed of the linear segments (presented as dash-dotted lines). The slope of each segment follows from the reciprocal probing rate $1/r$ and the y-axis intercept from the delay quantile $\widetilde{W}^{\xi}(r,n)$. The estimate is between the upper and lower bounds and its slope converges in the long-term to the long-term available bandwidth. In the short-term, the ε-effective service curve captures effects on short scales which are not contained in the long-term available bandwidth.

Example: We illustrate the system identification methodology given in Thm. 5.1 using the random On-Off server from Sec. 5.1. As given in the example in Sec. 5.1, the server forwards a packet in a time slot with probability of $p = 0.1$ and pauses with a probability of $1 - p = 0.9$, which implies a non-work-conserving system. To estimate the ε-effective service curve, we use packet trains of length 101 packets and set $\varepsilon = 10^{-3}$.

Fig. 5.2 shows the ε-effective service curve estimate for the scale of 21 packets. The estimate given in Thm. 5.1 is composed of the minimum of the linear segments, one segment per probing rate. These segments are added to the Fig. 5.2 as dash-dotted lines to visualize the construction of the service curve. The slope of each segment is the reciprocal probing rate $1/r$ and the axis intercept is given by the estimated delay quantile $\widetilde{W}^{\xi}(r, N-1)$. The resolution depends on the number of probing rates. Increasing the number of probing rates improves the resolution, but it increases at the same time the violation probability. The computation of the lower and upper bound is explained in the example of Sec. 5.1. This example shows the difference between the service curve and the long-term available bandwidth.

An ε-effective service curve includes information on different scales related to a violation probability, whereas the long-term available bandwidth is an average and so ignores the availability of the service on different scales. For example, to forward one packet with a probability of $1-\varepsilon$, it is expected to need about 75 time slots. In the long-term, the rate of the service curve estimate, which is indicated by the slope of the curve, approaches the long-term rate of 10 time slots per packet. The analytical upper and lower bounds confirm the estimate since it is in between. Furthermore, this example demonstrates the applicability to non-work-conserving systems due to the characteristic of the considered system.

5.3 SYSTEM IDENTIFICATION IN THE MIN-PLUS ALGEBRA

The min-plus algebra is prevalent in the framework of the network calculus. This is predicated by the simple derivation of results for multiplexing traffic since it follows by addition. Therefore, a representation of the system identification methodology in the min-plus algebra of the estimate is desirable.

In the following, we first show the derivation of the system identification methodology in the min-plus algebra using a fluid traffic model. The derivation resembles the derivation in the max-plus algebra presented before in Sec. 5.1 and Sec. 5.2, but estimates are based on backlog measurements instead of delay measurements. Due to the repetitive structure of the derivation, we omit the repetitive proofs in this section and present them in the appendix. For the detailed derivations, we refer to Sec. 5.2. In these sections, we first rely on a fluid traffic model. Later in Sec. 5.3.2, we prove that the methodology is applicable for networks with packetized traffic, too.

To combine the advantageous properties of the system identification procedures in the two algebras, we show a pseudo-inversion from the max-plus to the min-plus algebra in Sec. 5.3.3. The inversion transforms the ε-effective service curve $T^{\varepsilon}_{S}(n)$ that is defined in Thm. 5.1 in the max-plus algebra to the corresponding service curve $\mathcal{S}^{\varepsilon}(t)$ defined in Thm. 5.2 in the min-plus algebra. To the best of our knowledge, no inversion for the ε-effective service curves from the max-plus to the min-plus algebra exists, for details see Chap. 2. Offering such an inversion for the service curve combines the practicability of the methodology in the max-plus algebra and the broader applicability in the min-plus algebra.

5.3.1 *Systems with Fluid Traffic*

The service curves in the min-plus algebra are based on a fluid traffic model. To state the identification methodology, we connect the service process $S(\tau,t)$ and the ε-effective service curve $\mathcal{S}^{\varepsilon}(t)$. From Eq. (2.10) in Sec. 2.3.1.1, the time-varying service is specified by the bivariate function of the service process $S(\tau,t)$. For linear networks it is defined by

$$D(t) = \inf_{\tau\in[0,t]} \{A(\tau) + S(\tau,t)\} = A \otimes S(t).$$

Supplementary, the random service is also expressed by an ε-effective service curve, which specifies the service by an invariant probabilistic bound given in Eq. (2.14) in Sec. 2.3.2.1. A service curve $\mathcal{S}^{\varepsilon}(t)$ is an ε-effective service curve if

$$\mathsf{P}\left[D(t) \geq \inf_{\tau\in[0,t]} \{A(\tau) + \mathcal{S}^{\varepsilon}(t-\tau)\}\right] \geq 1-\varepsilon$$

holds, where ε is the violation probability.

Lem. 5.3 relates the ε-effective service curve to the service process by specifying the ε-effective service curve $\mathcal{S}^{\varepsilon}(t)$ as a probabilistic time-invariant bound on the service process $S(\tau,t)$.

Lemma 5.3 ([90]) *Given a linear system with service process $S(\tau,t)$ as in Eq.* (2.10). *Any function $\mathcal{S}^{\varepsilon}(t)$ that satisfies the sample path bound*

$$\mathsf{P}\left[S(\tau,t) \geq \mathcal{S}^{\varepsilon}(t-\tau)\ , \forall\tau \geq 0\right] \geq 1-\varepsilon$$

for $t \geq 0$ is an ε-effective service curve of the system in the sense of Eq. (2.14).

The proof is omitted here and is presented in the Appendix A.2.

The system identification in the min-plus algebra relies on backlog measurements obtained with constant rate probe traffic. The backlog, which is measured with constant rate r and resulting arrivals $A(t) = rt$ at time t, follows from the difference of the arrivals and the departures $B(r,t) = rt - D(t)$. With random service, $B(r,t)$ becomes a random variable. The steady state of $B(r,t)$ for $t \to \infty$, where the backlog distribution becomes independent of time, is denoted as $B(r)$. For the derivation of the ε-effective service curve, we require quantiles of the backlog distribution, which we define by

$$B^{\xi}(r,t) = \inf \{x \geq 0 : \mathsf{P}\left[B(r,t) \leq x\right] \geq 1 - \xi\}\,, \tag{5.9}$$

where we use ξ to denote the violation probability of the quantile. With constant rate probes and the definition of the quantile, the system identification methodology is presented in Thm. 5.2 and is proven in the Appendix A.2.

Theorem 5.2 ([90]) *Given a linear system with service process $S(\tau,t)$ as in Eq.* (2.10)*. Select a finite set R of rates $r \geq 0$. For all $t \geq 0$, the function*

$$\mathcal{S}^{\varepsilon}(t) = \max_{r \in R}\{rt - B^{\xi}(r)\}$$

is an ε-effective service curve of the system in the sense of Eq. (2.14) *with violation probability $\varepsilon = \sum_{r \in R} \xi$.*

Thm. 5.2 develops a methodology to obtain an ε-effective service curve from quantiles of the backlog distribution $B^{\xi}(r)$. The quantiles are extracted from empirical backlog distributions measured with constant rate packet trains. A measurement setup for an estimate is defined by the triple $\langle R, N, I \rangle$, i.e., for a set of rates R, packet trains with length N, each iterated I times. Each iteration for rate $r \in R$ provides a sample for the empirical backlog distribution. We denote the resulting estimate of the ε-effective service curve by $\widetilde{\mathcal{S}}^{\varepsilon}(t)$. For detailed properties and practical implications of Thm. 5.2, we refer to Thm. 5.1 since the properties and implications apply in a similar way.

Thm. 5.2 requires steady state backlog distributions. Lem. 5.4 states that these exist, if the probing rate r is below the long-term rate of the service process defined as $\liminf_{t\to\infty} \frac{S(\tau,t)}{(t-\tau)}$.

Lemma 5.4 ([90]) *Given arrivals* $A(\tau,t)$ *at a linear system with service process* $S(\tau,t)$*, satisfying Eq.* (2.10)*, and assuming joint stationarity of* $A(\tau,t)$ *and* $S(\tau,t)$*.*

1. *The backlog* $B(t)$ *is stochastically increasing in* t*.*
2. *If for all* t *it holds that*

$$\limsup_{\delta\to\infty} \frac{A(t-\delta,t)}{\delta} < \liminf_{\delta\to\infty} \frac{S(t-\delta,t)}{\delta}$$

almost surely, the backlog converges in distribution to a finite random variable B*.*

Note that we extend the processes $A(\tau,t)$ and $S(\tau,t)$ from $0 \le \tau \le t < \infty$ to $-\infty < \tau \le t < \infty$. The lemma generalizes Lem. 9.1.4 presented in [29] from a constant rate server to a server with random service. The proof is stated in Appendix A.2.

Thm. 5.2 relies on fluid traffic, which deviates from the estimation procedure with packetized traffic stated in the max-plus algebra. In Sec. 5.3.2, we prove that the estimation methodology also holds for packetized traffic in the min-plus algebra.

5.3.2 *Systems with Packetized Traffic*

Traffic in computer networks is packetized and not fluid, we show here that Thm. 5.2 also applies to packetized traffic. In Section 5.3.1, we assume fluid traffic for our system identification methodology to reduce the complexity of the derivation.

Fluid traffic is transformed to packetized traffic by the application of a packetizer, see Sec. 2.2.3. For unit sized packets, as assumed here, packetized traffic results from rounding down the continuous functions, e.g., arrivals are packetized if $A(t) = \lfloor A(t) \rfloor$ applies.

Next, we prove that the packetized ε-effective service curve $\lfloor \mathcal{S}^{\varepsilon}(t) \rfloor$ is an ε-effective service curve of a system with packetized arrivals and departures. Thereafter, we derive the system identification methodology, given in Thm. 5.2, for packetized traffic.

Lemma 5.5 *Given a system with ε-effective service curve as in Eq.* (2.14) *and packetized arrivals $A(t) = \lfloor A(t) \rfloor$. It holds that*

$$\mathsf{P}\big[\lfloor D(t) \rfloor \geq \lfloor A(t) \rfloor \otimes \lfloor \mathcal{S}^{\varepsilon}(t) \rfloor\big] > 1 - \varepsilon.$$

Proof From Eq. (2.14) we obtain that

$$\mathsf{P}\big[D(t) \geq A \otimes \mathcal{S}^{\varepsilon}(t)\big] \leq \mathsf{P}\big[\lfloor D(t) \rfloor \geq \lfloor A \otimes \mathcal{S}^{\varepsilon}(t) \rfloor\big], \tag{5.10}$$

since, generally, if $D(t) \geq A \otimes \mathcal{S}^{\varepsilon}(t)$, it holds that $\lfloor D(t) \rfloor \geq \lfloor A \otimes \mathcal{S}^{\varepsilon}(t) \rfloor$. Since the arrivals are assumed to be packetized, we can substitute $A(t) = \lfloor A(t) \rfloor$ so that

$$\lfloor A \otimes \mathcal{S}^{\varepsilon} \rfloor = \lfloor \lfloor A \rfloor \otimes \mathcal{S}^{\varepsilon} \rfloor = \lfloor A \rfloor \otimes \lfloor \mathcal{S}^{\varepsilon} \rfloor. \tag{5.11}$$

By inserting Eq. (5.11) into Eq. (5.10), it follows that $\lfloor \mathcal{S}^{\varepsilon}(t) \rfloor$ is an ε-effective service curve for the packetized departures $\lfloor D(t) \rfloor$. ∎

With Lem. 5.5 we restate Thm. 5.2 for packetized systems in Thm. 5.3. The proof is deferred to Appendix A.2 .

Theorem 5.3 *Given a packetized linear system with service process $S(\tau, t)$ as in Eq.* (2.10). *Select a finite set R of probing rates $r \geq 0$. For all $t \geq 0$, the function*

$$\mathcal{S}^{\varepsilon}(t) = \max_{r \in R}\{\lfloor rt \rfloor - B^{\xi}(r)\}$$

is an ε-effective service curve of the system in the sense of Eq. (2.14) *with violation probability $\varepsilon = \sum_{r \in R} \xi$.*

We illustrate the application of Thm. 5.3 by repeating the example of the On-Off server from Sec. 5.2.

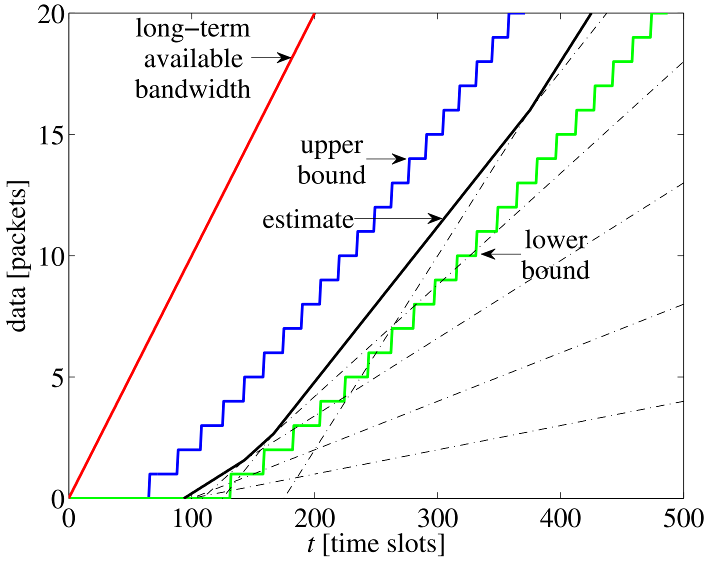


Figure 5.3: The ε-effective service curve estimate, derived in the min-plus algebra, compared to the analytical upper and lower bounds for packetized traffic and the long-term available bandwidth for fluid traffic. The ε-effective service curve is composed of linear segments (presented as dash-dotted lines). The slope of each segment follows from the probing rate r and the y-axis intercept from the negative backlog quantile $-\widetilde{B}^{\xi}(r,n)$. The estimate is between the upper and lower bounds and its slope converges to the long-term available bandwidth. In the short-term, the ε-effective service curve captures effects on short scales. Obviously, delay information is contained, indicated by the right-shift of the curve, which is neglected by the long-term available bandwidth.

Example: We simulate an On-Off server with Matlab, which forwards packets in each time slot with a probability of $p = 0.1$. Therefore, the system is a discrete random non-work-conserving system. The train duration of the packet train probes is 1000 time slots[9] and the violation probability $\varepsilon = 10^{-3}$.

Fig. 5.3 presents the ε-effective service curve $\widetilde{\mathcal{S}}^{\varepsilon}(t)$ together with the linear segments contributed by each probing rate $r \in R$, indicated by dash-dotted lines. The linear segments are linear equations with slope r and y-axis intercept $-B^{\xi}(r)$. Furthermore, the figure shows upper and lower bounds and the long-term available bandwidth. The upper bound follows from the number of packets forwarded in a given number of time slots with probability $1 - \varepsilon$ at least, which corresponds to a binomial distribution. The lower bound results also from the binomial distribution with the application of the union bound with respect to the number of time slots. The long-term available bandwidth follows directly from the probability $p = 0.1$, which results in a rate of 0.1 packets per time slot.

---

[9] This duration transfers to a scale of 100 packets using the mean rate of 0.1, this is about the scale used in previous example in the max-plus algebra.

Fig. 5.3 demonstrates the difference between the service curve estimate and the long-term available bandwidth of $\alpha^{\infty} = 0.1$ packets per time slot. The long-term available bandwidth neglects that on short time scales less service is available with respect to the violation probability ε and so it overestimates the available service. In contrast, the service curve accounts for initial delays and service availability on various time scales as reflected by the shift to the right of the origin of the curve and the increasing slope of the curve. The slope of the service curve converges to the slope of the long-term available bandwidth in the long-term. In addition, the upper and lower bounds validate the estimate since it is in between the bounds.

The figure also illustrates the differences and similarities to the example in the max-plus algebra, if compared to Fig. 5.2. The service curve plots amount of data (in packets) against time, whereas the representation in the max-plus algebra is reversed. Furthermore, in the max-plus algebra we plot the packet index instead of the amount of data. The lower bound and the upper bound from Fig. 5.3 refer to the upper bound and the lower bound in Fig. 5.2, respectively. By comparison of the curves in the min-plus and max-plus algebra, the better intuition of the min-plus algebra is also explainable since a higher value implies a higher throughput. In the max-plus algebra it is counterintuitive. A lower value indicates a higher throughput since the time to process data is reduced.

5.3.3 *Connection between the System Identification in the Max-plus Algebra and in the Min-plus Algebra*

We use the max-plus algebra to derive our system identification methodology in Sec. 5.2 and the min-plus algebra in Sec. 5.3.1 and in Sec. 5.3.2. The first approach in the max-plus algebra is advantageous for practical measurements since it directly exploits timestamps contained in probe packets for delay calculations, and it can represent lost packets by an infinite delay. For the second approach in the min-plus algebra, a comprehensive framework is available to build on. In Sec. 5.3.1, we deduce the system identification methodology from a fluid traffic model. This assumption is relaxed in Sec. 5.3.2, where we show that the min-plus approach also holds for packetized traffic. Next, we deduce a pseudo-inverse of an ε-effective service curve in the max-plus algebra to an ε-effective service curve in the min-plus. In this way, we enable the estimation in the max-plus algebra and continue the processing in the min-plus algebra.

We define the pseudo-inverse of the ε-effective service curve $T_{\mathcal{S}}^{\varepsilon}(n)$ by applying the pseudo inverse used for the inversion of the arrivals and the departures in Eq. (2.8)

$$(T_{\mathcal{S}}^{\varepsilon})^{-1}(t) = \sum_{n \geq 0} 1_{\{T_{\mathcal{S}}^{\varepsilon}(n) \leq t\}}. \tag{5.12}$$

The connection between the system identification in the max-plus algebra to the min-plus algebra is stated in the following theorem by the use of the pseudo-inverse given in Eq. (5.12).

Theorem 5.4 ([90]) *In a discrete system, given ε-effective service curves $\mathcal{S}^{\varepsilon}(t)$ from Thm. 5.3 and $T_{\mathcal{S}}^{\varepsilon}(n)$ from Thm. 5.1, and assuming that the system forwards traffic in order of its arrival, the following holds:*

$$(T_{\mathcal{S}}^{\varepsilon})^{-1}(t) - 1 \leq \mathcal{S}^{\varepsilon}(t) \leq (T_{\mathcal{S}}^{\varepsilon})^{-1}(t).$$

Thm. 5.4 states that the inverse of the service curve obtained by the identification in the max-plus algebra deviates at most by one packet to the one in the min-plus algebra. For the proof of the theorem, we first require a relation between the backlog and the delay, which provides the basis for the estimates from Thm. 5.3 and Thm. 5.1. This relation is stated in the next lemma.

Lemma 5.6 ([90]) *Given a system with strictly increasing arrival and departure timestamps $T_A(n)$ and $T_D(n)$, respectively. Assume the system serves arrivals in order. The backlog observed at the departure time $T_D(n)$ of packet n equals*

$$\begin{aligned} B(T_D(n)) &= A(T_D(n) - W(n), T_D(n)) \\ &= A(T_A(n), T_A(n) + W(n)), \end{aligned}$$

where $A(\tau, t)$ are the cumulative arrivals in $(\tau, t]$ and $W(n)$ is the delay of packet n.

Proof ([90]) From the definition of backlog we have

$$B(T_D(n)) = A(T_D(n)) - D(T_D(n)).$$

Since the arrivals are served in order and $T_A(n)$ and $T_D(n)$ are strictly increasing it holds that $D(T_D(n)) = A(T_A(n))$ and it follows by substitution that

$$B(T_D(n)) = A(T_A(n), T_D(n)).$$

Using the definition of the delay $W(n) = T_D(n) - T_A(n)$ completes the proof. ∎

Note additionally to the conventions from Sec. 2.1, we require that the arrival function $T_A(n)$ and departure function $T_D(n)$ are strictly increasing, i.e., only one packet arrives to or departs from the network at the same time. In [88], we show that Little's law can be proven in the framework of the network calculus with Lem. 5.6. Here, we use the relation between the backlog $B(t)$ at the time of packet departure $t = T_D(n)$ to prove Thm. 5.4.

Proof (of Thm. 5.4 [90]) Consider the argument of the indicator function $T_S^{\varepsilon}(n) \leq t$ in the definition of the pseudo-inverse from Eq. (5.12). By insertion of $T_S^{\varepsilon}(n)$ from Thm. 5.1, we have $\min_{r \in R}\{n/r + W^{\xi}(r)\} \leq t$ and it follows that

$$\begin{aligned}
&\min_{r \in R}\left\{\frac{n}{r} + W^{\xi}(r) - t\right\} \leq 0\\
\Leftrightarrow\, &\min_{r \in R}\{n + rW^{\xi}(r) - rt\} \leq 0\\
\Leftrightarrow\, &\max_{r \in R}\{rt - rW^{\xi}(r)\} \geq n.
\end{aligned}$$

Instantiating Lem. 5.6 with $A(t) = \lfloor rt \rfloor$ yields the backlog $B(T_D(n)) = \lfloor rW(r,n) \rfloor$. Letting $n \to \infty$ and taking quantiles, we obtain $B^{\xi}(r) = \lfloor rW^{\xi}(r) \rfloor = rW^{\xi}(r) - \vartheta(r)$, where $\vartheta(r) \in [0, 1)$. It follows that the condition $T_S^{\varepsilon}(n) \leq t$ is equivalent to

$$n \leq \max_{r \in R}\{rt - B^{\xi}(r) - \vartheta(r)\}.$$

With Thm. 5.3 and since $n = 0, 1, 2, \ldots$ is an integer, the claim follows. ∎

Thm. 5.4 allows us to transfer service curves obtained by the system identification methodology in max-plus algebra to the min-plus algebra. This inversion connects the advantages of both algebras: the identification in the max-plus algebra operates directly on timestamps included in probe packets, whereas the min-plus algebra features a comprehensive framework in the network calculus. The identification requires the selection of parameters for practical probing procedures, which are the set of probing rates R, the train length N, and the number of iterations I per train. Chap. 7 elaborates the transformation from the system identification methodology to practical probing procedures. In this chapter, we also deduce the system identification procedures directly in min-plus algebra for a fluid as well as a packetized system. In the next chapter, we make use of the system model defined in the min-plus algebra to state connections between the definitions of the available bandwidth and service curve concepts known in the network calculus.

6

CONNECTION BETWEEN SYSTEM MODELS IN THE NETWORK CALCULUS AND THE AVAILABLE BANDWIDTH

The available bandwidth and the service curves from the network calculus are both used to characterize the service of a network path available to a flow. We explicitly state the connection between these approaches and show that the available bandwidth can be expressed in the network calculus by the leftover service process introduced for deterministic systems in [29, Sec. 2.3.3] and for time-varying systems in [42].

We deduce the connection for single-hop networks as well as for multi-hop networks. In detail, the relation between the definition of the available bandwidth, given in Eq. (3.1), and the leftover service process for single-hop networks is derived in Sec. 6.1. This supports the reflection of the available bandwidth in the network calculus. For example, we show that using the expected value of the available bandwidth, as often implemented by available bandwidth estimation tools, leads to a systematic overestimation of the expected departures.

In Sec. 6.2, we present the relation for multi-hop networks and illustrate that the definition of the available bandwidth for multi-hop networks, see Eq. (3.2), only coincides with the service process in the long-term. By using the framework of the network calculus, we explain effects observed in available bandwidth estimation for multiple bottleneck links, e.g., as pointed out in [60]. We show that service curves are able to capture effects present in the short-term behavior of systems with random service, whereas available bandwidth estimation neglects such effects.

The presented results are based on the joint work with Markus Fidler and Jörg Liebeherr and are partially available in [90].

6.1 SINGLE-HOP NETWORKS

In this section, we deduce the connection between service curves and the available bandwidth for single-hop networks. In detail, we establish the relation between the leftover service process, which is established for time-varying systems in [42], to Eq. (3.1). The leftover service process specifies the unused capacity of a system by a service curve. Also the definition of the available bandwidth indicates the fraction of unused capacity. In Eq. (3.1), the unused capacity completely contributes to the available bandwidth, which implies a work-conserving system. The relation relies on multiplexing of cross traffic arrivals $A_c(t)$ and through traffic arrivals $A_t(t)$. The cross traffic describes the existing traffic, whereas the available bandwidth refers to the bandwidth available to the through flow. Since the relation is based on multiplexing traffic, we assume a linear system with fluid traffic in the min-plus algebra from Sec. 2.3.1.1, where multiplexing follows by addition. The assumption of linearity implies that the through flow uses at most the available bandwidth and does not displace the cross traffic. The corresponding departures of the arrivals are $D_c(t)$ and $D_t(t)$. The following lemma presents the relation.

Lemma 6.1 ([90]) *For a work-conserving linear fluid system with constant rate service process $S(\tau, t) = C(t - \tau)$ and time-varying utilization $u(t)$, where $u(t) \in \{0, 1\}$, it holds that*

$$\alpha(\tau, t) = \frac{\int_\tau^t C(1 - u(x))dx}{t - \tau} = \frac{S_l(\tau, t)}{t - \tau}, \tag{6.1}$$

where $\alpha(\tau, t)$ is the available bandwidth, given in Eq. (3.1), and $S_l(\tau, t)$ is the leftover service process, which we define as $S_l(\tau, t) = S(\tau, t) - D_c(\tau, t)$, where $D_c(\tau, t)$ are the departures of the cross-traffic.

Proof ([90]) For τ determining the infimum of Eq. (2.10) (i.e. the beginning of the last busy period before t) and using the assumption of a work-conserving system, it holds that $D(t) = A(\tau) + S(\tau, t)$ and $D(\tau) = D(t) - S(\tau, t)$. From this it follows that $A(\tau) = D(\tau) = D_c(\tau) + D_t(\tau) = A_c(\tau) + A_t(\tau)$ and $D_t(\tau) = A_t(\tau)$. Writing the departures as $D(t) = D_c(t) + D_t(t)$ yields

$$D_t(t) = A_t(\tau) + \underbrace{S(\tau,t) - D_c(\tau,t)}_{=:S_l(\tau,t)},$$

whereof the leftover service process $S_l(\tau, t)$ follows.

For τ not determining the infimum of Eq. (2.10), there exists one $\tau \in [0, t]$ at least for which $D_t(t) \geq A_t(\tau) + S_l(\tau, t)$ holds. Additionally, for all $\tau \in [0, t]$ by causality $D_t(\tau) \leq A_t(\tau)$, which yields $D_t(t) \leq A_t(\tau) + S_l(\tau, t)$. Combining the results, $D_t(t) = A_t \otimes S_l(\tau, t)$ holds.

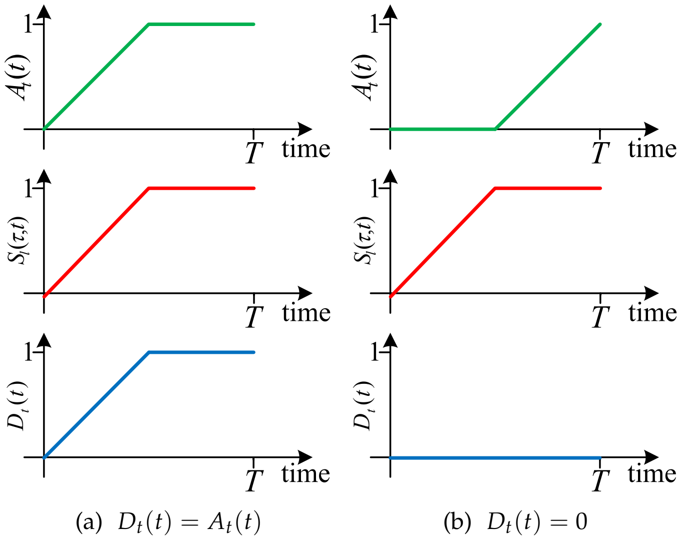
Given the service process $S(\tau, t) = C(t - \tau)$, cross-traffic departures following from the utilization $D_c(\tau, t) = C \int_\tau^t u(x)dx$ in $(\tau, t]$, and expressing the service process as a rate by $\frac{S_l(\tau,t)}{t-\tau}$ completes the proof with

$$\alpha(\tau, t) = \frac{\int_\tau^t C(1 - u(x))dx}{t - \tau} = \frac{S(\tau, t) - D_c(\tau, t)}{t - \tau}.$$

∎

Although we prove the equality between the available bandwidth and the leftover service in Lem. 6.1, a difference consists in how the result of the available bandwidth estimate is usually expressed. Often available bandwidth estimation tools express the estimate by a single value. Using only a single value discards information on the time scale belonging to the estimate. The assumption of a constant available bandwidth neglects effects, which occur due to a time-varying available bandwidth as inherent in real networks. These effects, which we substantiate below, result in less departures as indicated by the available bandwidth. Using a service process $S_l(\tau, t)$ or an estimate of it, as given by the ε-effective service curve from Thm. 5.3, includes the timing from $(0, t]$ and recovers the departures by the convolution: $D(t) = A \otimes S_l(t)$.

The before mentioned behavior is illustrated in Fig. 6.1. From top to bottom, the figure presents the arrivals $A_t(t)$ to a system with a time-varying service $S_l(\tau, t)$ resulting in the departures $D_t(t)$ in the interval $(0, T]$. The figure shows two examples, in Fig. 6.1a the timing is a fortunate coincidence since all arrivals



(a) $D_t(t) = A_t(t)$ (b) $D_t(t) = 0$

Figure 6.1: Example of the impact of timings between service and arrivals. In Fig. (a) the timing of the arrivals and the service is beneficial, since all arrivals are forwarded by the system. On the contrary, in Fig. (b) when data arrives no service is available in $(T/2, T]$. Ignoring the timing of arrivals and the service returns a misleading result, since for both cases the available bandwidth and the arrivals at time T are equal.

are forwarded by the system. On the contrary, in Fig. 6.1b when data arrive in the interval $(T/2, T]$ no service is available in this interval. This example shows that assuming the available bandwidth α to be constant over time is misleading, since departures depend on the timing of the arrivals and the service during the interval $(0, t]$.

Next, we analyze an effect that arises from the typical assumption of a system with fluid flow constant rate traffic as elaborated in Sec. 3.1.1. The system model used in available bandwidth estimation typically neglects variability of the service [60] and available bandwidth estimation tools often calculate the mean of repeated estimates as described in Sec. 3.1.1. Using the relation from Lem. 6.1 between the available bandwidth and the leftover service process, we are able to explain effects of the assumed model to available bandwidth estimates, which are conducted in networks with random service. In the following, we state in Lem. 6.2 that using the expected value of $\mathsf{E}[\alpha(\tau, t)]$ leads to a systematic overestimation of the departures.

Lemma 6.2 ([90]) *Given $D_t(t) = A_t \otimes S_l(t)$. It holds that*

$$\mathsf{E}[D_t(t)] \leq A_t \otimes \mathsf{E}[S_l](t).$$

Proof ([90]) Taking expectations we have

$$\begin{aligned}\mathsf{E}[D_t(t)] &= \mathsf{E}\left[\inf_{\tau\in[0,t]}\{A_t(\tau)+S_l(\tau,t)\}\right]\\ &= \sum_{S_l^\omega\in S_l^\Omega} p^\omega \inf_{\tau\in[0,t]}\{A_t(\tau)+S_l^\omega(\tau,t)\},\end{aligned}$$

where S_l^Ω is the sample space containing sample paths S_l^ω that occur with probability p^ω each. For any choice of $\tau'\in[0,t]$,

$$p^\omega \inf_{\tau\in[0,t]}\{A_t(\tau)+S_l^\omega(\tau,t)\} \le p^\omega A_t(\tau') + p^\omega S_l^\omega(\tau',t),$$

which yields for any $\tau'\in[0,t]$ that

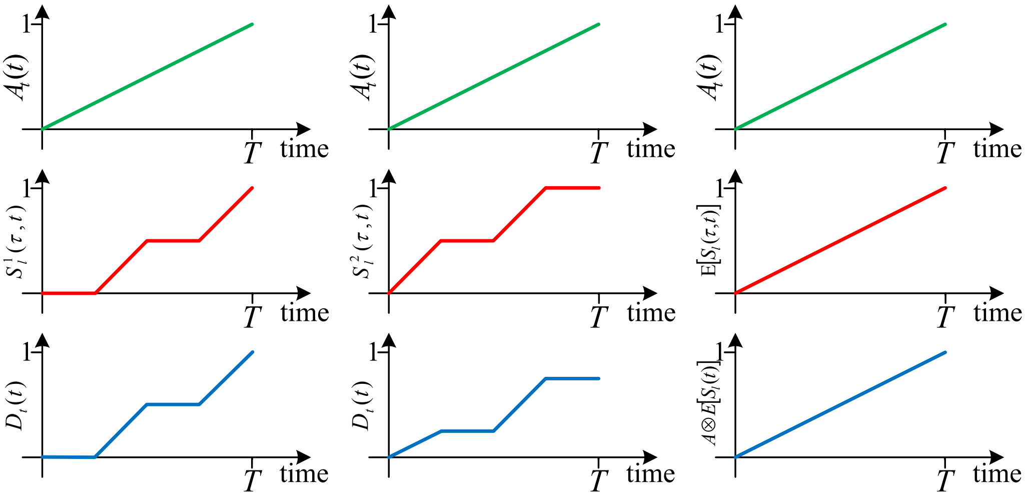
$$\mathsf{E}[D_t(t)] \le A_t(\tau') + \sum_{S_l^\omega\in S_l^\Omega} p^\omega S_l^\omega(\tau',t),$$

since $\sum_{S_l^\omega\in S_l^\Omega} p^\omega = 1$. It follows that

$$\mathsf{E}[D_t(t)] \le \inf_{\tau'\in[0,t]}\{A_t(\tau') + \mathsf{E}[S_l^\omega(\tau',t)]\},$$

which completes the proof. ∎

The intuition of Lem. 6.2 is that constructing an expected value of the service curve leads to an overestimation of the departures. This effect is illustrated in Fig. 6.2. From top to bottom, the figures show the arrivals $A_t(t)$ to a system with time-varying service $S_l(\tau,t)$ and resulting departures $D_t(t)$. In Fig. 6.2a and Fig. 6.2b two sample paths are shown, where in the example of Fig. 6.2a the system transmits all arrivals in the time interval $(0,T]$, whereas in the example of Fig. 6.2b the system forwards only a portion of the arrivals in the interval $(0,T]$. Assuming that each of the service curves occur with a probability of 0.5, the expected value of the service $\mathsf{E}[S_l(t)]$ is illustrated in Fig. 6.2c. Using $\mathsf{E}[S_l(t)]$ overestimates the departures since the concrete timings are lost and $\mathsf{E}[D_t(t)] \le A_t \otimes \mathsf{E}[S_l(t)]$, where $\mathsf{E}[D_t(t)]$ follows from $D_{t_1}(t)$ and $D_{t_2}(t)$ in Fig. 6.2a and Fig. 6.2b. This effect is well-known in queueing theory but is often ignored in the area of available bandwidth estimation as mentioned in [60]. Ignoring the variability and the time scale of the estimate



(a) $D_{t_1}(t) = A_t \otimes S_l^1(t)$ (b) $D_{t_2}(t) = A_t \otimes S_l^2(t)$ (c) $\mathsf{E}[D_t(t)] \leq A_t \otimes \mathsf{E}[S_l(t)]$

Figure 6.2: Example for Lem. 6.2. Assuming random service with two sample paths, $S_l^1(\tau,t)$ and $S_l^2(\tau,t)$ each with a probability of occurrence of 0.5, shown in figures (a) and (b). The expected service $\mathsf{E}[S_l(\tau,t)]$ follows in Fig. (c). Using $\mathsf{E}[S_l(\tau,t)]$ overestimates the departures since the departures in Fig. (b) are less than one. The reason is the timing of the service and the arrivals. The arrivals cannot fully utilize the service in Fig. (b). These accurate timings are lost if the expected value of the service is applied.

are identified as pitfalls for available bandwidth estimation in [60]. Our system identification approach avoids these pitfalls since a service curve expresses service availability on different time scales and the ε-effective service curve accounts for variability as an estimate of the service process.

6.2 MULTI-HOP NETWORKS

In addition to the available bandwidth definition for single-hop networks, a definition for multi-hop networks exists. In the following, we present the relation between the service process $S_{net}(\tau,t)$ from Sec. 2.3.1.1 and the available bandwidth definition for multi-hop networks $\alpha_{net}(\tau,t)$, given in Eq. (3.2).

For time-varying systems $h = (1 \ldots H)$ in series, the end-to-end service process $S_{net}(\tau,t)$ follows from the convolution of the individual service processes $S_h(\tau,t)$ (see Eq. (2.11) and e.g., [29, Chap. 5]). For time-varying systems the convolution operation is not commutative. Whereas the available bandwidth of a multi-hop network is defined by the minimum of the available bandwidth of the individual links in Eq. (3.2). To relate the individual system models to each other by $S_h(\tau, t)/(t-\tau) = \alpha_h(\tau, t)$, we use Lem. 6.1.

Using a constant rate fluid model, each individual system has an available bandwidth equal to a rate r_h, consequentially $S_{net}(\tau, t)/(t-\tau) = \alpha_{net}(\tau, t) = \min_h\{r_h\}$ holds [80]. Generally, the available bandwidth is time-varying and only the relation $S_{net}(\tau, t)/(t-\tau) \leq \alpha_{net}(\tau, t)$ holds. We illustrate the relation for two systems by using Lem. 6.1, it follows for the time interval $(t, t+\delta]$ that

$$\min\left\{\frac{S_1(t, t+\delta)}{\delta}, \frac{S_2(t, t+\delta)}{\delta}\right\} \geq \inf_{\theta \in [0,\delta]}\left\{\frac{S_1(t, t+\theta)}{\delta} + \frac{S_2(t+\theta, t+\delta)}{\delta}\right\}.$$

Since the convolution $S_1 \otimes S_2(t, t+\delta)$ seeks the infimum by passing through the interval from $[t, t+\theta]$, it is less than $\min\{\alpha_1(t, t+\delta), \alpha_2(t, t+\delta)\}$. This result applies to an arbitrary number of systems due to the definition of the convolution and the definition of the multi-hop available bandwidth.

We state in Lem. 6.3 and prove subsequently that equality is recovered in the long-term if each system has a long-term average available bandwidth α_h^∞ with

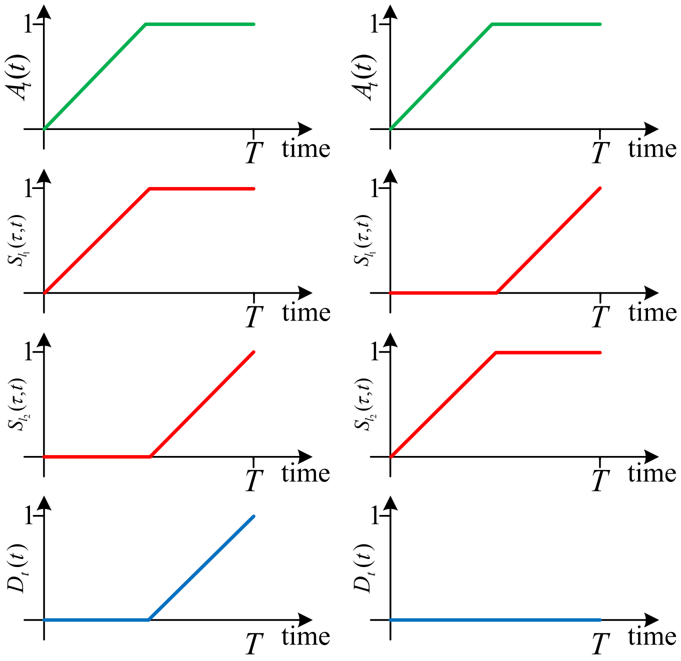
$$\alpha_h^\infty = \liminf_{\delta \to \infty} \frac{S_h(t, t+\delta)}{\delta}. \tag{6.2}$$

Lemma 6.3 ([90]) *Given the end-to-end service process* $S_{net}(\tau, t)$ *as in Eq.* (2.11) *where the service processes of the individual work-conserving systems* $S_h(\tau, t)$ *satisfy Eq.* (6.2). *For any* t *it holds that*

$$\liminf_{\delta \to \infty} \frac{S_{net}(t, t+\delta)}{\delta} = \min_h\{\alpha_h^\infty\}.$$

Proof ([90]) It is sufficient to show the proof for two systems. Due to the properties of the convolution, the result applies to an arbitrary number of systems by repeated application. We rewrite Eq. (2.11) as

$$\frac{S_{net}(t, t+\delta)}{\delta} = \inf_{\theta \in [t, t+\delta]}\left\{\frac{S_1(t, \theta)}{\delta} + \frac{S_2(\theta, t+\delta)}{\delta}\right\}.$$



(a) $D_t(0,T) = A_t(0,T)$ (b) $D_t(0,T) = 0$

Figure 6.3: Example for the impact of the timing on the service in multi-hop networks. The timing in Fig. (a) is beneficial for the arrivals since all arrivals will be processed in the interval $(0,T]$. On the contrary, the timing in Fig. (b) prevents any processing i.e., the departures up to time T are zero.

Now we allow $\delta \to \infty$. If $\alpha_1^\infty > \alpha_2^\infty$ the minimum will be attained for finite θ so that the first term of the sum goes to zero and the second term to α_2^∞. Otherwise, if $\alpha_1^\infty < \alpha_2^\infty$, the parameter θ will tend to ∞ such that the first term becomes α_1^∞ and the second term goes to zero. This gives us

$$\liminf_{\delta\to\infty} \frac{S_{net}(t,t+\delta)}{\delta} = \min\{\alpha_1^\infty, \alpha_2^\infty\}.$$

Finally, if $\alpha_1^\infty = \alpha_2^\infty$ the result holds trivially. ∎

Fig. 6.3 illustrates the impact of the timing on the service in multi-hop networks. The timing in Fig. 6.3a is beneficial for the arrivals since all arrivals will be processed in the interval $(0,T]$. On the contrary, the timing in Fig. 6.3b prevents any processing, i.e., the departures up to time T are zero. Since in Fig. 6.3b the sequence of systems is reversed compared to Fig. 6.3a, this example also shows that the sequence of systems is not arbitrary. The network calculus from Sec. 2.3.1 accounts for the sequence of systems in series by the convolution operation, but the sequence is ignored by the definition of the available bandwidth in Eq. (3.2). The effects illustrated in the example are short-term effects occurring in an interval $(0, T]$. In the long-term these effects vanish because data that are not processed in $(0, t]$, are buffered and processed later as long as packets are not dropped. However, this results in additional delay.

Ignoring the effect of multiple bottleneck links is also identified as one pitfall for available bandwidth estimation in [60]. Obviously, the definition for the available bandwidth in multi-hop networks ignores these effects, as shown in Lem 6.3. Service curves from the network calculus account for these effects. In Chap. 5, system identification methodologies are introduced that determine such service curves. In the next chapter, we implement these methodologies into practical procedures.

7

PROBING PROCEDURES FOR SYSTEM IDENTIFICATION

We now transform the methodology for the identification of the ε-effective service curve, outlined in Chap. 5, into practical estimation procedures. To this end, we implement Thm. 5.1, which specifies the estimate in the max-plus algebra. The estimate follows from delay quantiles measured by constant rate packet train probes. Employing the estimate derived in the max-plus algebra, allows the direct utilization of packet timestamps from probe packets to estimate the delay quantile and permits the consideration of lost packets by an infinite delay.

The estimation procedure requires the determination of the parameter triple $\langle R, N, I \rangle$, where R is the set of probing rates, N the train length, and I the number of iterations of a train with rate r as discussed in the context of Thm. 5.1. The theorem combines delay quantiles $\widetilde{W}^{\xi}(r)$ estimated with probing rates $r \in R$ to an ε-effective service curve. The delay quantile is extracted from the empirical delay distribution, which is obtained from sending I iterations of a train with rate r and length N.

In theory, infinitely long packet trains and an infinite number of iterations are required to estimate the stationary delay quantile. Also, each probing rate $r \in R$ contributes a linear segment to the service curve as depicted in Fig. 5.2. Consequently, only an infinite number of rates achieves an infinitesimal resolution of the curve. Apparently, practical probing requires finite parameters. We point out how to limit these parameters to finite values under two different design goals using statistical tools, which allow us to quantify the imprecision induced by using estimates.

First, we explain how to determine the required parameters with the goal of a fast estimation procedure by a reduction of the probing traffic in Sec. 7.1. This is typically required in productive networks, where the assumption of stationarity applies to limited time scales, so-called *change-free regions* [135]. For example, change-free regions of Internet path characteristics last for a couple of minutes up to several hours [135]. Available bandwidth estimation tools typically pursue a prompt estimation in the range of seconds.

Second, we introduce the parameter selection for the system identification of networks, where typically the setup is a dedicated experimental network for testing purposes only. In such networks the amount of traffic and the measurement duration become less important. Such an estimation procedure allows e.g., the long-term evaluation of networks and protocols deployed therein, which are challenging to be analyzed analytically due to their complexity. Stationarity can be enforced in such networks for an unlimited time.

The work in Sec. 7.1 is based on cooperation with Markus Fidler and Jörg Liebeherr and partially available in [88, 89, 90]. Sec. 7.2 is joint work with Markus Fidler.

7.1 PROCEDURE FOR NETWORKS WITH CHANGE-FREE REGIONS

Design goals for available bandwidth estimation tools are promptness, performing the measurements in change-free regions, and the generation of a small amount of probing traffic. In this section, we derive the selection of the parameter triple $\langle R, N, I \rangle$ for our service curve estimation procedure under consideration of these goals. The entire procedure is depicted in Fig.7.1[10] and is described in the following subsections. At the beginning, we explain the selection of the probing rates using a combination of a binary increase and binary search algorithm in Sec. 7.1.1. The transition between the binary increase phase and the binary search phase depends on the detection of steady state delays for a probing rate. The estimation of steady state delays is introduced in Sec. 7.1.2 and relies on Lem. 5.2, which gives evidence under which conditions a steady state delay distribution exists. We give details on the adaptation of the train length to detect stationarity, present a heuristic to reduce the required train length to detect stationarity, introduce the determination of quantiles from delay distributions, and describe the prediction of the quantile for

---

[10] A similar flowchart of our procedure is shown in [120, 121], whereas the authors implement roughly our methodology derived in the min-plus algebra.

small values of ξ from a small number of samples. All of the following experiments were conducted in the Emulab testbed [130] or in a local testbed using the Emulab software [1]. Both testbeds create isolated and controlled experiments using real hardware components and software. Fig. 7.2 shows a multi-hop network with multiple bottleneck links in series offering a capacity of 100 Mbps and 10 ms delay. The access links of the cross traffic senders, the cross traffic receivers, the probe traffic sender, and the probe traffic receiver have a capacity of 1 Gbps and 0 ms delay. In this topology, cross traffic enters the network at one router and departs from it at the next router in series. Cross traffic and probe traffic are multiplexed at these bottleneck links. The described network topology is implemented in the testbeds with different number of bottleneck links.

For the experiments presented in this section, we solely use a single-hop topology, i.e., the number of bottleneck links from the topology presented in Fig. 7.2 reduces to one. This topology consists of one probe traffic sender and receiver pair, one cross traffic sender and receiver pair, and two routers, whereas cross traffic and probe traffic are only multiplexed at one hop, which is the first router subsequent to the senders. Topologies with multiple bottleneck links are evaluated in the subsequent chapter. For the default experimental setup, we use a priority scheduler with priority for the cross traffic to ensure linearity and large buffer sizes to prevent packet loss (the buffer size was set of 10^6 packets on the routers). The default transport protocol for probe and cross traffic is the user datagram protocol (UDP).

Probe traffic is generated by the tool Rude/Crude [72]. We configure the sender Rude to transmit constant rate traffic with rate r using equally spaced packets of size 1500 Byte (IP packet size) with a gap of $\frac{1500\ Byte}{r}$ seconds. Timestamps are added to the packets by the sender Rude as well as by the receiver Crude and are stored at the receiver. Delays of packets are calculated from timestamps in Matlab and processed in Matlab and R [2].

We generate random cross traffic with a long-term rate of 50 Mbps, i.e., the service process has also a long-term available bandwidth of 50 Mbps as defined in Sec. 3.1. Cross traffic packets are generated equally spaced in time but with packet sizes drawn from a truncated Exponential or truncated Pareto distribution with shape parameter of 1.5. The two distributions stand for different kinds of traffic; the traffic

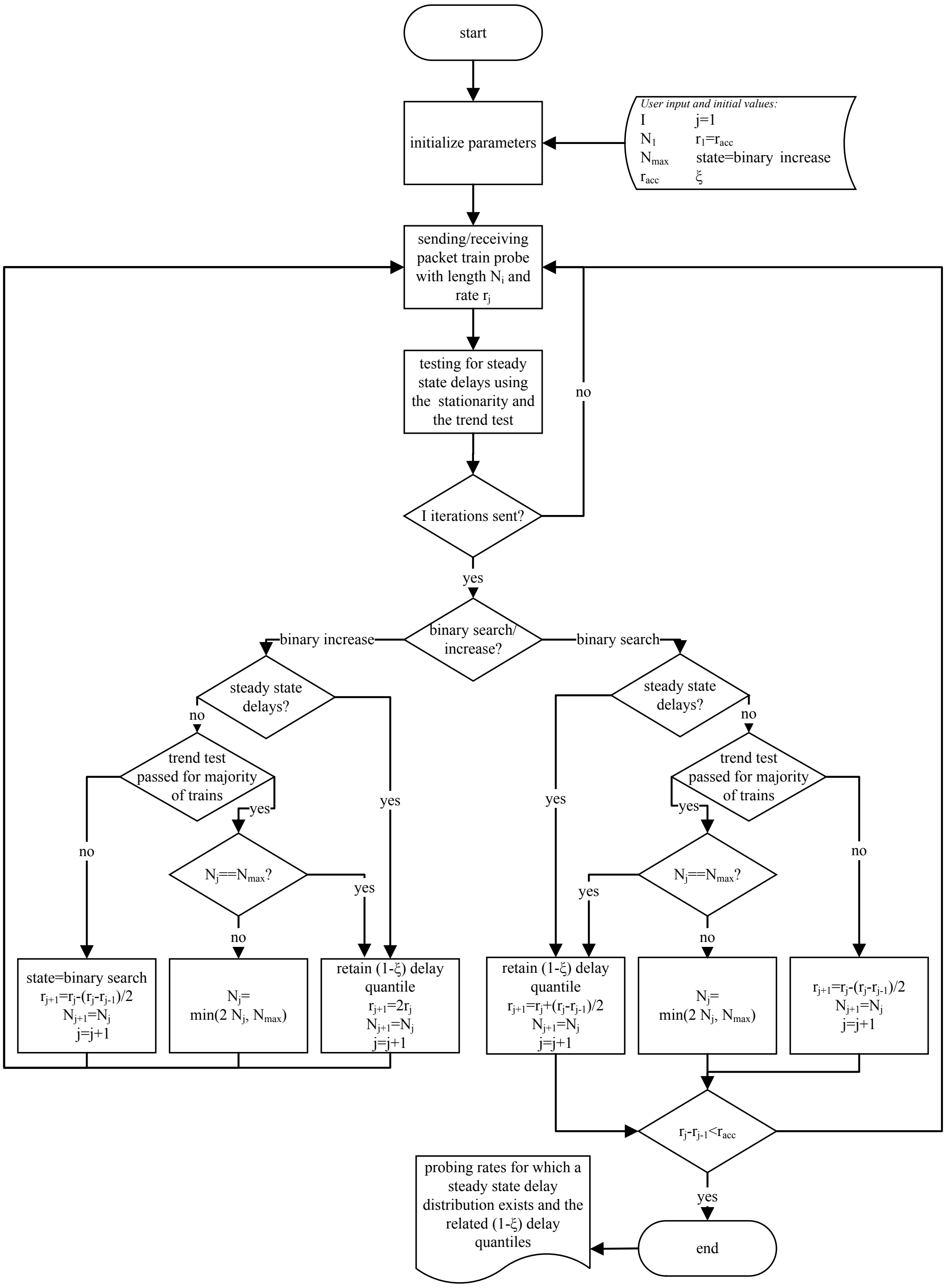


Figure 7.1: Flow chart of the probing procedure for networks with change-free regions, for a detailed description see Sec. 7.1.1 to Sec. 7.1.2.

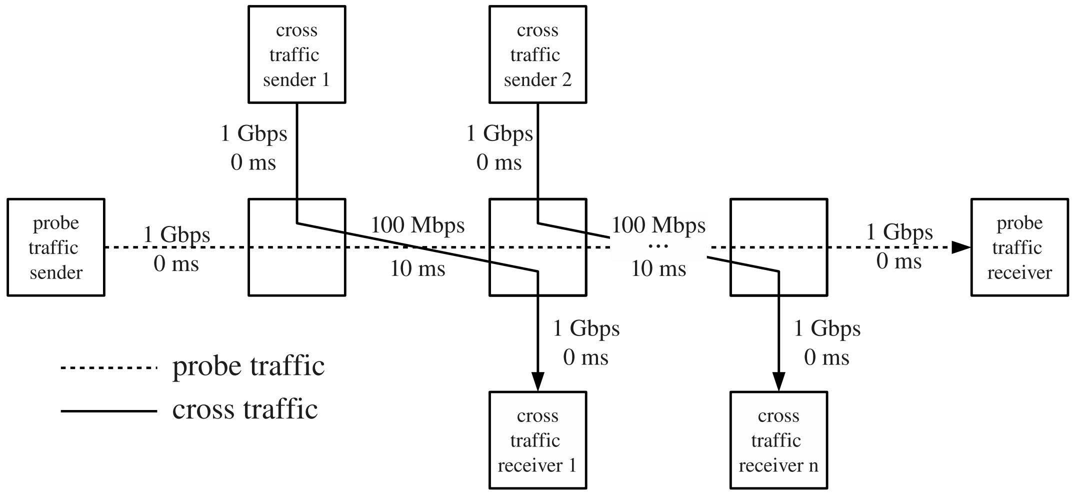


Figure 7.2: Multi-hop network with multiple 100 Mbps bottleneck links each with a delay of 10 ms.

following an Exponential distribution for friendly traffic, i.e., if averaged, it becomes smooth on short as well as on long time scales, whereas the Pareto traffic represents unfriendly traffic, which stays bursty, even if averaged on long time scales. For the generation of cross traffic, we use the tool D-ITG [18]. It truncates the maximum payload to 64000 Byte to comply with the maximum IP payload size, which leads to truncated distributions. Further fragmentation occurs due to the maximal transfer unit of 1500 Byte of Ethernet packets at the IP layer. Additionally, we require that the probe traffic sender and the receiver are synchronized in time to measure exact delays, therefore we use the network time protocol (NTP). In real networks, clocks can be asynchronous. An offset between the sender and the receiver will result in a shift of the estimated service curve, a shift to the right for a positive offset between the sender and the receiver, i.e., the local time of the receiver is larger than on the sender, and a shift to the left for a negative offset between the sender and the receiver. For details on asynchronous clocks due to a shift, we refer to [80]. For further effects caused by clocks that can occur in networks see e.g., [112]. In our testbeds we assume reliable clocks with a precision within one millisecond since time servers for synchronization are located in the local area network as described in [97].

7.1.1 *Selection of Probing Rates*

The selection of the set of probing rates R in Thm. 5.1 defines on the one hand the resolution of the service curve $T_S^{\varepsilon}(n)$ since each rate $r \in R$ contributes a linear segment, on the other hand each probing rate increases the violation probability $\varepsilon = \sum_{r \in R} \xi$. To balance the design goals of a high resolution and a small violation probability, we use a combination of a binary increase and a binary search algorithm, which is similar to the procedure in [58]. The selection is based on the existence of a steady state delay distribution for a probing rate. We refer to Sec. 7.1.2 for details on the detection of it based on I iterations performed for a probing rate r. During the binary increase phase the current rate r_j is doubled as long as steady state delays are detected for this rate, i.e., the next rate r_{j+1} becomes $r_{j+1} = 2r_j$, where $j = \{1, 2, 3, \ldots\}$ increases by one for each further probing rate. The first time no steady state is detected for rate r_j, a binary search is started in the interval $[r_{j-1}, r_j]$, with the next probing rate $r_{j+1} = r_j - \frac{r_j - r_{j-1}}{2}$. Further probing rates follow from halving the interval length: if a steady state is detected for r_j, $r_{j+1} = r_j + \frac{r_j - r_{j-1}}{2}$, if not, $r_{j+1} = r_j - \frac{r_j - r_{j-1}}{2}$. The initial rate and the termination condition are defined by r_{acc}, which specifies the desired resolution. The initial rate is $r_1 = r_{acc}$ and the algorithm terminates if $r_{j+1} - r_j < r_{acc}$. For the set of all probing rates, we refer to R', but only rates for which a steady state delay distribution exists contribute to the service curve. We collect these rates in the set R.

This algorithm enables a fast estimation and contributes significant rates since the binary increase ramps up the probing rate fast to find an interval that contains the long-term available bandwidth. In addition, the binary search presents a fast search in the interval to detect a rate close to the long-term available bandwidth α^{∞}, which is in $[\alpha^{\infty} - r_{acc}, \alpha^{\infty})$. The number of rates in R' is $2\lfloor \log_2(\alpha^{\infty}/r_{acc}) \rfloor + 2$, where $\lfloor \log_2(\alpha^{\infty}/r_{acc}) \rfloor + 2$ rates follow from the binary increase phase and $\lfloor \log_2(\alpha^{\infty}/r_{acc}) \rfloor$ rates from the binary search. Without further knowledge on the rates, the binary search algorithm is optimal with respect to its complexity $O(log(n))$ [99].

In the experiments in this section, a typical set of probing rates, which follows from the described algorithm for $\alpha^{\infty} = 50$ Mbps and $r_{acc} = 4$ Mbps, is

$R' = \{4, 8, 16, 32, 64, 48, 56, 52\}$ Mbps. At last, the rates $R = \{4, 8, 16, 32, 48\}$ contribute a linear segment to the service curve because only these rates exhibit a steady state delay distribution, which follows from Lem. 5.2.

7.1.2 *Estimation of Delay Quantiles*

Steady state delay distributions for the probing rates are required to implement Thm. 5.1. Estimating steady state delays implies that the delay distribution is independent of the train length. We prove the existence of such distributions in Lem. 5.2; on the condition that stationary arrivals are below the long-term available bandwidth of a network path with stationary service. Typically, the long-term available bandwidth of a network path is unknown, therefore it is not known beforehand if a probing rate produces a steady state delay distribution. Here, we present the detection of the steady state by using a stationarity test in Sec. 7.1.2.1. Since also the train length N that is required to detect stationary delays is not known a priori, we show an adaptation procedure in Sec. 7.1.2.2. Furthermore, the procedure of the train length adaptation offers a heuristic for the detection of probing rates that contribute a steady state delay distribution. In Sec. 7.1.2.3, we use this heuristic to reduce the probing traffic for the estimation of the long-term available bandwidth and of service curves that hold for a limited scale. This reduction enables a prompt estimation procedure. Finally, we show how to estimate the delay quantile of the empirical delay distribution obtained by iterating packet trains I times. Using confidence intervals for quantiles, we are able to quantify the accuracy of the estimate. Moreover, we present how to predict the tail of the distribution for small ξ from a small number of samples in Sec. 7.1.2.4.

7.1.2.1 *Stationarity Test*

To obtain samples of the delay in the steady state, we test the series of delays measured by a packet train for stationarity. For a packet train of length N, each packet returns a sample for the delay series $W(n)$, where $n \in [0, N-1]$. We test the series $W(n)$ for stationarity with the unit root test from Elliot, Rothenberg, and

Stock (ERS) [39, 107]. The test uses an auto-regressive moving average model for the data series with the null hypothesis that the auto-regressive model has a unit root implying non-stationarity. If the null hypothesis is rejected, we assume the data series to be stationary. The decision is based on the test statistic; if it is below a critical value, the null hypothesis is rejected. In detail, we use the P-test with the critical values from [39] and a level of 0.1 for the experiments outlined in Chap. 7 and Chap. 8.

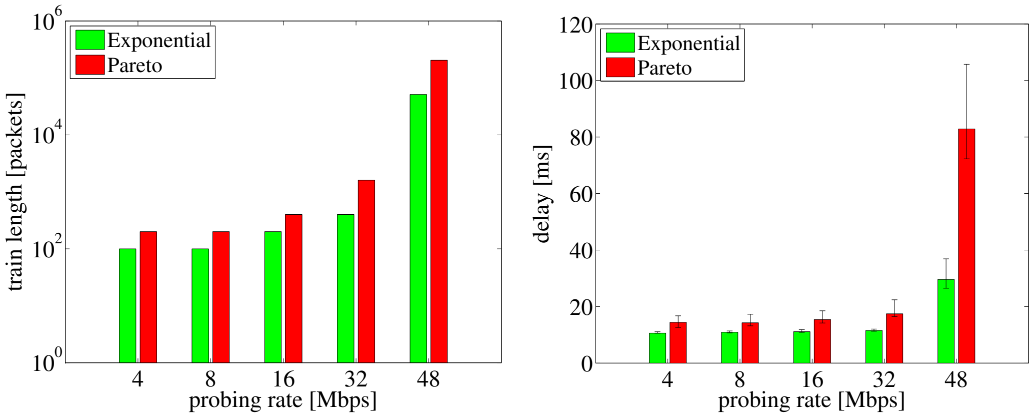
To ensure independence of the samples and because of the delay distribution is converging to the steady state for $n \to \infty$, we keep only the last delay sample $W(N-1)$ of each train and discard previous samples of that train due to possible correlations between delay values $W(n)$ of one train.

If stationarity is not detected for a train, we set the delay sample to infinity. Furthermore, we perform I iterations for each rate, as discussed in Sec. 7.1.2.4. If the ratio of infinite samples is equal to or greater than ξ the delay quantile $W^{\xi}(r)$ is infinite, too, and the rate r does not contribute to the service curve due to the minimum in Thm. 5.1. Setting the delay sample to infinity is therefore conservative.

7.1.2.2 *Adaptive Train Length*

As Lem. 5.2 indicates, the delay is stochastically increasing with the train length and converges to the steady state distribution. However, the train length to detect stationarity is not known a priori, therefore, we use an adaptive procedure to extend it. Detecting non-stationarity of a delay series can be caused obviously by non-stationarity but also by observing delays during the transient phase of the system, which occurs at the beginning of a packet train. In the second case, if the train length was chosen too short, a longer packet train would detect stationarity. We differentiate between these cases by evaluating the delay series of I iterations for rate r using the ERS statistic.

In general, we start with an initial train length N_1, and if the ratio of stationary trains of I iterations is greater than $1-\xi$ for this train length, we proceed with the next probing rate. If the ratio is less than $1-\xi$, we perform an additional test on each train that we name *trend test*. For the trend test the ERS statistic for the delays of the first half of each train is computed and is compared to the ERS statistic of



(a) required train length (b) steady state 0.95-delay quantile

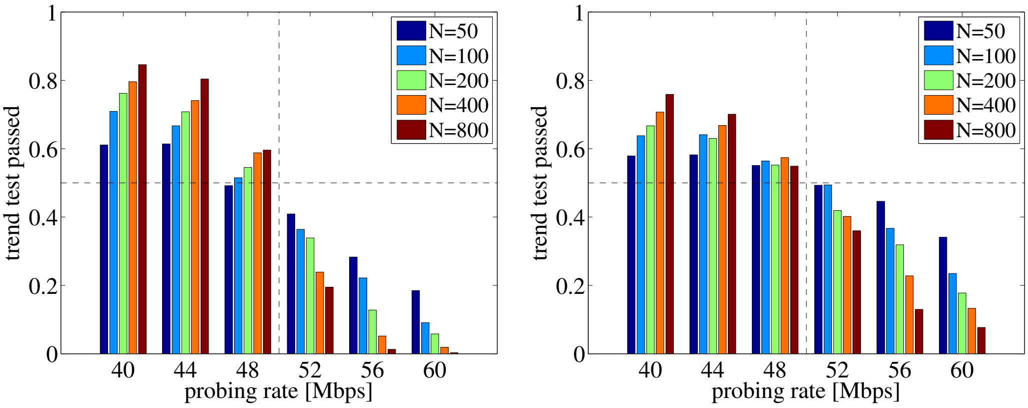
Figure 7.3: Fig. 7.3a shows the train length used to observe steady state delays in a single-hop network. The bottleneck link has a long-term available bandwidth of $\alpha^{\infty} = 50$ Mbps leftover by Exponential or Pareto cross traffic. Fig. 7.3b presents estimates of the 0.95-delay quantiles with 0.95 confidence intervals taken from the steady state delay distribution observed with the train length shown in Fig. 7.3a.

the entire train. This test is passed if the ERS statistic decreases with the entire train indicating that a longer train could see a stationary delay series.

The adaptation of the train length is based on a majority decision: If the majority of I iterations passes the trend test, the train length is doubled, i.e., $N_j = 2N_j$, and the probing is performed anew with the same rate, else the existence of a steady state delay distribution is rejected and the probing proceeds according to Sec. 7.1.1 with the next probing rate.

For the network setup described at the beginning of Sec. 7.1, we show the train length used to observe stationarity for a ratio of $1-\xi = 0.95$ of $I = 250$ iterations per probing rate in Fig. 7.3a. The train length is adapted according to the procedure described in this section and the probing rates arise from the algorithm described in Sec. 7.1.1 with $r_{acc} = 4$ Mbps. The minimum train length was set to $N_1 = 100$ packets. The figure illustrates that the required train length increases significantly when the probing rate approaches to the long-term available bandwidth of 50 Mbps.

The adaptation of the train length is based on the trend test, which we evaluate below. Again, we use the network setup introduced at the beginning of Sec. 7.1. The long-term available bandwidth of the network path is 50 Mbps. Therefore, we expect that the relative frequency for passing the trend test for a probing rate below 50 Mbps is greater than 0.5 since the rate is below the long-term available bandwidth,

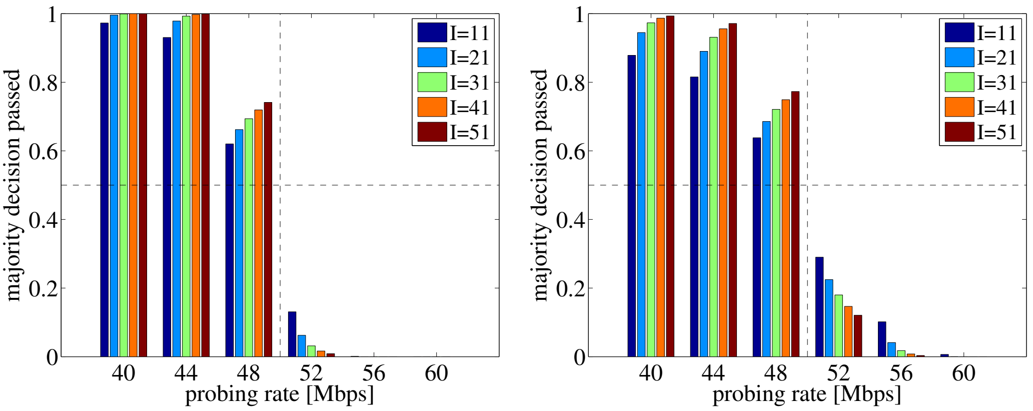


(a) Exponential cross traffic

(b) Pareto cross traffic

Figure 7.4: Relative frequency of 1000 packet trains that pass the trend test for probing rates close to the long-term available bandwidth of $\alpha^{\infty} = 50$ Mbps and train lengths $N = \{50, 100, 200, 400, 800\}$. In all cases, except one, the classification is correct because below 50 Mbps more than the majority of the trains pass the test and above 50 Mbps less than the majority of the trains pass the test. Only for the Exponential traffic, a rate of $r = 48$ Mbps, and $N = 50$ the classification is barely below the majority. However, the classification is challenging in cases close to the long-term available bandwidth using very short packet trains.

and less than 0.5 for a probing rate above 50 Mbps. The relative frequency of trend tests passed for 1000 iterations is shown in Fig. 7.4 for probing rates in the interval from 40 to 60 Mbps with a step size of 4 Mbps and five train lengths in the interval from 50 to 800 packets. The results show that for a train length of 100 packets and more the relative frequency of the trend tests passed or failed, classifies correctly the existence of a steady state. For a length of 50 packets the test only wrongly rejects the steady state for the Exponential distribution with a probing rate of 48 Mbps; however, the majority is missed slightly. The correct decision is more evident for probing rates distant from 50 Mbps and also more evident for the friendly Exponential traffic. As described above, to improve the robustness the heuristic is based on a majority decision of I iterations for probing rate r. For each iteration the trend test is applied to the packet train, and the majority of the trend test passed or failed is evaluated to decide if the steady state is reached or not. The probability that more than the majority of the trains pass the trend test is shown for a train length of $N = 200$ in Fig. 7.5. We assume that each packet train represents a Bernoulli trial, therefore, the probability for the majority decision in Fig. 7.5 follows from the binomial distribution. The results are shown for a small number of iterations $I = \{11, 21, 31, 41, 51\}$ to demonstrate the robustness for a small amount



(a) Exponential cross traffic

(b) Pareto cross traffic

Figure 7.5: Evaluation of the robustness of the majority decision on the trend test for $I = \{11, 21, 31, 41, 51\}$ iterations. The correct classification significantly increases with the number of iterations; already for a small number of iterations a correct classification is feasible. The figure shows results based Fig. 7.4 for a train length of $N = 200$.

of probing traffic. We choose odd numbers to ensure that a majority exists. Already for a few iterations of eleven the classification makes a correct decision for rates close to 50 Mbps, for a probing rate of 48 Mbps the decision is correct in about 60 percent of the cases and for a rate of 52 Mbps for about 70 percent of the cases (much better for Exponential traffic).

This section shows the need for long trains to estimate the steady state delay distribution. In the next section, we present that the design goal of a prompt probing procedure using a small amount of probing traffic is achievable by using the majority decision presented in this section as a heuristic.

7.1.2.3 *Estimates from Short Packet Train Probes*

Fig. 7.3a displays the need for long packet trains to observe the steady state delay distribution. However, the trend test gives a robust indication whether a steady state delay distribution exists or not using much shorter packet trains. We make use of this indication to establish a prompt procedure for the estimation of, first, the long-term available bandwidth and, second, service curves that apply to a limited scale as depicted in Eq. (5.8). Short packet trains are implemented by limiting the train length to a maximum N_{max}. The decision whether a steady state delay distribution exists or not is based on the trend test if the maximal train length is utilized and stationarity is not detected. In detail, if the train length has reached its maximum. If the trend test is passed for the majority of I iterations, we assume that for the rate r a steady state delay distribution exists. If the test is passed for less than the majority, we assume that no steady state exists. This heuristic shortens the required train length to detect rates for which steady state delays exists, but obviously, the extracted delay samples do not reflect the steady state delay distribution.

AVAILABLE BANDWIDTH ESTIMATION: For the available bandwidth, we seek for the highest rate that observes steady state delays. Using the heuristic described before, we estimate solely whether a steady state delay distribution exists or not. As Fig. 7.4 and Fig. 7.5 show the trend test already offers robust estimates with short packet trains and a small number of iterations. The long-term available bandwidth of 50 Mbps in the experiment is closely tracked. The estimate is 48 Mbps, which is the next probing rate below 50 Mbps assuming the rate adaptation from Sec. 7.1.1 with $r_{acc} = 4$ Mbps.

SERVICE CURVE ESTIMATION WITH LIMITED SCALE: If we consider also the delay samples, while allowing only a maximum train length N_{max}, the procedure also returns delay estimates. However, these estimates are not from the steady state distribution. These delay estimates provide a service curve that applies to a limited scale of N_{max} packets as specified in Eq. (5.8). In [25], such scales are introduced as time scale bounds in the min-plus algebra. Such bounds are useful if bounds on the maximal backlog, the maximal delay, or busy periods are known [25].

Evidently, probing traffic cannot be reduced as for the case of the long-term available bandwidth estimation since the extraction of delay quantiles requires a reasonable number of iterations. However, the robustness of the majority decision of the trend test makes a strong reduction of the probing traffic possible with respect to the train length.

7.1.2.4 *Estimation of the Delay Distribution*

To obtain an estimate of the delay quantile $\widetilde{W}^{\xi}(r, n)$ for rate r, which is required in Thm. 5.1, the empirical delay distribution is estimated by samples from I packet trains. The delay of the last packet of each train is used to compute the distribution. Other samples are discarded to avoid correlations between them. Furthermore, we assume independent samples of the delay distribution by using packet trains starting at random times, see [16] for a discussion. Finally, the quantile follows from Eq. (5.2).

Using a finite number of samples leads to an unavoidable inaccuracy of the delay distribution, which vanishes with an increasing number of samples. This resulting inaccuracy can be quantified by using confidence intervals for quantiles, which follow from the binomial distribution [73]. This allows to state the expected range of the confidence interval for a given number of samples or to adapt the number of samples until a given accuracy is achieved. Throughout all the experiments in this work, we use a fixed number of iterations.

In Fig. 7.3b, we present delay quantiles for $1 - \xi = 0.95$ of the steady state delay distribution and the related 0.95 confidence intervals resulting from $I = 250$ iterations. The figure shows that the confidence intervals increase as the probing rate converges to 50 Mbps. This effect is stronger for the more bursty Pareto traffic. Throughout the entire thesis, we extract the delay quantiles directly from the empirical distribution by using Eq. (5.2). Nevertheless, we state below a possibility to reduce the number of iterations. This could be employed if a small ξ is requested, else it would involve a large number of samples. The reduction can be achieved by the prediction of the tail of the delay distribution. For the prediction, we use the peaks over threshold (POT) method from the extreme value theory, see e.g., [17]. Based on a number of independent and identically distributed samples, the parameters of a generalized Pareto distribution are estimated to predict the delay quantile $\widetilde{W}^{\xi}(r)$. The accuracy of the prediction is also indicated by confidence intervals. To apply the POT method the distribution must be in the domain of attraction of an extreme value distribution, see [17] for details. We verify the assumption by using the tests from [36, 77].

We apply the method to the delay distribution of the delay quantiles shown in Fig. 7.3b measured by 250 iterations. We predict the tail to extract delay quantiles for small values of ξ and compare them to a delay distribution measured from 2000 iterations. Fig. 7.6 presents the results for the probing rate of 48 Mbps with confi-

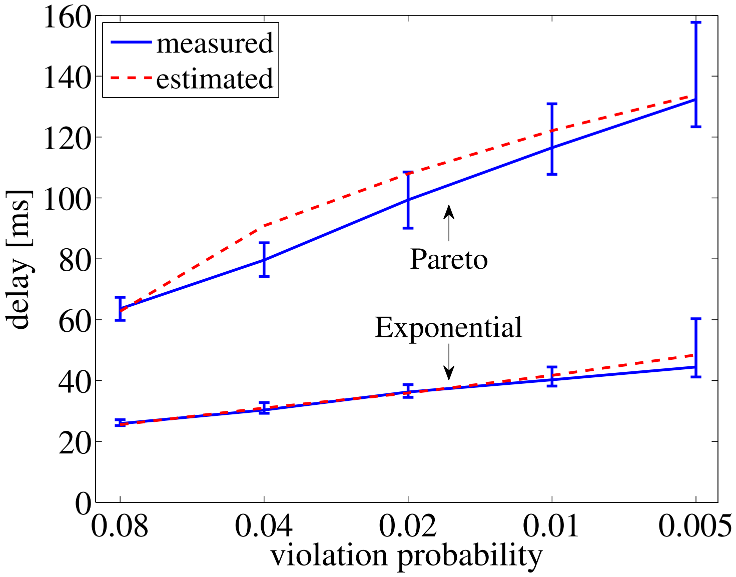


Figure 7.6: Comparison of estimates of the quantiles $W^{\xi}(r)$ obtained from 2000 samples using Eq. (5.2) and the estimate received by the POT method using 250 samples for $r = 48\ Mbps$. The POT method shows a good fit.

dence intervals for the delay quantiles derived by the POT method (the threshold parameter was set to 0.9). For the Exponential distributed traffic, both curves show a very good fit; for the more bursty Pareto distributed traffic the curve estimated by the POT method lies in the majority of the cases in the confidence interval of the curve measured by 2000 iterations.

7.2 PROCEDURE FOR NETWORKS WITH STATIONARY SERVICE

The scope for this procedure is the evaluation of network topologies and protocols in dedicated networks, i.e., controlled environments for the systems under test. In such environments the measurement time and the amount of probing traffic are less important than in the environments assumed in Sec. 7.1. Such testbeds facilitate also a stationary environment on the long run. This allows the analysis of effects, which affect the behavior on long time scales. We turn our attention to gathering a more detailed view of the system behavior. In the following, we discuss the procedure to define the probing parameter triple $\langle R, N, I \rangle$ with reference to the previous presented procedure from Sec. 7.1. We modify the selection of the probing rates and further obtain I independent delay samples of one train instead of multiple independent trains. This procedure is implemented in a separate software using C++ and the interface to the software R. Fig. 7.7 presents the flow chart of the procedure.

7.2.1 *Selection of Probing Rates*

To obtain a more detailed view of the system behavior, we use a linear increase of the probing rate. The probing starts with a rate $r_1 = r_{acc}$. The next rate follows by a linear increase $r_{j+1} = r_j + r_{acc}$. The probing stops for the first rate for which no steady state delays are detected.

Compared to the binary increase/binary search algorithm from Sec. 7.1.1, the network is usually probed by a larger number of rates. The binary increase algorithm described in Sec. 7.1.1 raises the probing rates fast, thus it may occur that probing rates that contribute meaningful parts to the service curve estimate are missed. For the long-term available bandwidth of α^{∞} the set of probing rates contains $\left\lfloor \frac{\alpha^{\infty}}{r_{acc}} \right\rfloor + 1$ different rates, e.g., choosing the same example from Sec. 7.1.1 with $\alpha^{\infty} = 50$ Mbps and $r_{acc} = 4$ Mbps results in thirteen probing rates compared to eight for the binary increase/binary search algorithm.

7.2.2 *Estimation of Delay Quantiles*

For the estimation of the steady state distribution, we rely again on the stationarity test described in Sec. 7.1.2.1. The difference is that we estimate the delay distribution from one long train taking samples randomly.

The initial train length N_1 is composed of an offset N_{off} and a second part with mean length N_1^S resulting in a length of $N_1 = N_{off} + N_1^S$. We use an offset, where no sampling is performed, to avoid sampling during transient phases. Subsequently, I samples are collected randomly. The distance between samples follows a geometric distribution with a mean of N_1^S/I. Random sampling is used to avoid the observation of possible periodicities inherent in the system [16]. After gathering I samples, three tests are performed on the data series, the stationarity test described

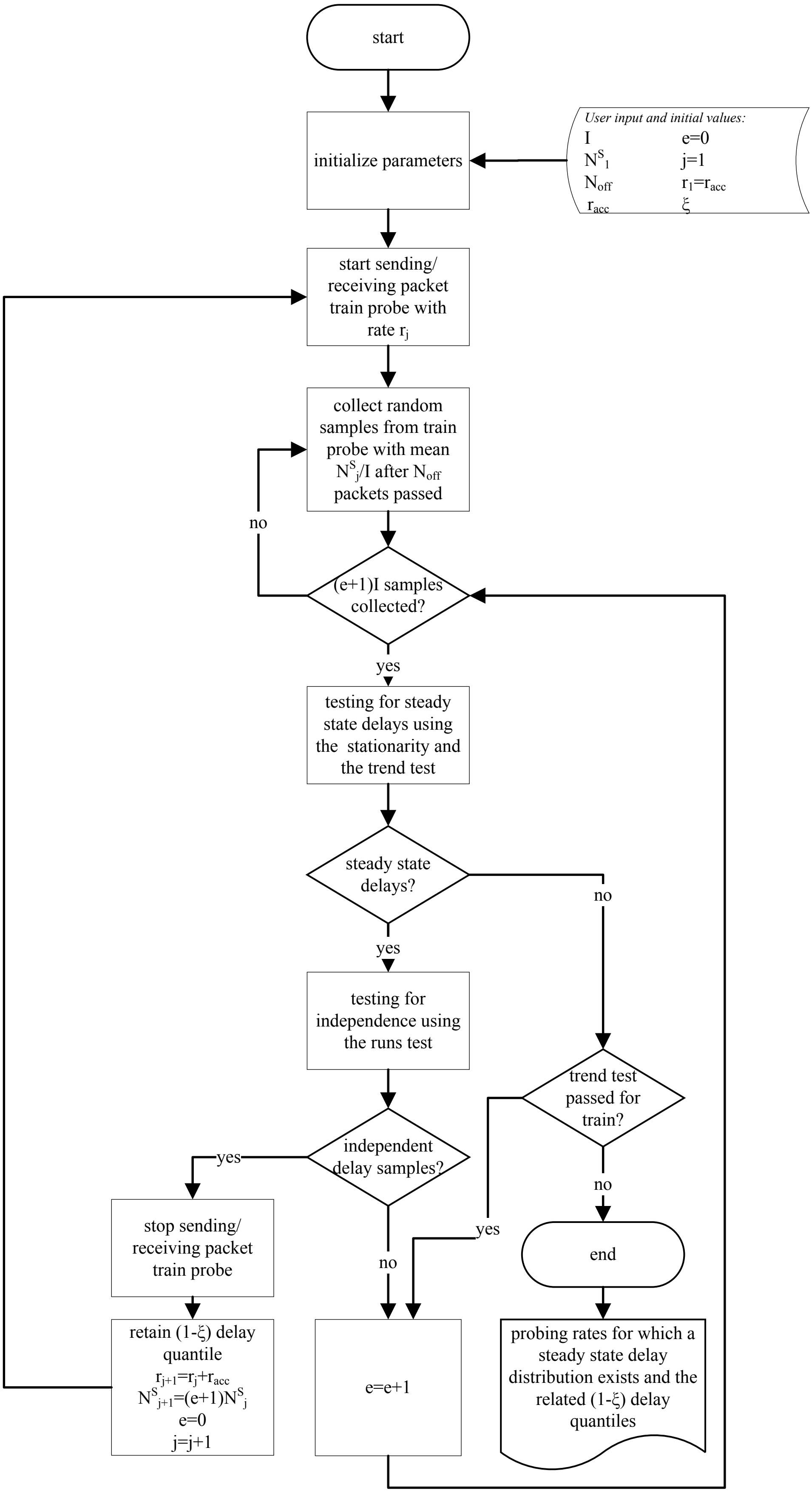
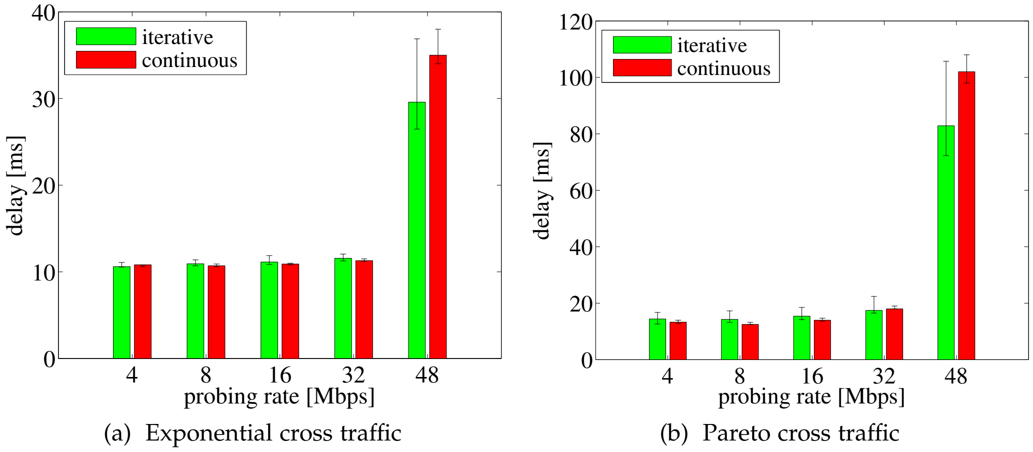

start
initialize parameters
User input and initial values:
I
e=0
N^S_1
j=1
N_{off}
$r_1=r_{acc}$
r_{acc}
ξ
start sending/ receiving packet train probe with rate r_j
collect random samples from train probe with mean N^S_j/I after N_{off} packets passed
no
(e+1)I samples collected?
yes
testing for steady state delays using the stationarity and the trend test
steady state delays?
no
yes
testing for independence using the runs test
trend test passed for train?
yes
independent delay samples?
no
yes
no
end
stop sending/ receiving packet train probe
retain (1-ξ) delay quantile
$r_{j+1}=r_j+r_{acc}$
$N^S_{j+1}=(e+1)N^S_j$
e=0
j=j+1
e=e+1
probing rates for which a steady state delay distribution exists and the related (1-ξ) delay quantiles

Figure 7.7: Flow chart of the probing procedure for networks with stationary service, for a detailed description see Sec. 7.2.1 and Sec. 7.2.2.

in Sec. 7.1.2.1, the trend test from Sec. 7.1.2.2, and the runs test [49] for testing the independence of the samples. Passing the stationarity and the runs test means that the samples are independent and identically distributed. The procedure continues with the next probing rate if both tests are passed. The train length is increased, first, if no stationarity is detected for the current rate but the trend test is passed, or second, if stationarity is detected but the runs test is not passed. Passing the trend test suggests that a longer train could see stationarity, and not passing the runs test indicates that samples are not independent. Using longer trains may lead to stationarity and independent samples. The probing is stopped if neither the trend test nor the stationarity test are passed.

The increase of the train length is performed without the interruption of the current measurement. This approach reduces the execution time in comparison to the procedure described in Sec.7.1, at which trains have to be repeated if the train length is adapted. During the statistical tests the probing is continued for the current rate. If stationarity or finally no stationarity is detected, the probing is stopped, otherwise the probing proceeds, until the next tests are performed after additional I samples are collected, thus the mean length of the extended train length becomes $N_{off} + (e+1)N_j^S$ with the eth extension. Since we keep the mean sampling interval N_j^S/I, we have $(e+1)I$ samples after the eth extension. To reduce the number of samples to I and simultaneously increase the sampling interval to $(e+1)N_j^S/I$, we only pass each $(e+1)$th sample to the tests. If the stationarity test and the runs test are passed, I independent samples from the steady state delay distribution of rate r are returned. The estimate of the delay quantile $\widetilde{W}^{\xi}(r,n)$ results from these samples for rate r and the probing is continued following the procedure in Sec. 7.2.1 with sample interval $N_{j+1}^S = (e+1)N_j^S/I$.

The probing procedure described in Sec. 7.1.2.2 requires multiple trains to increase the robustness of the detection of a steady state distribution by using a majority decision based on the outcome of the trend test. In Sec. 7.1.2.3, this robustness is achieved since the majority decision is assumed to be sufficient to observe the steady state distribution, and it allows the reduction of the train length. Here, one train per probing rate is used, which impairs the robustness of the majority decision on the trend tests. This is sufficient for this procedure because the decision whether



(a) Exponential cross traffic (b) Pareto cross traffic

Figure 7.8: Comparison of delay quantiles obtained by iterative sampling described in Sec. 7.1.2.4 and continuous sampling. For all probing rates both procedures show similar results since all confidence intervals overlap.

a steady state distribution exists or not relies solely on the stationarity test and the trend test is just used for the adaptation of the train length. However, this prevents the estimation of service curves for limited scales, but it allows for the continuous extraction of delay samples of one train.

We evaluate the continuous sampling procedure by repeating the experiment from Sec. 7.1.2.2 and estimate the delay quantiles for the rates $R = \{4, 8, 6, 32, 48\}$ with the parameters $\xi = 0.05$, $N_1^S = 100$, and $N_{off} = 10^6$. The estimated delay quantiles and related confidence intervals are presented in Fig. 7.8. For the procedure described in Sec. 7.1.2.2, the results are taken from Fig. 7.3b. The overlapping confidence intervals indicate approximately equivalent estimates for the iterative and continuous approach for all probing rates and for the Exponential as well as the Pareto cross traffic. The results indicate that both procedures are practical for the determination of the quantiles.

In the next chapter, we apply the presented probing procedures to various networks with random service by simulation and experiment. For the comparison to available bandwidth estimation tools and for the estimation for service curves with a limited scale, we apply the procedure from Sec. 7.1. For networks that feature correlations on long time scales, we employ the procedure from Sec. 7.2.

#

8

APPLICATION OF SYSTEM IDENTIFICATION

Using the probing procedures from Chap. 7, we apply our system identification approach to various networks and protocols by simulations and experiments. Thereby, we provide a validation of the estimation procedures. We conduct the validation by comparing the estimated long-term available bandwidth to the long-term available bandwidth that is known from the experimental setup or that is analytically deduced. Moreover, we validate the procedure by comparing estimates to analytical service curves, where they are known.

In Sec. 8.1 and Sec. 8.2, we start with comparative evaluations to known system identification procedures. First, we compare our procedure to well-known available bandwidth estimation tools, which are system identification procedures that usually describe the system by a single value. The similarities between the available bandwidth and the service curves from the network calculus are shown in Chap. 6. The service curve estimate consists of multiple rate segments, whereby the segment with the highest rate is an estimate of the long-term available bandwidth. Moreover, our estimation procedure provides the identification of the system by a service curve. Service curve estimates include information on the time scales of the availability of the service of the system, which varies in networks with random service. We utilize this additional information in the subsequent sections. Second, we compare our procedure to another service curve estimation procedure from [80] that is based on a deterministic system model. This evaluation highlights the impact of the assumed system class for identification i.e., systems with deterministic or random service.

After the comparative evaluation, the service of networks with different sources of randomness is evaluated, including networks where the randomness originates from random cross traffic, time-varying channels[11], and protocols. In Sec. 8.3, we initiate the evaluation by service curve estimation of wired networks, where

---

[11] Recently, our procedure was applied in [120] to estimate the service of a Rayleigh fading channel.

typically a constant capacity of the link can be assumed. The evaluation includes results for different kinds of random cross traffic, scheduling disciplines with unlimited and limited queues, single-hop as well as multi-hop networks. Estimates are obtained by probe packet trains with non-restricted and restricted length. We show the representation for the service curve as available bandwidth, thereby we also validate the estimate by comparison to an analytical reference.

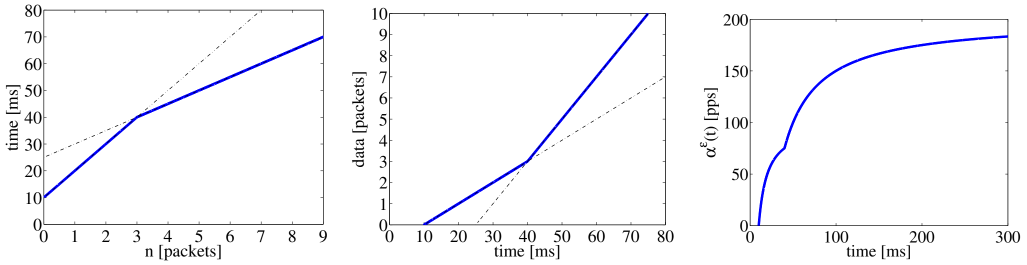
Next, Sec. 8.4 presents estimates for an IEEE 802.11a wireless network. In such networks, the channel can no longer be assumed to be deterministic as in wired networks due to interference and collisions. Collisions occur because the channel is shared between stations in the network. An additional source of randomness is the MAC protocol. Using this protocol, stations have to wait a random time before they access the channel. Thereby, the MAC protocol achieves a fair channel usage between wireless stations in the long-term [21].

Window flow control with static and adaptive windows is evaluated in Sec. 8.5. Therein, we show the applicability of our service curve estimation procedure to such protocols by simulation.

Finally in Sec. 8.6, we evaluate TCP, which is the prevalent transport protocol in the Internet [92], by service curve estimation. TCP implements a congestion control algorithm that adapts the transmission rate to the utilization of the network, e.g., on basis of packet loss, which occurs randomly in the network.

TCP and the MAC protocol in IEEE 802.11a wireless networks are non-deterministic and non-work-conserving, i.e., even if the link or channel is idle, the protocols may hold packets before transmission. This is a fundamental difference to previous experiments conducted in wired networks, where only multiplexing random cross-traffic causes randomness, and where the schedulers are work-conserving. Almost all available bandwidth estimation tools assume work-conserving scheduling. Few exceptions are tools designed for Wifi networks, see Sec. 3.1. By these experiments, we demonstrate that our system identification procedure can deal with work-conserving as well as non-work-conserving behavior.

The evaluations for the MAC protocol and TCP show that our system identification procedure enables the determination of system models where modeling is



(a) max-plus (b) min-plus (c) available bandwidth

Figure 8.1: Representation of service curves in the max-plus algebra, in the min-plus algebra, and as available bandwidth.

typically challenging due to complex protocol characteristics and channel conditions.

All evaluations, except the simulations in Sec. 8.5, are conducted by the use of real hardware equipment. If not stated otherwise, we use the Emulab testbed [130] or our local testbed using the Emulab software [1] for the experiments.

We use three types of representations for the service curves: The max-plus representation $T^{\varepsilon}_{\mathcal{S}}(n)$, which directly follows from Thm. 5.1. It plots the service curve by time against data. The min-plus representation $\mathcal{S}^{\varepsilon}(t)$ is presented by using the lower bound from Thm. 5.4, which leads to a small imprecision of at most one packet. It displays data against time, which is usually more intuitive since a higher value also indicates a greater service. Furthermore, we define the ε-effective available bandwidth as $\alpha^{\varepsilon}(t) = \mathcal{S}^{\varepsilon}(t)/t$. This representation as a rate function yields a probabilistic lower bound on the service process $\mathsf{P}\left[\frac{S(\tau,t)}{(t-\tau)} \geq \alpha^{\varepsilon}(t-\tau), \forall \tau \geq 0\right] \geq 1-\varepsilon$, i.e., the probability that an average rate equal to or greater than $\alpha^{\varepsilon}(t)$ is achievable for the duration of $t-\tau$ is equal to or greater than $(1-\varepsilon)$.

Fig. 8.1 demonstrates the representations for a devised service curve consisting of two segments with the rates 100 pps and 200 pps and the delay quantiles 10 ms and 25 ms, respectively. Note that in Fig. 8.1a the label of the x-axis is the packet index $n \geq 0$, where $n = 0$ specifies already the first packet, whereas in Fig. 8.1b the data (in number of packets) are presented on the y-axis. For example, in Fig. 8.1a the first packet with $n = 0$ exhibits a delay of 10 ms. In Fig. 8.1b the first packet ($data = 1$) experiences a delay of 20 ms due to the imprecision of one packet introduced by the inversion of the service curve from the max-plus to the min-plus algebra. In this example, the inversion has an obvious impact on the curve. As transmission rates increase, the impact vanishes due to much shorter packet transmission times. The representation $\alpha^{\varepsilon}(t)$ is illustrated in Fig. 8.1c, which intuitively shows the impact on the rate. Although the service curves bend to the long-term available bandwidth at about 40 ms, the impact on the ε-effective available bandwidth continues for a much longer period. In this example, the available bandwidth still differs from the long-term available bandwidth after 300 ms. The rippled shape of the curve follows from the limited resolution of the service curve estimate. This resolution results from the finite number of probing rates, of which each contributes a linear segment to the curve.

The work in Sec. 8.1, Sec. 8.2, Sec. 8.3.1 and Sec. 8.4 is based on a cooperation with Markus Fidler and Jörg Liebeherr and partially available in [88, 89, 90]. The other sections in this chapter are joint work with Markus Fidler.

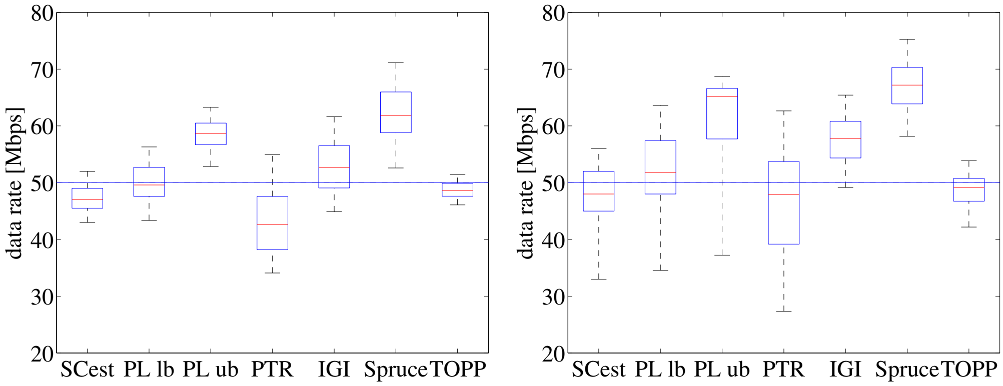
8.1 COMPARISON TO AVAILABLE BANDWIDTH ESTIMATION

For the comparison of our procedure from Sec. 7.1 to well known available bandwidth estimation tools, we customize the system description to a single value, the available bandwidth. Sec. 7.1.2.3 depicts in detail how our procedure can be used to estimate the available bandwidth. In summary, the estimate is given by the greatest probing rate for which a steady state delay distribution is assumed. We present results for single-hop as well as multi-hop network paths and also recover the analytical findings from Sec. 6.

We reuse the network topology illustrated in Fig. 7.2 and the cross-traffic characteristics specified in Sec. 7.1. Summarized shortly, the random cross traffic emits packets at constant rate with packet sizes following a truncated Exponential or truncated Pareto distribution. Each cross-traffic flow shares exactly one link only with the probe traffic. The bottleneck capacity is 100 Mbps that is utilized by cross traffic with a mean rate of 50 Mbps, whereof a long-term available bandwidth of $\alpha^{\infty} = 50$ Mbps follows. To be consistent with existing experimental evaluations of available bandwidth estimation tools, FIFO scheduling[12] is used.

---

[12] Here, we anticipate results from Sec. 8.3.2, where effects of various scheduling disciplines are analyzed. In short, FIFO scheduling is non-linear for short time periods. In Sec. 8.3.2, we show the applicability of the system identification procedure to networks with FIFO scheduling.



(a) Exponential cross traffic (b) Pareto cross traffic

Figure 8.2: Available bandwidth estimates from various tools. Each box shows the median, 0.25 and 0.75 percentiles, and 0.05 and 0.95 percentiles from 100 trials. Our procedure SCest provides estimates that correspond well to the long-term available bandwidth of 50 Mbps for Exponential as well as Pareto cross traffic. It also performs comparable to the established available bandwidth estimation tools Pathload and TOPP.

8.1.1 *Single-hop Network*

We evaluate our system identification procedure by comparing it to well-known available bandwidth estimation tools as Pathload, IGI/PTR, Spruce, and TOPP (implemented as dietTOPP [63]) in the following. All available bandwidth estimation tools return the estimate as a single value, except the tool Pathload. It expresses the available bandwidth as a range to account for variability of the available bandwidth. We set the buffer size at the bottleneck link to 200 packets, which causes moderate packet loss in this scenario. Since our probing procedure is similar to the one of Pathload, we choose a similar configuration with $r_{acc} = 1$ Mbps, a train length of $N_1 = N_{max} = 100$ packets and $I = 11$ iterations. Deviantly, Pathload uses 12 iterations; we choose here an odd number to achieve a clear majority decision as described in Sec. 7.1.2.3. For all other tools, we use their default parameters. We repeat the measurement for each tool 100 times.

Fig. 8.2 presents the results as box-plots, each box shows the median, the 0.25 and 0.75 percentiles, and the 0.05 and 0.95 percentiles for 100 repetitions of each tool. For Pathload the upper bound and lower bound of the range are labeled as *PL lb* and *PL ub*, respectively. We denote our service curve estimation tool as *SCest*. As reference, the long-term available bandwidth is added as a horizontal line.

The performance of our procedure is comparable to Pathload and TOPP, which track the available bandwidth well. In case of the bursty Pareto cross traffic, our procedure performs better than all tools except TOPP, which provides comparable results. The amount of probing traffic is comparable to the available bandwidth estimation tool Pathload, which uses packet trains of length 100 and 12 iterations [59]. In addition to the representation of the estimate as one value, our system identification procedure returns a system model, namely a service curve, that considers the available bandwidth on arbitrary scales.

8.1.2 *Multi-hop Network*

In this section, we present available bandwidth estimation results for a network path with multiple bottleneck links. In such networks a significant underestimation of the long-term available bandwidth is reported e.g., in [60]. We explain this underestimation analytically in Sec. 6.2, in which we prove that the service process of such systems given by Eq. (2.11) is less than or equal to the available bandwidth defined in Eq. (3.2). Equality only holds in the long-term as the measurement duration approaches infinity. By varying the train length, we examine these analytical findings in this section by comparing estimation results of our probing procedure from Sec. 7.1 and the tool Pathload, which is often consulted as a benchmark.

We set up a network topology according to Fig. 7.2 with one, three, and five bottleneck links. We use large buffers of 10^6 packets to avoid additional effects due to packet loss since we are only interested in the observation of effects of the multiple bottleneck links. We restrict the cross traffic to Exponential cross traffic and use the tool Pathload to confirm the findings described in Sec. 6.2.

We estimate the available bandwidth for train lengths ranging from 100 packets to 1600 packets. For the tool Pathload, we report the upper bound and lower bound, and for the estimate from our tool SCest, we show the long-term available bandwidth for $r_{acc} = 1$ Mbps. We slightly modify Pathload to support a configurable train length, by default Pathload sends only packet trains of 100 packets. We perform

Table 8.1: Available bandwidth estimates for a multi-hop topology with multiple bottleneck links.

train length [packets]	available bandwidth estimate [Mbps]								
	SCest			Pathload					
				lower bound			upper bound		
	bottleneck links			bottleneck links			bottleneck links		
	1	3	5	1	3	5	1	3	5
100	48	43	41	50	43	38	58	53	44
200	48	45	44	51	45	41	54	52	45
400	48	47	46	51	47	43	53	51	45
800	48	47	47	51	49	45	52	51	46
1600	48	48	48	51	49	46	51	51	47

100 runs for each train length and compute the median of the results, which are presented in Tab. 8.1.

The results show that for a single bottleneck link both approaches track the long-term available bandwidth well by using short and long trains. However, increasing the number of bottlenecks and using short trains, leads to a systematic underestimation of the long-term available bandwidth as expected. The long-term available bandwidth is recovered if the train length is increased, as also predicted in Lem. 6.3. This is confirmed by our procedure as well as Pathload and shows the need for a train length adaptation as we propose in Sec. 7.1.2.2 and Sec. 7.2.2 to observe the long-term available bandwidth.

For the previous comparative evaluations, we estimate the long-term available bandwidth. However, our system identification is able to estimate the service on arbitrary scales, which we demonstrate in the following sections.

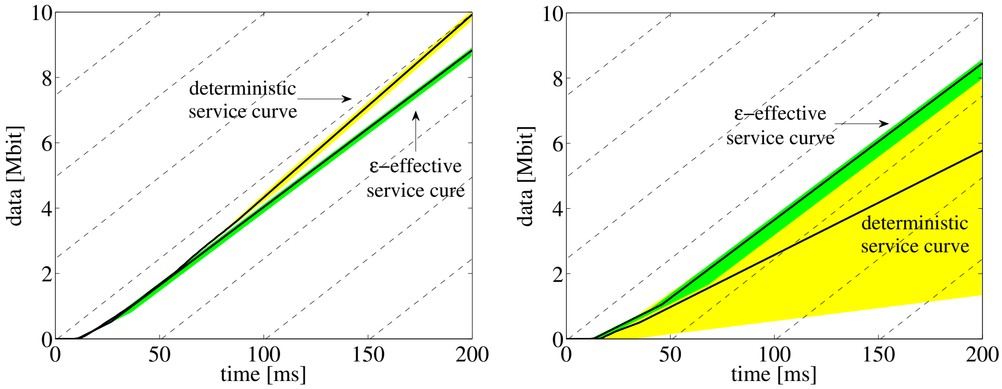
8.2 comparison of service curve estimation procedures

Another system identification approach developed in the network calculus is given in [80]. It shares many similarities with our approach since it also relies on a black-box model and uses a similar probing procedure for the identification of the system. The difference is the assumed system class in [80]. It assumes a deterministic time-invariant linear system, whereas we rely on a time-variant system that accounts for randomness. We provide a demonstration of the impact of the assumed system by the comparison of these two approaches.

For the comparison, we conduct an experiment with a single-hop topology with FIFO scheduling and large buffers (of 10^6 packets) according to Fig. 7.2. The approach in [80] is based on constant rate packet train probes with fixed train length and with increasing probing rates for the single trains. For each probing rate multiple iterations are performed. Each iteration yields one service curve, the average of these curves gives the final estimate. We set the fixed train length to 800 packets (similar to [80]), for each rate we perform 200 iterations, and the increment of sequent rates is set to 8 Mbps. Using the probing procedure from Sec. 7.1, these parameters yield a maximum train length of $N_{max} = 800$ packets with an initial length of $N_1 = 100$ packets, $r_{acc} = 8$ Mbps, and $I = 200$ iterations. Further, we set $\xi = 0.05$. We perform the comparison in the min-plus algebra as it is used in [80]. We therefore perform the estimation in the max-plus algebra and transform it into the min-plus algebra as specified in Sec. 5.3.3.

In detail, we compare the ε-effective service curve estimate to the average deterministic service curve estimate from [80]. Moreover, we show confidence intervals for both approaches to compare the reliability of the estimates. For the ε-effective service curve estimate, the confidence intervals follow from the confidence intervals of the quantiles, see Sec. 7.1.2.4. The confidence intervals of the deterministic approach are computed for the mean values of the individual samples. We use a confidence level of 0.95.

The results are presented for Exponential cross traffic and Pareto cross traffic in Fig. 8.3. The ε-effective service curve and the deterministic service curves are plotted as black lines, the range between the upper and lower confidence intervals is shown as a filled area, and reference lines with a slope of 50 Mbps are added as dashed lines. In case of the friendly Exponential cross traffic both approaches perform similar; the deterministic approach slightly overestimates the service since the slope increases above 50 Mbps. Furthermore, the confidence intervals show a reliable estimate. In the case of the bursty Pareto cross traffic our new procedure still delivers an accurate estimate of the service since it discovers the long-term available



(a) Exponential cross traffic (b) Pareto cross traffic

Figure 8.3: Comparison of the ε-effective service curve estimates to deterministic service curve estimates. For the friendly Exponential traffic both estimates show similar results, but for the bursty Pareto traffic only the ε-effective service curve estimate matches the long-term available bandwidth of 50 Mbps (indicated by dashed lines) well and shows a robust result. The estimates are indicated by black lines and the shaded areas specify the 0.95 confidence intervals.

bandwidth closely. In addition, the confidence intervals are close to the estimate. The deterministic approach underestimates the service significantly with confidence intervals that indicate an unreliable estimate. For the approach from [80], these results are expected since it was designed for deterministic networks. The results show exemplary for networks with small variability that deterministic approaches can still be applied with reasonable accuracy. However, in networks where the assumption of small variability does not apply, deterministic approaches obviously fail. Moreover, deterministic approaches do not contain any information on the probability of the service. On the contrary, the ε-effective service curve inherits additionally a violation probability that accounts for it.

Next to this comparative evaluations, we present ε-effective service curves and the representation as ε-effective available bandwidth of various networks with random service obtained by system identification.

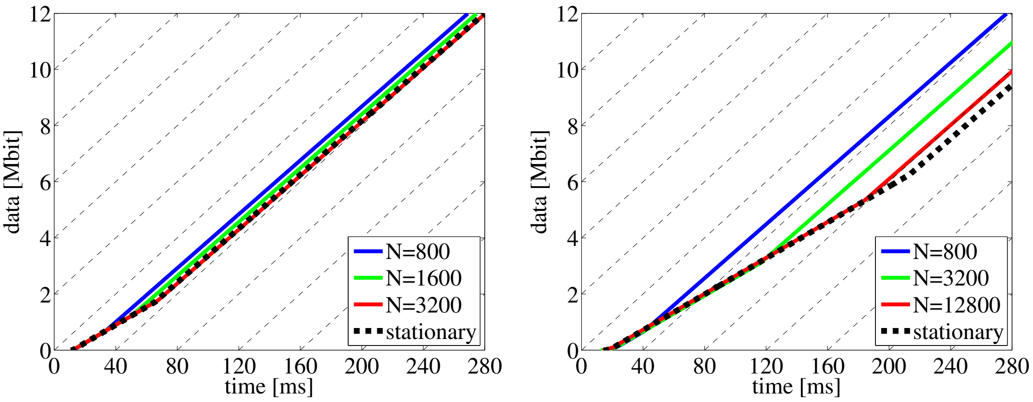
8.3 WIRED NETWORKS WITH RANDOM CROSS TRAFFIC

In the following, we present service curve estimates for wired networks using the network topology presented in Fig. 7.2. If not stated otherwise, a single-hop topology employing priority scheduling with priority for the cross-traffic at the bottleneck link is used. We present estimates for limited and unlimited scales using the procedure described in Sec. 7.1, where also the estimation of limited scales is illustrated. For the limited scale, the train length N is set to a maximal value N_{max}. We further present effects of various scheduling disciplines and packet loss due to small buffers at the bottleneck link. Thereby, we also demonstrate the difference between a service curve estimate and the available bandwidth. Moreover, we present estimates for networks with multiple bottleneck links and analyze the intrusiveness of our procedure.

We use random cross traffic following a truncated Exponential distribution or a truncated Pareto distribution as specified in Sec. 7.1 with a mean rate of 50 Mbps, resulting in a long-term available bandwidth of $\alpha^{\infty} = 50$ Mbps. As in Sec. 8.1.2, for multi-hop networks, we only consider the Exponentially distributed cross traffic. The configured probing parameters are $r_{acc} = 4$ Mbps, $I = 250$, $N_1 = 100$ packets, and $\xi = 0.05$ for the experiments in this section. The results are presented by a service curve in the min-plus algebra due to the more intuitive representation. Therefore, we use the lower bound from the inversion defined in Thm. 5.4.

8.3.1 *Unlimited and Limited Scales*

In Sec. 8.1, we show that an estimation of the long-term available bandwidth is feasible with a small amount of probing traffic using a heuristic based on our trend test and a majority decision, which is explained in Sec. 7.1.2.3. Applying the heuristic allows a robust estimation without estimating the steady state delay distribution. As described in the context of Thm. 5.1 and derived in Eq. (5.8), the use of the delay estimates measured during the transient state, allows the construction of a service curve for a limited scale. Such a service curve estimate $\widetilde{T}^{\varepsilon}_{S}(n)$ is obtained by using a maximum train length N_{max} that does not observe the steady state delay distribution. The estimate holds then for the limited interval $[0, N_{max} - 1]$. Using an interval of interest by choosing N adequate, we obtain a service curve for that interval. Such intervals were introduced in the min-plus algebra as time scale bound in [25].



(a) Exponential cross traffic (b) Pareto cross traffic

Figure 8.4: Comparison of service curve estimates for unlimited and limited scales. All estimates converge to the long-term available bandwidth indicated by dashed lines. For Exponential traffic the estimates can hardly be distinguished, but for the Pareto traffic estimates obtained from longer trains result in less service due to the capability to observe larger delays.

Fig. 8.4 presents service curve estimates for limited scales with maximum train lengths in the interval $N_{max} = [800, 12800]$ and service curve estimates with train lengths that are increased until stationary delays are observed. Dashed reference lines are added to the figure indicating the analytical long-term available bandwidth of 50 Mbps by their slope.

The results show that all service curve estimates closely recover the long-term available bandwidth of 50 Mbps. The service curves differ when the segments bend and converge to their maximum rate. Evidently, if longer packet trains are used, service curve estimates bend further from the origin, and due to their longer duration they are able to observe larger delays. For the friendly Exponential traffic, the effect is marginal as Fig. 8.4a indicates, and all estimates are close to the estimate using delay quantiles from the steady state delay distribution. For the more bursty traffic, the service curve estimates with restricted train lengths still show a deviation for large packet trains of 12800 packets.

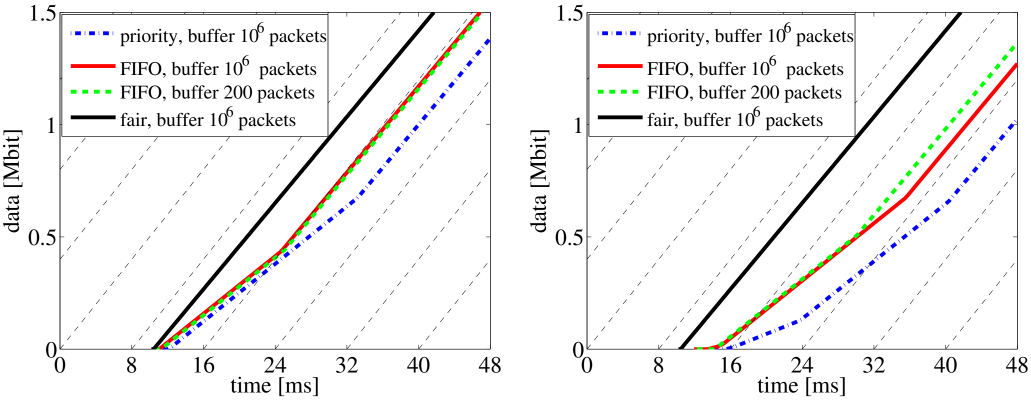
In general, long packet trains may be required to estimate service curves from the steady state. For this reason, a train length adaptation is needed to increase the train length to observe it. If the scale of interest is known in terms of packets or as a time scale bound, the maximum train length can be reduced notably. For example, using a train length of 12800 packets already covers a time scale of 3.2 seconds at the maximal probing rate of 48 Mbps. Such a time scale already exceeds the range of interest for many real time applications since delay requirements are often stricter.

8.3.2 *Scheduling, Limited Buffers and Multiple Bottleneck Links*

In real networks various scheduling disciples are deployed, schedulers use buffers with limited size, and network paths typically consists of multiple schedulers in series. Multiple bottlenecks may occur at these schedulers since the path is shared with random cross traffic. We relax some of the assumptions made in the previous sections and present results for the scenarios listed before. We also illustrate the representation of the service curve as an available bandwidth.

SCHEDULING AND LIMITED BUFFERS We evaluate the service for various scheduling disciplines and small buffer sizes that cause minor packet loss. The scheduling discipline decides about system properties as linearity, which is described in Chap. 2. The assumption of a max-plus linear system made in Thm. 5.1 is not always met. For example, the commonly used FIFO scheduler becomes non-linear if the total incoming traffic rate exceeds the maximal capacity of the scheduler. If not overloaded, the scheduler acts as a linear system [80]. The rate selection of our procedure ensures that on average the incoming traffic to the system, which consists of probe traffic and cross traffic, does not saturate the system. However, short-term violations of the linearity assumption occur since the traffic is packetized and the cross traffic is also bursty.

In the preceding experiments and simulations, packet loss is avoided by using sufficiently large buffers. However, the usage of the max-plus algebra enables the consideration of loss. We model packet loss as an infinite delay, i.e., we set the delay of packet n to $T_D(n) = \infty$ if it is lost. Hence, if the ratio of delay samples (i.e. the last packet of a train) that indicates loss, is equal to or greater than ξ, the probing rate r yields a delay quantile of $\widetilde{W}^{\xi}(r) = \infty$. Hence, the probing rate r does not contribute to the service curve estimate due to the minimum in Eq. (5.3).



(a) Exponential cross traffic (b) Pareto cross traffic

Figure 8.5: Comparison of service curve estimates for the scheduling disciplines: priority, fair and FIFO with large and small buffers. All estimates converge to the long-term available bandwidth of 50 Mbps. Fair scheduling provides the best service since up to the long-term available bandwidth the service is instantaneously available. Priority scheduling gives the worst service since cross-traffic is scheduled first. The service of FIFO scheduling is in between because probe and cross traffic is scheduled in order of the arrivals, it further improves for small buffers since large bursts are pruned.

The network setup is as depicted in Fig. 7.2 with one bottleneck link, whereby we estimate the service for various scheduling disciplines and buffer sizes. We set the maximum train length to $N_{max} = 800$. Fig. 8.5 comprises results for these systems with FIFO scheduling and large buffers of 10^6 packets as well as small buffers with 200 packets. The experimental results also compare these service curve estimates to estimates of systems with fair scheduling or priority scheduling with priority for the cross traffic; both scheduling disciplines are max-plus linear. All service curve estimates indicate a delay of about 10 ms that conforms with the propagation delay in the network topology, see Fig. 7.2. We recover as long-term available bandwidth 48 Mbps, i.e., also under packet loss the long-term available bandwidth is accurately detected. In this experiment, the packet loss is below 1% for all probing rates in the setup with small buffers.

Different scheduling disciplines result in different shapes of the curves. Fair scheduling guarantees the fair proportion for the probe traffic and the cross traffic regardless of the kind of cross traffic. After the propagation delay of about 10 ms a straight line with a slope reciprocal to the maximal probing rate follows. For the other scheduling disciplines, segments with lower probing rates contribute to the service curve estimate. The service curve estimate of the priority scheduler converges to the long-term available bandwidth last. The curve of the FIFO scheduler is between the curves of the fair scheduler and the priority scheduler. The service for FIFO scheduling is greater as for priority scheduling since FIFO scheduling forwards traffic in the order of arrivals, whereas for the priority scheduler the priority is given to the cross traffic.

Moreover, the impact of the different types of cross traffic is clearly visible for priority scheduling and FIFO scheduling. For these disciplines, the service curve converges later to the long-term available bandwidth for the more bursty Pareto traffic since large bursts induce additional delay at the queues of the schedulers. The fair scheduler guarantees a fair share, thereby, the burstiness of the cross traffic does not impact the service of the through flow. For the Exponential traffic, a difference between small and large buffers is hardly visible, on the contrary for the Pareto traffic, the service is greater for FIFO scheduling with small buffers. This is due to the burstiness of the Pareto traffic at which large burst are cut off by the limited buffer size resulting in more leftover service.

REPRESENTATION OF THE SERVICE CURVE AS AVAILABLE BANDWIDTH As we noted, the available bandwidth expressed by a single value or a range conceals information about the time scale on which the bandwidth is available, and whether the time scale is sufficient to observe the long-term available bandwidth. A service curve includes information on the available bandwidth as well as on the time scale of its availability. Next, we highlight this by expressing the ε-effective service curve estimate $\widetilde{\mathcal{S}}^{\varepsilon}(t)$ as an estimate of the ε-effective available bandwidth $\widetilde{\alpha}^{\varepsilon}(t) = \widetilde{\mathcal{S}}^{\varepsilon}(t)/t$. In Fig. 8.6, we show the estimation result for Exponential cross traffic and compare it to an estimate obtained by the tool Pathload. In the previous comparison presented in Sec. 8.1, we did not consider the information on the scale of availability of the service since we constrained the estimation to the long-term available bandwidth. As depicted in Sec. 7.1.2.3, for the estimation of service curves and the available bandwidth estimation the probing traffic cannot be reduced in the same extent as for available bandwidth estimation. In particular, the number of iterations has to be sufficient large to compute delay quantiles.

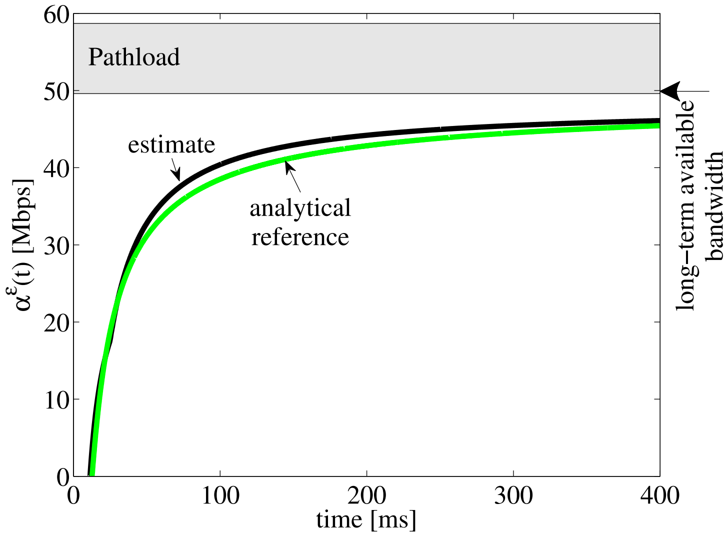


Figure 8.6: Representation of the ε-effective service curve as ε-effective available bandwidth $\alpha^{\varepsilon}(t)$ and comparison to estimates obtained with the tool Pathload. Also an analytical reference is added for this experiment that is matched well by our estimate.

We use the same network topology as before with FIFO scheduling and set the probing parameters $N_{max} = 800$ packets and $I = 250$ iterations.

Furthermore, we also present a validation of our system identification procedure by comparing the estimate to an analytical reference for the Exponential cross traffic. The analytical ε-effective service curve follows from a leftover service curve [25] by using a sample path bound for the cross traffic. This sample path bound on the cross-traffic follows from the Erlang distribution and the application of the union bound. The Erlang distribution arises from the sum of exponentially distributed random variables and yields a point-wise bound. At last, the sample path bound is obtained by using the union bound. Fig. 8.6 shows the results: The lower bound of the tool Pathload tracks the long-term available bandwidth well, but it does not include information about the time scale of service availability. The estimated ε-effective service curve includes this information and shows that on time scales up to 300 ms the available bandwidth $\alpha^{\varepsilon}(t)$ is significantly smaller. The comparison to the analytical reference shows that the estimate matches the analytical reference well.

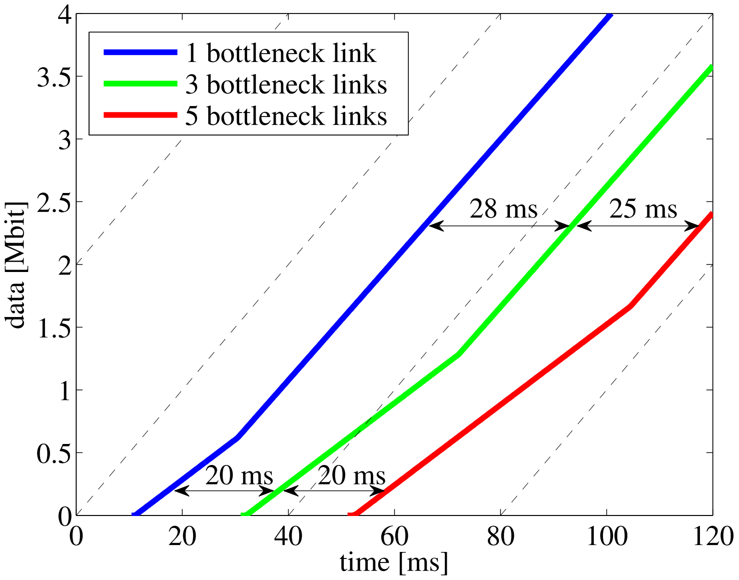
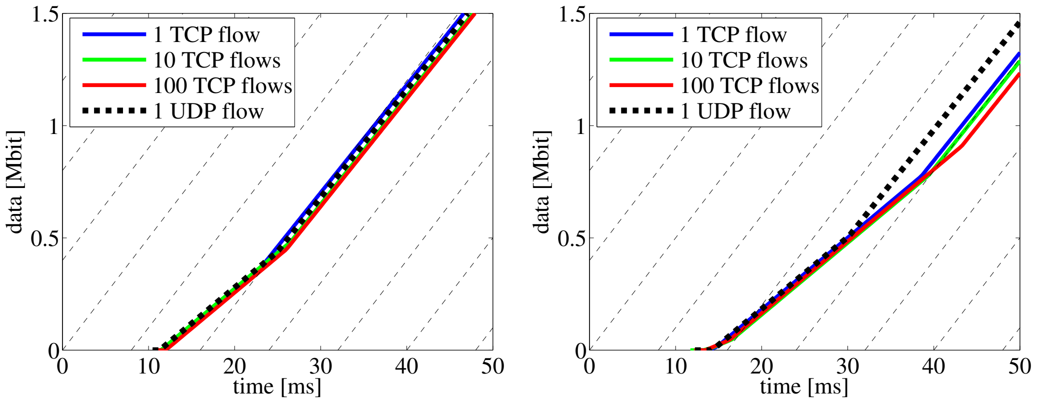


Figure 8.7: Service curve estimates for network topologies for one to five bottleneck links. Close to the abscissa the propagation delay is recovered. Increasing the number of bottleneck links induces further queueing delay indicated by the estimates, which converge later to the long-term rate.

MULTI-HOP NETWORK In the experiments before, we consider a single bottleneck. In multi-hop networks with random service, the bottleneck can alternate between the links, i.e., the link that limits the service may vary, and also multiple bottlenecks may occur simultaneously on a network path.

We present the service curve estimate for one, three, and five bottleneck links with large buffers of 10^6 packets featuring identical characteristics in Fig. 8.7. We set the maximum train length to $N_{max} = 1600$; Tab. 8.1 presents that this limit is sufficient to track the long-term available bandwidth closely. Each service curve indicates the propagation delay of the network path, which is 10 ms for one bottleneck link, 30 ms for three bottleneck links, and 50 ms for five bottleneck links. The propagation delay is visible at the first increase of the service curves close to the abscissa. The first increase of all service curve estimates show that on short time scales the service is less than the long-term available bandwidth of 50 Mbps. This is indicated by the slope of the service curves, that is less than the slope of the dashed reference lines, which have a slope corresponding to a rate of 50 Mbps.

In the long-term, all estimates converge to the long-term rate. Note that, the horizontal distances between the estimates increase, which indicate that the queueing delay increases with multiple bottleneck links.



(a) Exponential cross traffic

(b) Pareto cross traffic

Figure 8.8: Service curve estimates for TCP and UDP cross traffic. Since all estimates recover the long-term available bandwidth and converge to a rate of about 50 Mbps, our procedure is non-intrusive. Otherwise, the probing would displace the TCP cross traffic resulting in rates above 50 Mbps.

8.3.3 *Evaluation of the Intrusiveness*

To evaluate the intrusiveness of our probing procedure, we replace the non-elastic UDP cross traffic used in all experiments before by elastic TCP traffic in this experiment. Since TCP integrates a congestion control algorithm, the constant rate UDP probe traffic could displace the TCP cross traffic if the probing procedure was intrusive. We use the congestion control algorithm TCP Cubic [51], which is the default algorithm in Linux today. We generate 1, 10, and 100 TCP cross traffic flows, which transmit in total an average rate of 50 Mbps. The network topology and the cross traffic distributions are the same as before with the difference that the cross traffic is transmitted by TCP. The scheduling is FIFO with large buffers (of 10^6 packets) and the maximum train length is $N_{max}=800$ packets.

Fig. 8.8 presents the service curve estimates. All service curves recover the long-term available bandwidth of 50 Mbps closely, which shows that our probing procedure is non-intrusive because it does not displace the elastic TCP cross traffic and does not return an estimate above 50 Mbps. Furthermore, the experiment shows the impact of the transport protocol TCP. For the Exponential traffic a difference is hardly visible, but for the bursty Pareto traffic the service is lowered by the use of TCP and reduces further with the number of flows. Since numerical delay values can hardly be read from Fig. 8.8, we further show the estimates of the delay

Table 8.2: Long-term available bandwidth estimates and delay quantile estimates for elastic cross traffic.

cross traffic flows	Exponential		Pareto	
	available bandwidth [Mbps]	latency [ms]	available bandwidth [Mbps]	latency [ms]
1 UDP	48	16	48	19
1 TCP	48	15	48	22
10 TCP	48	16	48	23
100 TCP	48	16	48	24

quantiles for a probing rate of 48 Mbps in Tab. 8.2. We restrict the presentation in the table to that rate since it exhibits the strongest effect on the delay quantile. The delay increases from 19 ms for 1 UDP flow to 24 ms for 100 TCP flows. The figure and the table show that TCP intensifies the effect of the greater delay quantile for bursty traffic because the delay quantiles further increase with the number of TCP flows.

8.4 IEEE 802.11A WIFI NETWORKS

In wired networks, the source of randomness of the service is usually the multiplexing of random traffic. Wireless networks exhibit additional sources of randomness. IEEE 802.11a Wifi networks have a complex MAC protocol, whereby the assumption of a work-conserving link with constant capacity does not hold. These networks use carrier sense multiple access and collision avoidance (CSMA/CA) for medium access. In the context of IEEE 802.11a networks CSMA/CA is implemented by the DCF. Expressed in simplified terms, before a station accesses the medium, it senses the medium and only proceeds if the channel is free. If it is free, it waits a random time and then sends the packet. An acknowledgement packet confirms the successful transmission. If the acknowledgement is not received, the range of the random waiting time is increased and the packet is retransmitted using the same procedure. This channel access procedure achieves fairness between the stations in the long-term [21]. For further details on the protocol specification,

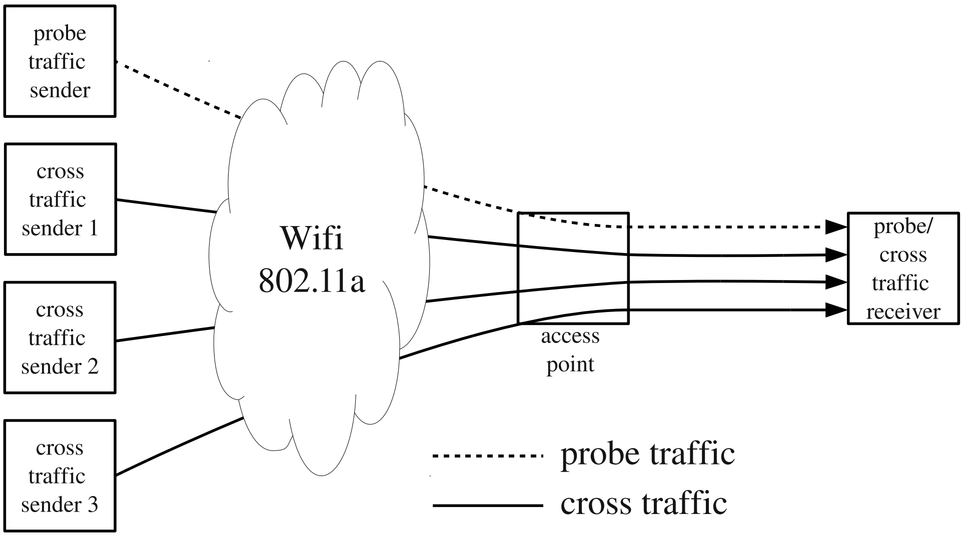


Figure 8.9: IEEE 802.11a network topology with one probe traffic sender and up to three cross traffic senders that share a wireless network, which is connected via a 100 Mbps wired link to a receiver.

see [5]. Due to this channel access procedure, the service that a station receives is random (because of random waiting times and failed transmissions) and non-work-conserving scheduling (because of waiting times even if the channel is idle). Further randomness occurs from the variability of the wireless channel caused by interference and fading. We minimize this source of randomness in this experiment by using a channel, at which no other IEEE 802.11 network is operating. We use a free channel since we are interested in the influence of the DCF on the service on short as well as on long time scales.

Fig. 8.9 shows the experimental setup arranged in a local testbed. The wireless network is shared by one probe traffic sender and up to three cross traffic senders. All flows are transmitted via an access point to a receiver. The connection between the access point and the receiver has a rate of 100 Mbps. To estimate the impact of the DCF on the service, we use constant rate cross traffic, which avoids additional randomness. If a packet size of 1500 Byte is used, the maximal throughput of a IEEE 802.11a network is about 30 Mbps, see e.g., [20, 66]. We therefore generate cross traffic with an aggregate throughput of 25 Mbps. This rate is equally divided between the cross traffic senders, which ensures that each station uses its fair share.

Since the DCF achieves fairness in the long-term [21], we compare the results to a wired experimental setup, in which the access point is replaced by a router. Thereby, a fair scheduling discipline is deployed and the wireless channels are exchanged

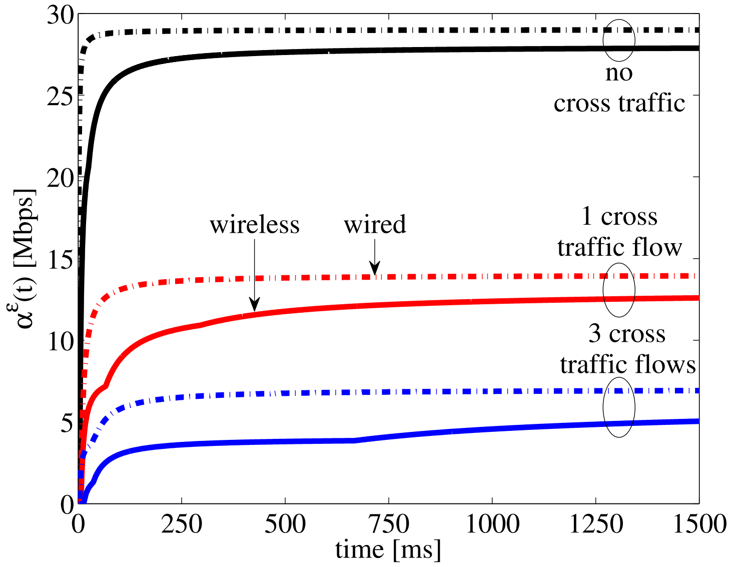


Figure 8.10: Available bandwidth $\alpha^{\varepsilon}(t)$ for an IEEE 802.11a wireless network compared to the available bandwidth of a wired network with fair scheduling for up to three cross traffic senders. The wireless network shows a much slower convergence and also the long-term rate stays below the one of the wired network. The smaller service in the wireless network results from the random channel access, possible interference on the channel, and retransmissions induced by the MAC protocol.

by wired links. In the wired network, it can be assumed that the channel access is deterministic and that no packet collisions occur on the link. This allows us to compare the service curve estimates of a work-conserving fair scheduler to the estimates of the non-work-conserving long-term fair channel access in IEEE 802.11a networks. The probing procedure described in Sec. 7.1 is used with a maximal train length of $N_{max} = 800$ packets, $I = 250$ iterations, $\xi = 0.05$, and $r_{acc} = 1$ Mbps. As an intuitive form of presentation, we show the estimates as a ε-effective available bandwidth $\widetilde{\alpha}^{\varepsilon}(t) = \widetilde{S}^{\varepsilon}(t)/t$ in Fig. 8.10. The solid lines show the estimates of the wireless network and the dash-dotted lines the estimates of the wired network.

For the wired experimental setup with a fair scheduling discipline, all curves converge fast to their theoretical rate, which is 30 Mbps without cross traffic, 15 Mbps for one cross traffic flow, and 7.5 Mbps for three cross traffic flows. The rates follow from the fair share between the total number of stations. All estimates converge to their expected rate that is in the range of the analytical rate minus the accuracy defined by r_{acc}.

In comparison to the standard fair scheduling discipline, the ε-effective available bandwidth of the IEEE 802.11a network converges slower to its long-term rate.

This results from the additional variability introduced by random waiting times at the medium access, collisions, interference, and retransmissions of packets. This also lowers the long-term available bandwidth if compared to the wired network topology.

8.5 WINDOW FLOW CONTROL PROTOCOLS

In this section, we evaluate the probing procedure from Sec. 7.2 by two simulations of flow control protocols. First, we show the applicability to a window flow control protocol for which analytical results exist in the network calculus. Second, we present results for a congestion control protocol. For the following simulations, the discrete event simulator OMNeT++ [4] is used. To estimate the service curve the probing procedure from Sec. 7.2 is simplified by using fixed train lengths. We present the estimates in the max-plus algebra since analytical results are also available in this algebra for the first simulation. This avoids the application of the inversion to the min-plus algebra, which would yield an imprecision of one packet.

8.5.1 *Window Flow Control with Fixed Window Size*

Window flow control regulates the transmission rate based on packets in transit on a network path. The flow from the sender to the receiver is controlled by a window flow control element, where the window size w determines the maximum number of packets in transit on this path. The element acts as follow: if less than w packets are in transit, new packets may enter the network; else new packets are buffered at the flow control element and enter the network when packets depart from the network. The departure of a packet from the network is signaled by an acknowledgement packet sent from the receiver to the flow control element. In this simulation, the network consists of a sender and a receiver connected by a link with unlimited capacity. The packets as well as the acknowledgements experience the same delay d. No packet loss and no further delays occur due to processing or

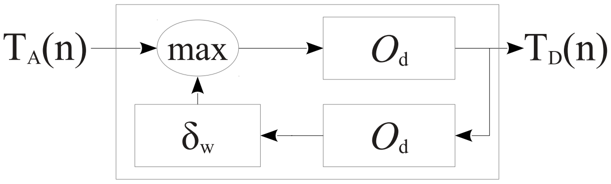


Figure 8.11: System model for window flow control with delay d on the forward and reverse path and a window size of w.

transmission. This example shows a deterministic behavior because of the constant window size and constant network path properties.

Fig. 8.11 illustrates the system model for which analytical results exist e.g., in [8, 29]. To compare our estimates to the analytical results, we use the service curves from [29, Lem. 6.3.5] and [29, Sec. 6.3.4], which are given in the max-plus algebra. Therein, the service curve of the delay element O_d is defined by $O_d(n) = d$, i.e., each packet experiences a delay of d. The service curve of the window element $\delta_w(n)$ is given by:

$$\delta_w(n) = \begin{cases} -\infty & n < w \\ 0 & n \geq w \end{cases} .$$

From this definition, it follows that a packet is instantaneously forwarded without any delay if $n \geq w$, else it experiences a delay of $-\infty$. The service curve of the network is

$$\begin{aligned} T_S(n) &= \max\{O_d(n); O_d \otimes O_d \otimes O_d \otimes \delta_w(n); O_{5d} \otimes \delta_{2w}(n)\ldots\} \\ &= (O_d \otimes O_d \otimes \delta_w)^* \otimes O_d(n) \end{aligned} , \tag{8.1}$$

The structure of the maximum of the right term in the first line of Eq. (8.1) is given by the feedback loop shown in Fig. 8.11: for the first w packets, $T_S(n) = d$ due to the delay of the forward path, any further terms are masked by $-\infty$ since $n < w$; for $w \leq n < 2w$, $T_S(n) = 3d$ due to the repetitive structure of the loop; $T_S(n)$ increases further in n to $(2\lfloor n/w \rfloor + 1)d$. This repetitive structure is also expressed by the super-additive closure $(.)^*$ in the second line of Eq. (8.1), for details see [29]. The service curve becomes a staircase function, which is shown in Fig. 8.12.

We configure the parameters $w = 100$ packets and $d = 50$ ms that yield a long-term available bandwidth of $\alpha^\infty = \frac{w}{RTT} = 1000$ pps, where $RTT = 2d$ is the round trip time.

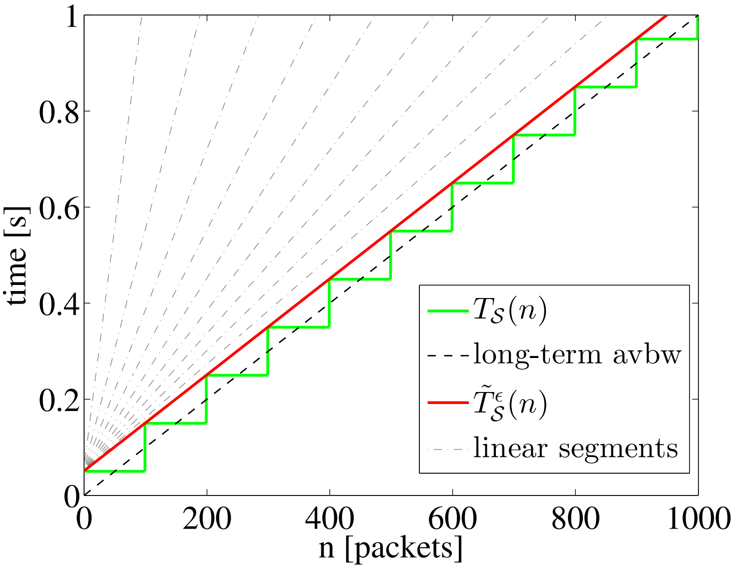


Figure 8.12: Analytical service curve, estimated service curve constructed of linear segments, and long-term available bandwidth (avbw) for window flow control. The estimated service curve $\widetilde{T}^{\varepsilon}_{\mathcal{S}}(n)$ gives an upper bound to the analytical one $T_{\mathcal{S}}(n)$. The estimated service curve recovers the long-term available bandwidth, but it also accounts for delays. The dash-dotted lines show the single service curve elements that see a stationary delay. Due to this deterministic example, only the segment with the highest rate of 1000 pps contributes to the final estimate.

If packets can enter the network instantaneously, they and the acknowledgements only experience the constant delay d.

For the estimation, we use a linear rate selection with $r_{acc} = 100$ pps and collect 250 delay samples of a sufficiently long train with a mean gap of $N^I_{\mathcal{S}}/I = 1000$ packets and no offset $N_{off} = 0$. For all probing rates less than or equal to the long-term available bandwidth, packets experience a delay of 50 ms, greater rates see an increasing delay. The constant delay for rates below the long-term available bandwidth follows from the deterministic behavior in this scenario, which results in a deterministic service curve (i.e. $\varepsilon = 0$). A similar estimation approach for deterministic networks is given in [80]. We include this example to demonstrate that our estimation procedure is applicable to protocols such as window flow control.

Fig. 8.12 shows the analytical service curve (by a green line) from Eq. (8.1) and the estimated service curve (by a red line). The result confirms that the estimate is an upper bound on the service. Since the service curve is constructed of linear segments, which are indicated by the dashed-dotted lines, the estimate cannot track the steps of the analytical curve. The delay up to a probing rate of α^{∞} is in this deterministic example 50 ms (therefore it is also stationary), the estimate follows from the linear segment with a y-axis intercept of 50 ms and a slope of 1/1000. This example illustrates the difference between a service curve and the long-term available bandwidth indicated as a dashed line. Although both curves have the same slope, the service curve contains delay information.

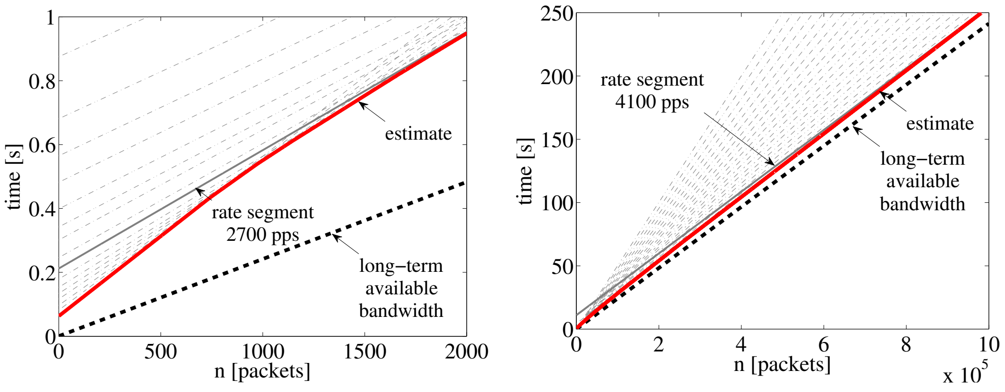
8.5.2 *Congestion Control with Congestion Notification*

The size of the window of the flow control protocol presented in the former section can be made variable, e.g., to implement a rate control that adapts dynamically to congestion experienced on a network path. Such an adaptive behavior results in a time-varying rate of the sender.

In the following, we evaluate the service by simulation of a congestion control protocol that reacts on congestion notifications indicated by marked acknowledgements. As in the simulation before, the outgoing traffic of the sender is regulated by a window, but now the window size is adapted by congestion notifications. We refer to this window as the congestion window. The sender halves the window on receiving a congestion notification. If the acknowledgement is not marked, it increases the window by one each RTT. The basic principle of this scenario resembles to TCP Reno congestion avoidance without packet loss, which is achievable by using explicit congestion notification (ECN) [46, 110] for marking. The long-term available bandwidth of such a network is given by the Mathis equation [93], which we use to validate our estimate. The Mathis equation is

$$\alpha^{\infty} = \frac{MSS}{RTT}\frac{K}{\sqrt{p}}, \tag{8.2}$$

where MSS is the maximum segment size, RTT the round trip time, p the packet loss probability, and K a constant, which depends on the kind of acknowledgements and the kind of congestion notification. By the use of acknowledgements for every packet and random notification, where packets are marked according to a Bernoulli distribution with probability p, the constant is $K = 1.31$, see [93].



(a) short-term estimate (b) long-term estimate

Figure 8.13: Service curve estimate for congestion control for a loss rate of $p = 10^{-3}$. Fig. (a) shows the construction of the service curve estimate by the single segments (dash-dotted lines) in the time scale of 1 s. The last rate that contributes to the estimate is 2700 pps, which is clearly below the long-term rate. On a large time scale the estimate converges to the long-term rate of 4100 pps, which is in the expected range of $[\alpha^{\infty} - r_{acc}, \alpha^{\infty})$ given by Eq. (8.2).

For the experimental setup, we choose following parameters: the network path has an unlimited capacity, a delay of 5 ms, and a marking probability of $p = 10^{-3}$ following a Bernoulli distribution. The long-term rate for this setup is 4143 pps, which follows from Eq. (8.2). We collect 15000 delay samples of sufficiently long trains with $N_{off} = 10^6$ packets and a mean distance between samples of $N_1^S / I = 10^6$ for probing rates from 2000 pps to 4200 pps with a linear increase of $r_{acc} = 100$ pps. For rates up to 4100 pps, stationary and independent samples are detected as expected due to the long-term rate of 4143 pps.

Fig. 8.13 presents the short-term and the long-term evolution of the service curve estimate with $\xi = 10^{-3}$. The curve converges in the long-term, which is here about 250 s, to the rate of 4100 pps, but in the short-term of about 1 s, the service is significantly lower. This first observation already shows the impact of congestion control on the service. Expected rates, e.g., as predicted by the Mathis equation, are achieved only in the long-term, which is in this example about 250 s. Real-time applications, which rely on such protocols, and which have delay requirements, must account for these effects.

In this simulation, assumptions apply that are made in [93]. However, in real networks effects occur that disturb these ideal conditions, e.g., packet loss features correlations or implementations impact the expected protocol behavior. In the following section, we present experimental results obtained in real networks that deploy the transport protocol TCP, whereof this last simulation is a simplification.

8.6 TRANSMISSION CONTROL PROTOCOL

TCP is the prevalent transport protocol in the Internet as identified in [92]. It was presented in 1974 in [26] and standardized in 1980 in [109]. To this day, it is an active area of research since it is continuously adapted to current networks e.g., to networks with high bandwidth delay products (BDP), which is the product of the round trip time (RTT) and the bandwidth. One fundamental extension was a congestion control algorithm [56], which became a major area of research and led to various algorithms, e.g., [9, 51]. Moreover, analytical models exist to determine the performance of TCP; famous models for the long-term throughput are presented in [93, 100]. However, the short-term behavior of TCP is also of interest since real-time applications use TCP for data transport. Well-known examples are Skype for audio and video telephony and the Adobe Flash platform for audio and video streaming, see e.g., [24, 92]. In [24], the short-term behavior is analyzed by modeling the delay of TCP connections. The results already show that the end-to-end delay using TCP can comprise several seconds. However, analytical models of TCP offer only a limited view on its behavior since the models typically use an idealized protocol behavior due to manifold extensions and implementations of TCP. These details are difficult to integrate in the analytical models. Our system identification procedure enables the estimation of the service that TCP provides. It is able to display the short-term as well as the long-term behavior of TCP and includes details specific to extensions and implementations. The applicability of our procedure is supported by [14], where it is shown that TCP is max-plus linear. We estimate the service at the application layer. This comprises the characteristics of TCP as well as of the network path and their interaction. In terms of operating systems, our estimation result is the end-to-end service available to an application between the sending socket and the receiving socket.

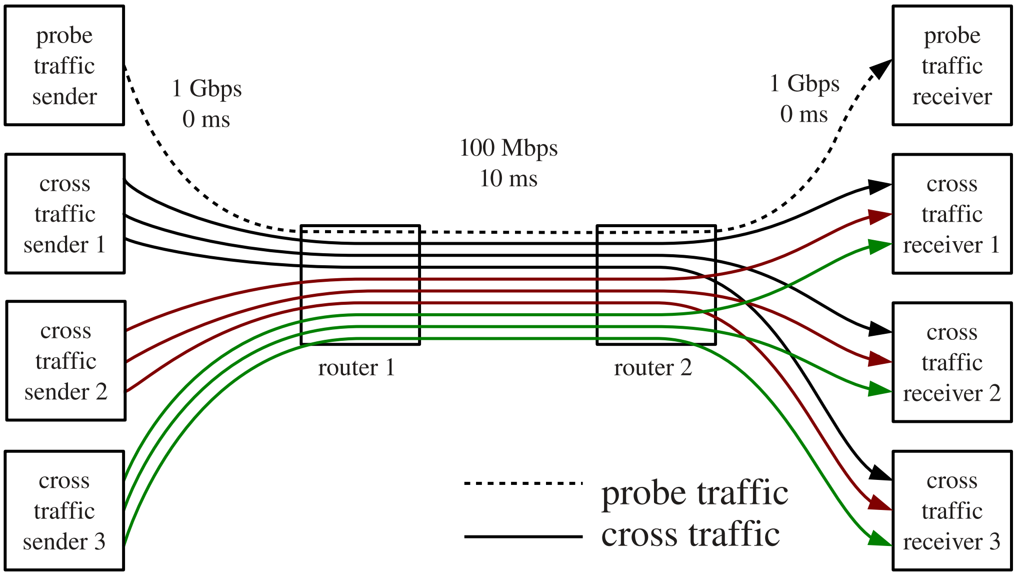


Figure 8.14: Dumbbell network topology with one probe traffic sender and nine cross traffic flows. The bottleneck is at router 1, where the link capacity is 100 Mbps with a mean delay of 10 ms between router 1 and router 2.

Table 8.3: One way delays of network paths.

delay [ms]	probe traffic receiver	cross traffic receiver 1	cross traffic receiver 2	cross traffic receiver 3
probe traffic sender	10	-	-	-
cross traffic sender 1	-	12	11	9
cross traffic sender 2	-	11	8	10
cross traffic sender 3	-	9	8	12

In the following, we present estimates of the ε-effective available bandwidth for essential configuration parameters such as the queue size at the intermediate routers, the congestion control algorithm at the end hosts, and the queueing discipline at the intermediate routers. We chose the representation as ε-effective available bandwidth since the comparison of the estimates is more illustrative by a rate function.

For the experiments, we use the topology shown in Fig. 8.14, which is adapted from [10]. The probe traffic and the cross traffic flows use TCP as transport protocol. Cross traffic is sent by three senders and received by three receivers using the software Iperf [3], which is a greedy application, i.e., it utilizes completely the service the TCP connection provides. The cross traffic senders establish nine flows from which each individual sender transmits one flow to each cross traffic receiver.

The propagation delays of the flows are uniformly distributed in the interval from eight to twelve milliseconds, see Tab. 8.3, to avoid synchronization between the flows. The mean delay is ten milliseconds. TCP aims at fairness among flows, which results in an expected long-term available bandwidth of 9 Mbps[13].

The probing procedure described in Sec. 7.2 is used since the steady state behavior is only observed for long time periods as already discovered by simulation of congestion control in Sec. 8.5.2. The probing parameters used throughout this section are $N_{off} = 10^6$ packets, $N_1^S / I = 2000$ packets, $I = 750$ iterations, $\xi = 0.05$, and $r_{acc} = 1$ Mbps. The IP packet size is 1500 bytes, whereof 20 bytes and 32 bytes are the header sizes of IP and TCP (including TCP timestamps in the option field), respectively. We use blocking TCP sockets, i.e., if the TCP socket is full, the application blocks at the socket until free buffer space is available. This avoids any packet loss at the TCP socket. Socket buffer sizes at the sender and receiver are chosen large enough, so that transmission rates of the TCP socket are only limited by the congestion control algorithm.

8.6.1 *Queue Size*

The queue size at the router in front of the bottleneck link impacts the performance of TCP significantly. For example, TCP Reno at the sender reduces its transmission rate after experiencing packet loss. Thereby, the transmission rate may be reduced below the link capacity. Meanwhile, the queue in front of the bottleneck link has to hold enough data to utilize the link to achieve a full utilization. The queue must therefore be able to store sufficient data. If only one flow utilizes the link, this queue size should be set to the BDP, see [11, 126]. For many flows, smaller queue sizes are sufficient for an almost full saturation of the link [11].

We estimate the service of a TCP flow for the topology depicted in Fig. 8.14. We employ the congestion control protocol TCP Reno for all flows since analytical results exist for the dimensioning of queues for TCP Reno as mentioned before. We

---

[13] We display here the rate of the application layer; by subtracting packet headers a rate of 94 Mbps is achievable on a 100 Mbps Ethernet link, whereof the fair share of ten flows is 9.4 Mbps and using $r_{acc} = 1$ Mbps gives an expected estimate of 9 Mbps for the long-term rate.

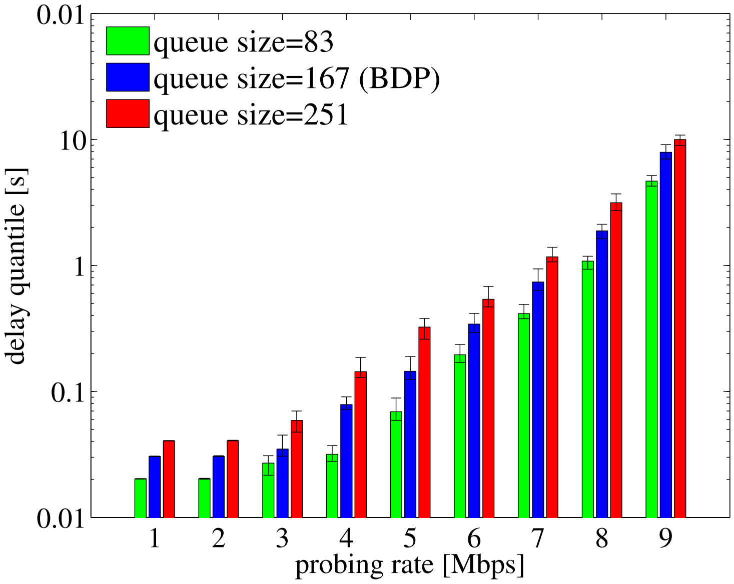


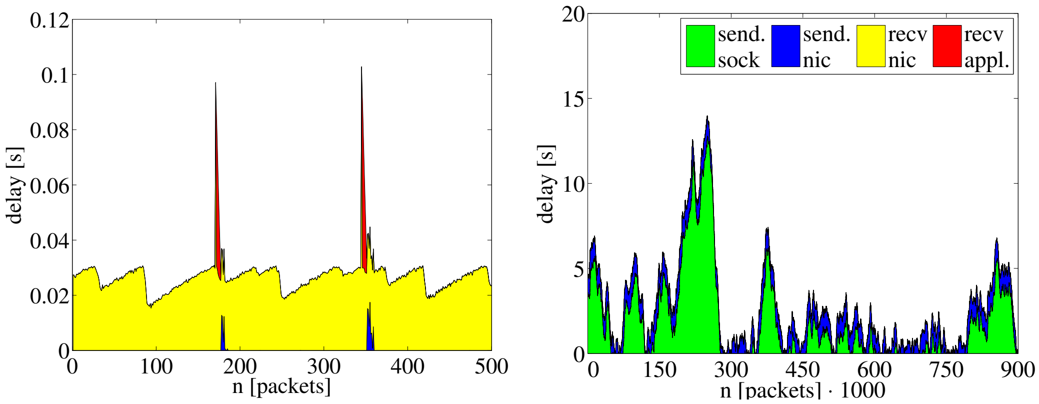
Figure 8.15: The figure shows the end-to-end delays for probing rates from 1 to 9 Mbps for the TCP through flow as 0.95 quantiles. The delay quantile increase significantly with the probing rate and the queue size.

estimate the service for a queue size equal to the BDP (167 packets), half of the BDP (83 packets), and one and a half of the BDP (251 packets).

The estimated delay quantiles for the construction of the service curves are presented in Fig. 8.15. It demonstrates that the delay increases with the queue size. An increase is expected since larger queues store more packets, which have to be processed. However, the magnitude of the increase is substantial. Only for small probing rates the delay quantile is close to the propagation delay of 10 ms plus the maximal queueing delay, e.g., it is 20 ms if the size is equal to the BDP. For higher rates the delay quantiles increase significantly. From this it follows that the queue size impacts the behavior of TCP by inducing further delays. To examine this effect, Fig. 8.16a and Fig. 8.16b show exemplary delay series of subsequent packets for a rate of of 1 Mbps and 9 Mbps for a queue size of 167 packets. Timestamps are captured at different measuring points to analyze where the delays originate from. The measuring points at the sender are: the scheduled transmission time at the application, the time at which the packet is delivered to the TCP socket [14], and the time at which the packet is passed to the network interface card (NIC). At the receiver the points are the time when the packet is transmitted from the NIC to

---

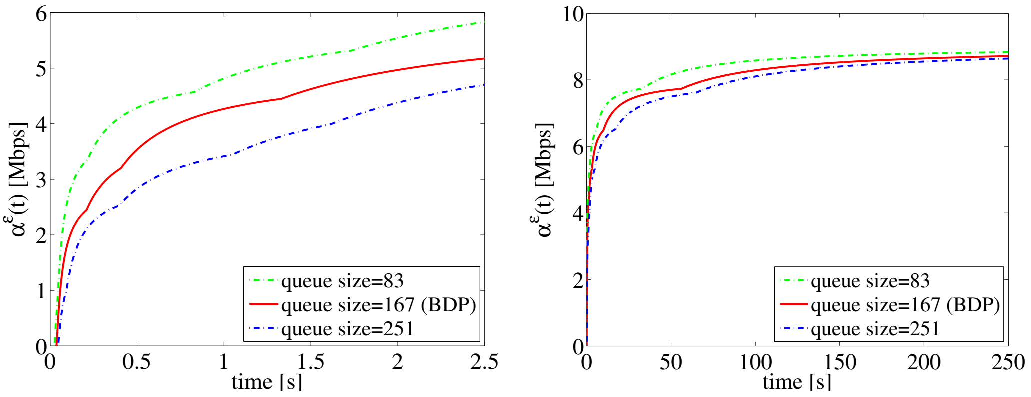
[14] To measure delays from the generation of a packet until it is transfered to the socket, we use (deviant to the other experiments in this section) non-blocking sockets and repeat the transfer until a packet is successfully transfered. This time interval equals the time the socket would block otherwise.



(a) arrival rate 1 Mbps (b) arrival rate 9 Mbps

Figure 8.16: The figures show the delay series for different sections for an end-to-end path. These sections are from the application to the socket at the sender (green area), from the socket to the NIC at the sender (blue area), from the NIC at the sender to the NIC at the receiver (yellow area), and from the NIC to the application (red area) at the receiver for a TCP flow with an application rate of 1 Mbps and 9 Mbps. For the low rate, delays are prevalent due to propagation and queueing at the router and reordering at the receiver; whereas for the high rate delays primarily arise from buffering and blocking at the socket of the sender. The cumulative value of the delays presents the end-to-end delay.

the TCP socket and the time at which the packet is received by the application. These five measuring points result in four delay values for each packet, which are presented as stack lines. Fig. 8.16a demonstrates that the delay for a low application rate arises from propagation and queueing in the network and buffering at the receiver, which is caused by head of line blocking if data are lost and are retransmitted to guarantee in-order delivery of data. On the contrary, Fig. 8.16b illustrates that for a probing rate, which is close to the long-term rate, high delay values arise due to buffering in the socket at the sender and due to waiting times at the application if the socket buffer is full. The network delay and the delay at the receiver are negligible in this case. This figure also shows that the delay values last for long periods. The accumulation of delays at the senders results from variations of the transmission rate of the TCP socket and the transmission rate of the application; if the application rate is higher than the transmission rate of the socket, delays accumulate due to buffering and blocking of the sockets. For example, the congestion window and thereby the transmission rate of the sender is adapted according to the experienced packet loss, which again is influenced by the queue size. Small buffers reduce the delay but may prevent the full utilization of the



(a) short-term (b) long-term

Figure 8.17: The figures show the available bandwidth $\alpha^{\varepsilon}(t)$ of TCP for different queue sizes. The estimates show the strong impact of the queue size and the beneficial characteristics of using a queue size equal to the BDP.

link. In this experiment, if the queue size is half of the BDP, the long-term rate of 9 Mbps is still achievable. This is indicated by the highest probing rate for which a steady state delay quantile exists. Delay series just give an extraction of the behavior of TCP, we present the complete behavior in Fig. 8.17 by the presentation of the service as ε-effective available bandwidth $\alpha^{\varepsilon}(t)$. We show the short-term and the long-term behavior in two separate figures since it extends to various magnitudes. Fig. 8.17b demonstrates the time scale at which the available bandwidth converges to the long-term rate; for the greatest queue size the rate converges at about 200 s. We show the short-term behavior in Fig. 8.17a for a time scale of 2.5 s. For example, this could be a time scale of interest due to buffering constrains or burst period lengths. The figure shows that for the queue size of one and a half of the BDP, the ε-effective available bandwidth $\alpha^{\varepsilon}(t)$ is about half of the long-term rate. For smaller queue sizes the ratio improves, but it is still significantly lower than the long-term rate. Practical relevance is that queues can be set to smaller sizes than the BDP even for a small number of flows. The maximal long-term rate can still be utilized. For a large number of flows, a nearly full link utilization may also be achieved with much smaller queues, which are significantly smaller than the BDP as shown in [11]. Using large buffers not only increases the delay due to greater processing times in the queues but also increases the delay induced by TCP itself substantially.

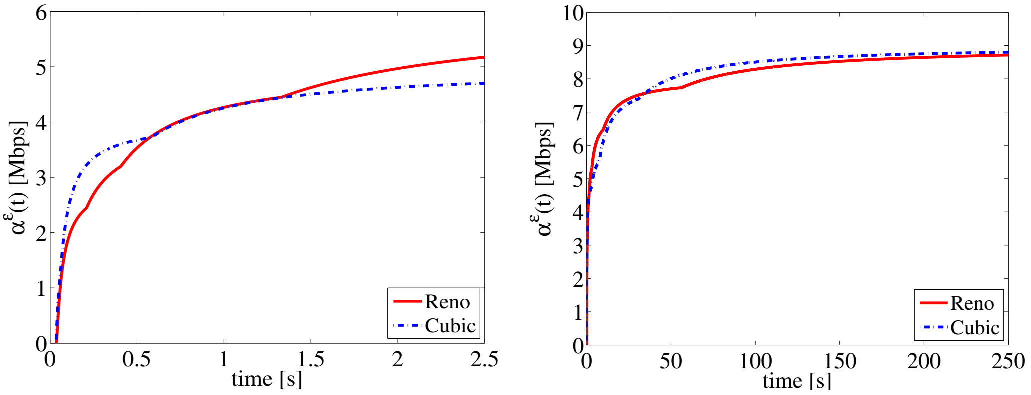
8.6.2 *Congestion Control Algorithm*

Many congestion control algorithms were developed in the past. TCP Reno[15] was the prevalent algorithm for many years, but it is replaced by TCP Compound (default in the Windows operating system) and TCP Cubic (default in the Linux operating system). Further TCP variants exist, e.g., TCP Vegas [19] and TCP FAST [129], just to name a few. Here, we compare the service of TCP Reno and TCP Cubic by estimation of the ε-effective available bandwidth $\alpha^{\varepsilon}(t)$. TCP Cubic and its predecessor TCP BIC are identified as the prevalent congestion control protocols on web servers in [133]. The most significant difference of TCP Reno and TCP Cubic is the adaptation of the congestion window. The size of the window regulates the amount of data that can be sent in one RTT by the sender, which leads to a time-varying transmission rate. Modifications to this adaptation behavior influence thereby the temporal behavior of the rate. TCP Reno and TCP Cubic differ in the way of the adaptation of the window during congestion avoidance: TCP Reno uses a linear increase, whereas TCP Cubic adapts the congestion window according to a cubical function. Furthermore, both protocols reduce the congestion window if packet loss is observed, but TCP Cubic reduces it by a smaller amount than TCP Reno. TCP Cubic is designed for networks with a high BDP to adapt its transmission rate fast to the high bandwidth of the link. For the topology in Fig. 8.14, the expected mean rate of one flow is about 10 Mbps and the RTT is 20 ms resulting in a BDP for this flow of $2 \cdot 10^5$ bit; in [57] such a network is already classified as a network with a long delay path.

We estimate the service for the congestion control protocols TCP Reno and TCP Cubic and illustrate the results as available bandwidth $\alpha^{\varepsilon}(t)$ in Fig. 8.18. The queue size at the routers equals the BDP of the link, which is 167 packets. The results show for the given scenario that the congestion control algorithm TCP Cubic slightly outperforms TCP Reno on short time-scales. Nevertheless, in the long-term both congestion control protocols are able to achieve the expected rate of 9 Mbps. Our system identification procedure gives a complete picture on all time scales of the two TCP congestion control algorithms.

---

[15] We employ the implementations available in the operating system Linux with kernel 3.5. We do therefore not differentiate between TCP Reno and TCP NewReno since also the TCP versions we use deviate from the specifications.



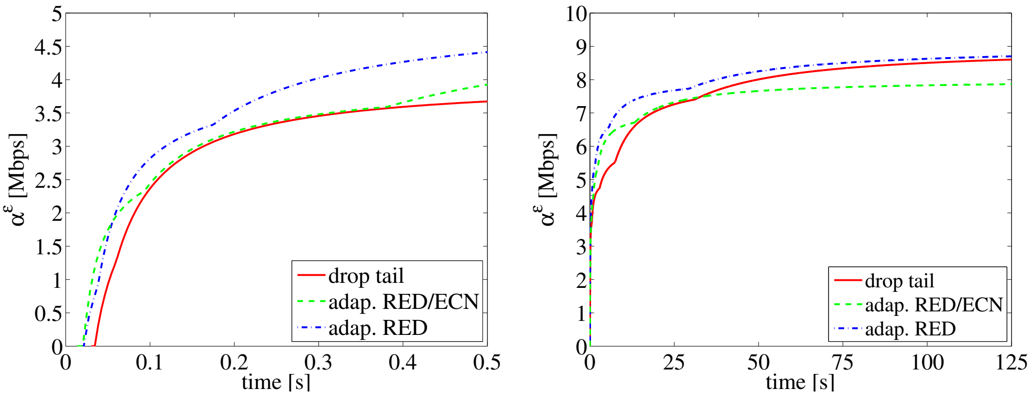
(a) short-term (b) long-term

Figure 8.18: The figures illustrate the ε-effective available bandwidth $\alpha^{\varepsilon}(t)$ of TCP for the congestion control protocols Reno and Cubic. The estimates in Fig. (b) show that both protocols achieve the expected rate of 9 Mbps in the long-term, but on short time-scales the performance of TCP Cubic is slightly better as Fig. (a) indicates.

8.6.3 *Active Queue Management*

A further approach to improve the performance of TCP is active queue management (AQM). Routers often use a FIFO drop tail queue, i.e., packets are processed in the order of arrival and if the buffer of the queue is full, arriving packets are dropped, which leads to correlated losses. AQM avoids this by signaling the congestion state of a queue to the sender before a queue overflows and so avoids bursty packet loss. Congestion signaling is performed by dropping single packets or marking by using ECN. One well-known implementation is RED [46] and its successor adaptive RED [47], which improves the sensitivity of RED with respect to the parameter settings. RED drops or marks packets randomly before the queue is filled completely. The probability that a packet is dropped or marked increases with the filling level of the queue after a minimal queue size is exceeded.

Here, we compare estimates for a FIFO drop tail queue, a queue using adaptive RED, the implementation in Linux since kernel 3.3., with and without ECN. Again, we use TCP Reno since RED was originally designed for it. For the FIFO drop tail queue, we set the queue size to the BDP (167 packets). We configure RED accordingly to the guidelines from [45], except the minimal threshold, which is set to 20 packets to achieve the long-term rate of 9 Mbps using RED without ECN.



(a) short-term (b) long-term

Figure 8.19: The figure compares the ε-effective available bandwidth estimates $\alpha^{\varepsilon}(t)$ for drop tail queueing to queues using adaptive RED. The estimates show that AQM can increase in particular the short-term behavior of TCP.

Using the recommended value of 5 packets, RED with and without ECN does achieve a long-term rate of 8 Mbps.

The results for the scenario, which is illustrated in Fig. 8.14, are presented as ε-effective available bandwidth in Fig. 8.19. All AQM implementations improve the short-term performance of TCP. Also in the long-term the performance improves as presented in Fig. 8.19b, except for adaptive RED with ECN. It achieves a long-term rate of 8 Mbps instead of 9 Mbps, which is achieved otherwise.

However, adaptive RED with ECN shows a better performance as FIFO drop tail queueing on short time-scale as presented in Fig. 8.19a. The best performance shows adaptive RED without ECN, the available bandwidth converges significantly faster to the long-term rate of 9 Mbps.

Since RED drops packets randomly, it reduces correlations between losses compared to the experiments using drop tail queueing. This allows the assumption that the correlation of losses or congestion signals has a fundamental impact on the performance of TCP since it changes the temporal behavior of the adaptation of the transmission rate. The smaller long-term rate achieved by the use of ECN may result from the faster notification of congestion signals. Reductions of the window occur immediately if a marked packet arrives at the sender instead of the identification by three duplicated acknowledgements or by a timeout. A further reason may be the implementation of the TCP state machine in the TCP stack in Linux. It handles such events as ECN by a different state than congestion identified by packet loss, see e.g. [117]. These conclusions are drawn from the knowledge of the protocols and scheduling disciplines and have to be investigated further by experiments that go beyond system identification. We started an analysis of the autocorrelation of the congestion window for the experimental setups in this Sec. 8.6, which is presented in the Appendix A.3. This analysis confirms our assumption that reducing correlations can positively impact the end-to-end delay.

9

CONCLUSION AND FUTURE WORK

In this thesis, we derived a system identification methodology in the framework of the stochastic network calculus for networks with random service and implemented it in practical probing procedures. The system model is represented by an ε-effective service curve that is identified by measurements. The identification methodology shows that with the usage of constant rate packet trains an inversion from delay or backlog measurements obtained in networks with stationary service to a service curve is feasible. Our system identification methodology applies to the class of linear time-variant systems. Thereby, it comprises networks in which service estimation is challenging due to e.g., variability of cross traffic, multiple bottleneck links, complex protocols, and non-work-conserving characteristics.

Furthermore, we showed the connection between service curves, which are established in the network calculus, and the definition of the available bandwidth for single-hop and multi-hop networks. This connection demonstrates the common ground of service curves and the available bandwidth but also the advantages of using a service curve for the representation of systems. We proved the equality of the left-over service curve, a concept from the network calculus, and the definition of the available bandwidth for single-hop networks. This connection between available bandwidth estimation and the network calculus enables us to analytically explain practical effects observed in available bandwidth estimation. When averaging is used to account for randomness inherent in networks, as many available bandwidth tools do, it leads to a systematic overestimation of the departures. For multi-hop networks, we showed that the service available is generally less than the definition of the available bandwidth for multi-hop systems predicts.

The methodology leaves open its practical implementation and in theory requires infinitely long packet trains, probing rates, and iterations of packet trains. By means of statistical tools, we implement practical probing procedures by limiting the probing parameters to finite values, which also allows the specification of the precision of the estimates by confidence intervals. We developed two practical probing procedures, one for the prompt estimation of service curves in networks in which assumptions of steady characteristics hold for short time scales. This objective is similar to the one of available bandwidth estimation. The second probing procedure aims at networks for testing purposes, in which stationarity can be assumed in the long-term. We support the development of the probing procedures by experiments to verify their functionality.

For the evaluation and validation of the system identification procedures, we compared our estimates to another service curve estimation approach, various available bandwidth estimation tools, and analytical references. The results show a good accuracy and the advantages of using a system identification procedure developed in the stochastic network calculus. These advantages include the representation of the service over arbitrary time scales by an ε-effective service curve and the derivation of probabilistic performance bounds for e.g., backlog and delay.

We applied the system identification to networks where modeling is challenging due to random cross traffic, various scheduling disciplines, non-work-conserving behavior, complex protocol characteristics, and time-varying channel conditions as existing in wireless channels. In detail, we estimated ε-effective service curves for wired networks with random cross-traffic and several scheduling disciplines, wireless IEEE 802.11 networks, and wired networks using the transport protocol TCP, which is the prevalent transport protocol in the Internet. Using the procedure, we could identify the service for such networks, where before only asymptotic results existed. Especially for TCP, we demonstrate the service of different congestion control protocols, buffer size configurations, and queueing disciplines by estimation.

In the area of available bandwidth estimation, two general probing procedures exist. Procedures that are congestion inducing, as our procedure, and procedures relying on single packet pairs or small trains consisting of few packets. We showed in this work that by using a congestion inducing procedure the inversion problem from measurements to a service curve is feasible. Packet trains carry information about the path they traverse, the train length reflects the scale for which this information is valid, or for sufficiently long trains the information applies to an unlimited scale. The developed inversion methodology relies on a black-box model since no specific assumptions are made on the internal system structure. It is left open, which information packet pairs or short packet trains contain about a network path. The challenge is to show if a system identification procedure that describes the path for larger scales as the packet pairs or trains last, can be established. Packet pairs or short trains last for a much shorter duration as the trains employed in this thesis. Thereby, they probably contain less information about the path, which may prevent inversion methodologies that use a black-box model, and the methodologies require additional assumptions to develop inversion methods using a gray-box model. Such a procedure is left for future work since it would deviate significantly from the methodology derived in this work.

So far, the system identification approach was applied to a subset of interesting networks, at which the estimation and modeling of the system is challenging. The approach allows the application to networks for which no system models exist so far. Already, for TCP the estimates show interesting effects on short time scales, which are worth further examination, and which can lead to a better understanding of the protocol behavior, the improvement of protocols, and the development of new protocols.

Part II

APPENDIX

A

AUXILIARY MATERIAL AND PROOFS

A.1 SYSTEM PROPERTIES IN THE MIN-PLUS AND MAX-PLUS ALGEBRA

In the classical system theory, systems are often categorized by the properties linearity and time-invariance. A dynamical system is linear if it fulfills the properties additivity $f(x+y) = f(x) + f(y)$ and homogeneity of degree one $af(x) = f(ax)$. It is time-invariant if a time-shift of the input signal $x(t-\delta)$ that produces the output $y(t-\delta) = \Pi(x(t-\delta))$ is equal to the signal $y(t) = \Pi(x(t))$ for all $\delta \geq 0$, where Π is the system operator and δ is a time-shift. In the following, we specify this properties in the min-plus and the max-plus algebra.

A.1.1 *Min-plus Algebra*

In the min-plus algebra, the plus is replaced by the infimum and the multiplication is replaced by the plus. The properties additivity, homogeneity, and time-invariance are listed below.

ADDITIVITY: $\inf(D_1(t), D_2(t)) = \Pi(\inf(A_1(t), A_2(t)))$

HOMOGENEITY OF DEGREE 1: $D(t) + a = \Pi(A(t) + a)$

TIME-INVARIANCE: A time-shift δ of the input $A(t)$ that produces the output $D(t) = \Pi(A(t))$, results in the same but time-shifted, output $D(\delta, t+\delta) = \Pi(A(\delta, t+\delta)), \forall\delta \geq 0$.

A.1.2 *Max-plus Algebra*

In the max-plus algebra, the plus is replaced by the maximum and the product is replaced by the plus. The properties additivity, homogeneity, and invariance become:

ADDITIVITY: $\max(T_{D_1}(n), T_{D_2}(n)) = \Pi(\max(T_{A_1}(n), T_{A_2}(n)))$

HOMOGENEITY OF DEGREE 1: $T_D(n) + a = \Pi(T_A(n) + a)$

INVARIANCE: A shift m of the input $T_A(n)$ that produces the output $T_D(n) = \Pi(T_A(n))$, results in the same, but shifted, output $T_D(m, n+m) = \Pi(T_A(m, m+n))$, $\forall m \geq 0$.

A.2 PROOFS

Proof (of Lem. 5.3) Consider a sample path $S^\omega(\tau, t)$ of $S(\tau, t)$ and fix $t \geq 0$. If $S^\omega(\tau, t) \geq \mathcal{S}^\varepsilon(t-\tau)$ for all $\tau \in [0, t]$, it follows from the monotonicity of the min-plus convolution that

$$D(t) = A \otimes S^\omega(t) \geq A \otimes \mathcal{S}^\varepsilon(t).$$

Since, by assumption, the condition $S^\omega(\tau, t) \geq \mathcal{S}^\varepsilon(t-\tau)$ holds for all $\tau \in [0, t]$ with probability $1 - \varepsilon$ at least, which completes the proof. ∎

Proof (of Thm. 5.2, [90]) From $B(t) = A(t) - D(t)$ and Eq. (2.10) it follows that

$$B(t) = \sup_{\tau \in [0,t]} \{A(\tau, t) - S(\tau, t)\}. \tag{A.1}$$

The supremum in Eq. (A.1) implies that $B(t) \geq A(\tau,t) - S(\tau,t)$ for all $\tau \in [0,t]$, permitting us to write

$$S(\tau,t) \geq A(\tau,t) - B(t)\ , \forall \tau \in [0,t].$$

Inserting $A(\tau,t) = r(t-\tau)$ and using the backlog quantile yields

$$\mathsf{P}\left[S(\tau,t) \geq r(t-\tau) - B^{\xi}(r,t)\ , \forall \tau\right] \geq 1 - \xi.$$

Using the complement and applying the union bound for a set of rates R it follows that

$$\mathsf{P}\left[\bigcup_{r\in R} S(\tau,t) < r(t-\tau) - B^{\xi}(r,t)\ , \forall \tau\right] \leq \sum_{r\in R} \xi.$$

Taking again the complement, it holds for the maximum over the individual rates that

$$\mathsf{P}\left[S(\tau,t) \geq \max_{r\in R}\{r(t-\tau) - B^{\xi}(r,t)\}\ , \forall \tau\right] \geq 1 - \sum_{r\in R} \xi.$$

With Lem. 5.3 we obtain that $\mathcal{S}^{\varepsilon}(t-\tau)$ defined as

$$\mathcal{S}^{\varepsilon}(t-\tau) = \max_{r\in R}\{r(t-\tau) - B^{\xi}(r,t)\}$$

for all $\tau \in [0,t]$ is an ε-effective service curve with violation probability $\varepsilon = \sum_{r\in R} \xi$. Letting $t \to \infty$ and inserting the steady state backlog quantile $B^{\xi}(r)$ completes the proof. ∎

Proof (of Lem. 5.4, [90]) From Eq. (A.1) it follows for any x and $\delta > 0$ that

$$\begin{aligned} &\mathsf{P}[B(t+\delta) \geq x] \\ &= \mathsf{P}\left[\sup_{\tau\in[0,t+\delta]} \{A(\tau,t+\delta) - S(\tau,t+\delta)\} \geq x\right] \\ &\geq \mathsf{P}\left[\sup_{\tau\in[0,t]} \{A(\tau+\delta,t+\delta) - S(\tau+\delta,t+\delta)\} \geq x\right]. \end{aligned}$$

From the assumption of joint stationarity $A(\tau+\delta, t+\delta)$ and $S(\tau+\delta, t+\delta)$ are equal in distribution to $A(\tau, t)$ and $S(\tau, t)$, respectively, for all $\tau \leq t$, and $\delta > 0$. The last line equals $\mathsf{P}[B(t) \geq x]$ so that we get $\mathsf{P}[B(t+\delta) \geq x] \geq \mathsf{P}[B(t) \geq x]$, thus, proving the first claim.

For the second claim, if the given inequality holds, then there exists a finite random variable

$$T = \sup\{\delta \geq 0 : A(t-\delta, t) \geq S(t-\delta, t)\},$$

for any t. Consequently, $A(t-\delta, t) < S(t-\delta, t)$ holds for all $\delta > T$ with probability one. Moreover, since $A(t-\delta, t)$ is non-decreasing in $\delta \geq 0$, it follows that $A(t-\delta, t) < S(t-T-\vartheta, t)$ for all $0 \leq \delta \leq T$ and any $\vartheta > 0$. Combining the two statements and using that $S(t-\delta, t)$ for $\delta \geq 0$ and $S(t-T-\vartheta, t)$ are non-negative, yields

$$A(t-\delta, t) - S(t-\delta, t) \leq S(t-T-\vartheta, t)$$

for all $\delta \geq 0$. Hence,

$$\sup_{\delta \geq 0}\{A(t-\delta, t) - S(t-\delta, t)\} \leq S(t-T-\vartheta, t).$$

With $\sup_{\delta \geq 0}\{A(t-\delta, t) - S(t-\delta, t)\} = B(t)$ from Eq. (A.1) it follows for any x that

$$\sup_{t}\{\mathsf{P}[B(t) \geq x]\} \leq \sup_{t}\{\mathsf{P}[S(t-T-\vartheta, t) \geq x]\}.$$

Since T is finite and $B(t)$ is stochastically increasing there exists a finite random variable B such that

$$\lim_{t\to\infty} \mathsf{P}[B(t) \geq x] = \sup_{t}\{\mathsf{P}[B(t) \geq x]\} = \mathsf{P}[B \geq x],$$

which completes the proof of the second claim. ∎

Proof (of Thm. 5.3) From $B(t) = \lfloor A(t) \rfloor - \lfloor D(t) \rfloor$, Eq. (5.11), and Eq. (2.10) it follows that

$$B(t) = \sup_{\tau \in [0,t]} \{ \lfloor A(t) \rfloor - \lfloor A(\tau) \rfloor - \lfloor S(\tau, t) \rfloor \}. \tag{A.2}$$

The supremum in Eq. (A.2) implies that $B(t) \geq \lfloor A(t) \rfloor - \lfloor A(\tau) \rfloor - \lfloor S(\tau, t) \rfloor$ for all $\tau \in [0, t]$, permitting us to write

$$\lfloor S(\tau, t) \rfloor \geq \lfloor A(t) \rfloor - \lfloor A(\tau) \rfloor - B(t) \ , \forall \tau \in [0, t].$$

With packet rate r, inserting $A(t) = \lfloor rt \rfloor$ and using the backlog quantile yields

$$\mathsf{P}\left[\lfloor S(\tau, t) \rfloor \geq \lfloor rt \rfloor - \lfloor r\tau \rfloor - B^{\xi}(r, t) \ , \forall \tau \right] > 1 - \xi.$$

Since it holds that $\lfloor rt \rfloor - \lfloor r\tau \rfloor = \lfloor \lfloor rt \rfloor - \lfloor r\tau \rfloor \rfloor = \lfloor rt - \lfloor r\tau \rfloor \rfloor \geq \lfloor rt - r\tau \rfloor$, we get

$$\mathsf{P}\left[\lfloor S(\tau, t) \rfloor \geq \lfloor r(t - \tau) \rfloor - B^{\xi}(r, t) \ , \forall \tau \right] \geq 1 - \xi.$$

By application of the union bound it follows that

$$\mathsf{P}\left[\lfloor S(\tau, t) \rfloor \geq \max_{r \in R} \{ \lfloor r(t - \tau) \rfloor - B^{\xi}(r, t) \} \ , \forall \tau \right] \geq 1 - \sum_{r} \xi.$$

With Lem. 5.3 we obtain that $\mathcal{S}^{\varepsilon}(t - \tau)$ defined as

$$\mathcal{S}^{\varepsilon}(t - \tau) = \max_{r \in R} \{ \lfloor r(t - \tau) \rfloor - B^{\xi}(r, t) \}$$

for all $\tau \in [0, t]$ is an ε-effective service curve with violation probability $\varepsilon = \sum_r \xi$ for packetized arrivals. Finally, letting $t \to \infty$ and inserting the steady state backlog $B^{\xi}(r)$ gives

$$\mathcal{S}^{\varepsilon}(t) = \max_{r \in R} \{ \lfloor rt \rfloor - B^{\xi}(r) \},$$

which completes the proof. Note that for the backlog it holds that $\lfloor B^{\xi}(r,t) \rfloor = B^{\xi}(r,t)$ and hence $\lfloor \mathcal{S}^{\varepsilon}(t) \rfloor = \mathcal{S}^{\varepsilon}(t)$ as well as

$$\mathcal{S}^{\varepsilon}(t) = \lfloor \max_{r \in R} \{rt - B^{\xi}(r)\} \rfloor .$$

∎

A.3 CORRELATION OF THE TCP CONGESTION WINDOW

In Sec. 8.6.3, we presume that the performance variations observed for TCP in the experiments presented in Sec. 8.6 arise from correlations in the adaptation of the congestion window (CWND). Here, we elaborate on this argument by the evaluation of the autocorrelation of the CWND. The experiments show that most TCP variants achieve the expect mean rate of 9 Mbps but deviate significantly in the short-term behavior. If an application rate is below the long-term rate a TCP connection provides, the CWND must at least support this application rate on average, which follows from the self-clocking behavior described in [56]. Hence, we compare in particular the empirical autocorrelation function of the CWND process $C(t)$ for the experiments from Sec. 8.6. The autocorrelation function $\mathcal{C}_\tau$ at lag τ is defined as [50]:

$$\mathcal{C}_\tau = \frac{\mathsf{E}[(C(t) - \mu_C)(C(t+\tau) - \mu_C)]}{\sigma_C^2}, \tag{A.3}$$

where μ_C is the mean value of $C(t)$ and σ_C^2 its variance. This enables a comparison of different experimental setups since it is adjusted by the mean, which in turn reflects roughly the average rate. A direct comparison of the CWND is impractical since its mean and its absolute values vary due to the configurations, congestion control protocols, or the AQM strategy. If adjusted by the mean and normalized as in Eq. (A.3), the values become comparable. A positive correlation that is sustained for longer time periods implies that the transmission rate varies slowly. For example, if the CWND falls below the mean, a strong positive correlation for long time-periods implies that the window stays below the mean for long-time periods, vice versa,

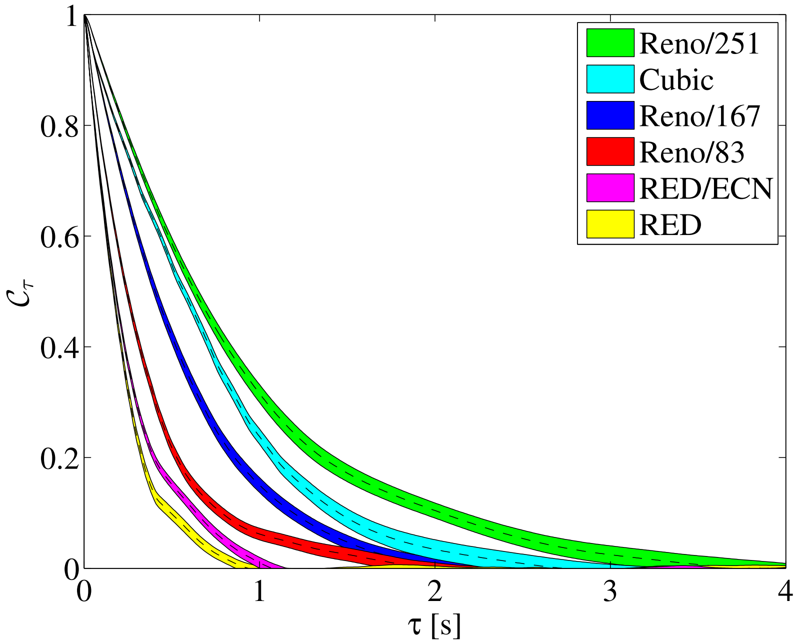


Figure A.1: Auto correlation of the CWND. For large queue sizes the autocorrelation spans across large periods. Small queue sizes and the AQM strategy RED reduce the correlation significantly. Comparing the autocorrelation to the delay quantiles in Tab. A.1 indicates that autocorrelation of the CWND for long periods induce high delays.

Table A.1: 95% delay quantile in milliseconds for a probing rate of 5 Mbps

RED	Reno/83	RED/ECN	Reno/167	Cubic	Reno/251
56	69	105	144	147	325

if the CWND is above the mean, it remains above it for long periods. During the time in which the CWND is below its mean value, data from the application are buffered at the socket, and in periods above the mean the buffer is drained. We sample the CWND at the sender each 10 ms, which is about twice per RTT, during the transmission of $1 \cdot 10^6$ packets sent at an application rate of 5 Mbps. This rate represents a moderate utilization with end-to-end delays whereof a fraction of the delays is due to buffering in the network stack since the delays are above the maximal queueing delay in the network.

For each experimental setup from that section, we repeat the sampling 20 times to calculate confidence intervals. Fig. A.1 presents the autocorrelation and the confidence intervals. Comparing the development of the autocorrelations of the CWND and the delay quantiles shown in Tab. A.1, it is noticeable that for the experiments with a larger correlation the delay quantile is greater, too. Using larger queue sizes, such a behavior is obvious since the time periods between packet drops increase due to larger queues. Moreover, the results also show that the use of the congestion control algorithm Cubic, which changes the adaptation of the CWND, increases the correlation. The correlation is greater than for TCP Reno for a small number of lags but decays faster for larger lags. The autocorrelation decreases fast for RED (with and without ECN). Since RED drops or marks packets randomly, it reduces the correlation of the CWND.

These findings imply that for the configuration of TCP, the design of protocols, or queueing strategies for TCP a major characteristic is the temporal correlation behavior besides the long-term rate. In particular, these results show that reducing the correlation of the CWND has a positive impact on the end-to-end delays.

BIBLIOGRAPHY

[1] Future Internet Lab. URL `http://www.filab.uni-hannover.de/`.

[2] R. URL `http://www.r-project.org/`.

[3] Iperf. URL `http://sourceforge.net/projects/iperf/`.

[4] OMNeT++. URL `http://www.omnetpp.org`.

[5] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, ieee 802.11-2012 edition, 2012.

[6] F. Agharebparast and V. C. M. Leung. Slope domain modeling and analysis of data communication networks: A network calculus complement. In *Proc. IEEE ICC*, pages 591–596, June 2006.

[7] R. Agrawal and R. Rajan. Performance bounds for guaranteed and adaptive services. Technical report, IBM T.J. Watson Research Center, May 1996.

[8] R. Agrawal, R. Cruz, C. Okino, and R. Rajan. Performance bounds for flow control protocols. *IEEE/ACM Trans. Netw.*, 7(3):310 –323, June 1999.

[9] M. Allman, V. Paxson, and E. Blanton. TCP Congestion Control. RFC 5681 (Draft Standard), Sept. 2009. URL `http://www.ietf.org/rfc/rfc5681.txt`.

[10] L. Andrew, E. S. Floyd, and G. Wang. Common TCP Evaluation Suite, Jan. 2010. URL `http://tools.ietf.org/id/draft-irtf-tmrg-tests-02.txt`. expired.

[11] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. *ACM SIGCOMM Comput. Commun. Rev.*, 34(4):281–292, Aug. 2004.

[12] K. Åström and P. Eykhoff. System identification—a survey. *Automatica*, 7(2): 123 – 162, Mar. 1971.

[13] K. J. Åström and B. Wittenmark. *Adaptive control*. Courier Dover Publications, 2008.

[14] F. Baccelli and D. Hong. TCP is max-plus linear and what it tells us on its throughput. *ACM SIGCOMM Comput. Commun. Rev.*, 30(4):219–230, Aug. 2000.

[15] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems.* John Wiley & Sons Ltd., 1992.

[16] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot. The role of PASTA in network measurement. *IEEE/ACM Trans. Netw.*, 17(4):1340–1353, Aug. 2009.

[17] J. Beirlant, Y. Goegebeur, J. Teugels, J. Segers, D. De Waal, and C. Ferro. *Statistics of extremes. Theory and applications.* John Wiley and Sons., 2004.

[18] A. Botta, A. Dainotti, and A. Pescape. Multi-protocol and multi-platform traffic generation and measurement. In *Proc. IEEE INFOCOM Demo*, May 2007.

[19] L. Brakmo and L. Peterson. TCP Vegas: end to end congestion avoidance on a global Internet. *IEEE J. Sel. Areas Commun.*, 13(8):1465–1480, Oct. 1995.

[20] M. Bredel and M. Fidler. A measurement study of bandwidth estimation in IEEE 802.11g wireless LANs using the DCF. In *Proc. IFIP Networking*, pages 314–325, May 2008.

[21] M. Bredel and M. Fidler. Understanding fairness and its impact on quality of service in IEEE 802.11. In *Proc. IEEE INFOCOM*, pages 1098–1106, Apr. 2009.

[22] M. Bredel, Z. Bozakov, and Y. Jiang. Analyzing router performance using network calculus with external measurements. In *Proc. IEEE IWQoS*, June 2010.

[23] R. Bringhurst. *The Elements of Typographic Style.* Version 2.5. Hartley & Marks, Publishers, 2002.

[24] E. Brosh, S. Baset, V. Misra, D. Rubenstein, and H. Schulzrinne. The delay-friendliness of TCP for real-time traffic. *IEEE/ACM Trans. Netw.*, 18(5):1478 –1491, Oct. 2010.

[25] A. Burchard, J. Liebeherr, and S. Patek. A min-plus calculus for end-to-end statistical service guarantees. *IEEE Trans. Inf. Theory*, 52(9):4105–4114, Sept. 2006.

[26] V. Cerf and R. Kahn. A protocol for packet network intercommunication. *IEEE Trans. Commun.*, 22(5):637–648, May 1974.

[27] C.-S. Chang. Stability, queue length and delay of deterministic and stochastic queueing networks. *IEEE Trans. Autom. Control*, 39(5):913–931, May 1994.

[28] C.-S. Chang. On deterministic traffic regulation and service guarantees: A systematic approach by filtering. *IEEE Trans. Inf. Theory*, 44(3):1097–1110, May 1998.

[29] C.-S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[30] F. Ciucu. *Scaling Properties in the Stochastic Network Calculus*. PhD thesis, University of Virginia, Aug. 2007.

[31] R. L. Cruz. A calculus for network delay, Part I: Network elements in isolation. *IEEE Trans. Inf. Theory*, 37(1):114–131, Jan. 1991.

[32] R. L. Cruz. A calculus for network delay, Part II: Network analysis. *IEEE Trans. Inf. Theory*, 37(1):132–141, Jan. 1991.

[33] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE J. Sel. Areas Commun.*, 13(6):1048–1056, Aug. 1995.

[34] R. L. Cruz and C. M. Okino. Service guarantees for window flow control. In *Proc. Allerton Conf. on Communications, Control and Computing*, Oct. 1996.

[35] B. K. Dey, D. Manjunath, and S. Chakraborty. Estimating network link characteristics using packet-pair dispersion: A discrete-time queueing theoretic analysis. *Computer Networks*, 55(5):1052–1068, Apr. 2011.

[36] H. Drees, L. d. Haan, and D. Li. Approximations to the tail empirical distribution function with application to testing extreme value conditions. *Journal of Statistical Planning and Inference*, 136(10):3498–3538, 2006.

[37] N. Dukkipati, M. Mathis, Y. Cheng, and M. Ghobadi. Proportional rate reduction for TCP. In *Proc. ACM IMC*, pages 155–170, Nov. 2011.

[38] S. Ekelin et al. Real-time measurement of end-to-end available bandwidth using Kalman filtering. In *Proc. IEEE/IFIP NOMS*, pages 73–84, Apr. 2006.

[39] G. Elliott, T. J. Rothenberg, and J. H. Stock. Efficient tests for an autoregressive unit root. *Econometrica*, 64(4):813–836, July 1996.

[40] B. Fan, H. Zhang, and W. Dou. A max-plus network calculus. In *Proc. IEEE/ACIS ICIS*, pages 149–154, June 2009.

[41] D. Ferrari. Client requirements for real-time communication services. *IEEE Commun. Mag.*, 28(11):65 –72, Nov. 1990.

[42] M. Fidler. An end-to-end probabilistic network calculus with moment generating functions. In *Proc. IEEE IWQoS*, pages 261–270, June 2006.

[43] M. Fidler. A survey of deterministic and stochastic service curve models in the network calculus. *IEEE Commun. Surveys Tuts.*, 12(1):59–86, Feb. 2010.

[44] M. Fidler and S. Recker. Conjugate network calculus: A dual approach applying the legendre transform. *Computer Networks*, 50(8):1026–1039, June 2006.

[45] S. Floyd. RED: Discussions of setting parameters, Nov. 1997. URL `http://www.icir.org/floyd/REDparameters.txt`.

[46] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1(4):397–413, Aug. 1993.

[47] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. Technical report, AT&T Center for Internet Research at ICSI, 2001.

[48] J. Gettys and K. Nichols. Bufferbloat: dark buffers in the Internet. *ACM Commun.*, 55(1):57–65, Jan. 2012.

[49] J. Gibbons and S. Chakraborti. *Nonparametric Statistical Inference*. CRC Press, 4th edition, 2003.

[50] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, New York, 2001.

[51] S. Ha, I. Rhee, and L. Xu. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Oper. Syst. Rev.*, 42(5):64–74, July 2008.

[52] P. Hága, K. Diriczi, G. Vattay, and I. Csabai. Understanding packet pair separation beyond the fluid model: The key role of traffic granularity. In *Proc. IEEE INFOCOM*, pages 2374–2386, Apr. 2006.

[53] T. Hisakado, K. Okumura, V. Vukadinovic, and L. Trajkovic. Characterization of a simple communication network using Legendre transform. In *Proc. IEEE ISCAS*, pages 738–741, May 2003.

[54] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE J. Sel. Areas Commun.*, 21(6):879–894, Aug. 2003.

[55] V. Jacobson. Pathchar - a tool to infer characteristics of Internet paths. URL `ftp://ftp.ee.lbl.gov/pathchar/`.

[56] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM Comput. Commun. Rev.*, 18(4):314–329, Aug. 1988.

[57] V. Jacobson and R. Braden. TCP extensions for long-delay paths. RFC 1072 (Historic), Oct. 1988. URL `http://www.ietf.org/rfc/rfc1072.txt`. Obsoleted by RFCs 1323, 2018, 6247.

[58] M. Jain and C. Dovrolis. Pathload: A measurement tool for end-to-end available bandwidth. In *Proc. PAM*, pages 14–25, Mar. 2002.

[59] M. Jain and C. Dovrolis. End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput. *IEEE/ACM Trans. Netw.*, 11(4):537–549, Aug. 2003.

[60] M. Jain and C. Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *Proc. ACM IMC*, pages 272–277, Oct. 2004.

[61] Y. Jiang and Y. Liu. *Stochastic Network Calculus*. Springer, 2008.

[62] A. Johnsson and M. Bjorkman. On measuring available bandwidth in wireless networks. In *Proc. IEEE LCN*, pages 861–868, Oct. 2008.

[63] A. Johnsson, B. Melander, and M. Björkman. Diettopp: A first implementation and evaluation of a simplified bandwidth measurement method. In *Proc. Swedish National Computer Networking Workshop*, Nov. 2004.

[64] A. Johnsson, B. Melander, and M. Björkman. Bandwidth measurement in wireless networks. In *Proc. Med-Hoc-Net*, pages 89–98, June 2005.

[65] A. Johnsson, M. Bjorkman, and B. Melander. An analysis of active end-to-end bandwidth measurements in wireless networks. In *Proc. IEEE/IFIP E2EMON*, pages 74–81, Apr. 2006.

[66] J. Jun, P. Peddabachagari, and M. Sichitiu. Theoretical maximum throughput of IEEE 802.11 and its applications. In *Proc. IEEE NCA*, pages 249–256, Apr. 2003.

[67] P. Kanuparthy, C. Dovrolis, K. Papagiannaki, S. Seshan, and P. Steenkiste. Can user-level probing detect and diagnose common home-WLAN pathologies. *ACM SIGCOMM Comput. Commun. Rev.*, 42(1):7–15, Jan. 2012.

[68] A. Karumanchi, S. Talabattula, K. Rao, and S. Varadarajan. A shift varying filtering theory for dynamic service guarantees. In *Proc. QoS-IP*, pages 613–625, Feb. 2005.

[69] K. Keesman. *System Identification: An Introduction*. Advanced Textbooks in Control and Signal Processing. Springer, 2011.

[70] L. Kleinrock. *Queueing Systems: Theory*. Number Volume 1: Theory in A Wiley-Interscience publication. Wiley, 1976.

[71] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Proc. ACM SIGMETRICS*, pages 128–139, June 1992.

[72] J. Laine, S. Saaristo, and R. Prior. (c)rude. URL `http://rude.sourceforge.net/`.

[73] J.-Y. Le Boudec. *Performance Evaluation of Computer and Communication Systems*. EPFL Press, 2010.

[74] J.-Y. Le Boudec and P. Thiran. *Network Calculus A Theory of Deterministic Queuing Systems for the Internet*. LNCS. Springer, 2001.

[75] J.-Y. Le Boudec and D.-C. Tomozei. A demand-response calculus with perfect batteries. In *Proc. WoNeCa*, pages 273–287, Mar. 2012.

[76] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, Feb. 1994.

[77] D. Li and J. Huesler. Testing extreme value conditions. URL `http://my.gl.fudan.edu.cn/teacherhome/lideyuan/research.html`.

[78] J. Li, M. Claypool, and R. R. Kinicki. Wbest: A bandwidth estimation tool for IEEE 802.11 wireless networks. In *Proc. IEEE LCN*, pages 374–381, Oct. 2008.

[79] M. Li, Y.-L. Wu, and C.-R. Chang. Available bandwidth estimation for the network paths with multiple tight links and bursty traffic. *Network and Computer Applications*, 36(1):353 – 367, Jan. 2013.

[80] J. Liebeherr, M. Fidler, and S. Valaee. A system theoretic approach to bandwidth estimation. *IEEE/ACM Trans. Netw.*, 18(4):1040–1053, Aug. 2010.

[81] J. Liebeherr, A. Burchard, and F. Ciucu. Delay bounds in communication networks with heavy-tailed and self-similar traffic. *IEEE Trans. Inf. Theory*, 58(2): 1010–1024, Feb. 2012.

[82] X. Liu, K. Ravindran, and D. Loguinov. A queuing-theoretic foundation of available bandwidth estimation: Single-hop analysis. *IEEE/ACM Trans. Netw.*, 15(4):918–931, Aug. 2007.

[83] X. Liu, K. Ravindran, and D. Loguinov. A stochastic foundation of available bandwidth estimation: Multi-hop analysis. *IEEE/ACM Trans. Netw.*, 16(1):130–143, Apr. 2008.

[84] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 2nd edition, Jan. 1999.

[85] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34 (1):1 – 12, 2010.

[86] R. Lübben and M. Fidler. On the delay performance of block codes for discrete memoryless channels with feedback. In *Proc. IEEE Sarnoff Symposium*, May 2012.

[87] R. Lübben and M. Fidler. Non-equilibrium information envelopes and the capacity-delay-error-tradeoff of source coding. In *Proc. IEEE WoWMoM*, June 2012.

[88] R. Lübben, M. Fidler, and J. Liebeherr. A foundation for stochastic bandwidth estimation of networks with random service. *Computing Research Repository*, abs/1008.0050, July 2010. URL `http://arxiv.org/abs/1008.0050`. extended version of [89].

[89] R. Lübben, M. Fidler, and J. Liebeherr. A foundation for stochastic bandwidth estimation of networks with random service. In *Proc. IEEE INFOCOM*, pages 1817–1825, Apr. 2011.

[90] R. Lübben, M. Fidler, and J. Liebeherr. Stochastic bandwidth estimation in networks with random service. *IEEE/ACM Trans. Netw.*, (99), May 2013. accepted for publication.

[91] S. Machiraju, D. Veitch, F. Baccelli, and J. Bolot. Adding definition to active probing. *ACM SIGCOMM Comput. Commun. Rev.*, 37(2):19–28, Apr. 2007.

[92] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband Internet traffic. In *Proc. ACM IMC*, pages 90–102, Nov. 2009.

[93] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM SIGCOMM Comput. Commun. Rev.*, 27(3):67–82, July 1997.

[94] B. Melander, M. Björkman, and P. Gunningberg. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In *Proc. IEEE GLOBECOM*, pages 415–420, Nov. 2000.

[95] B. Melander, M. Björkmann, and P. Gunningberg. First-come-first-served packet dispersion and implications for TCP. In *Proc. IEEE GLOBECOM*, pages 2170–2174, Nov. 2002.

[96] C. A. Melo and N. L. da Fonseca. Envelope process and computation of the equivalent bandwidth of multifractal flows. *Computer Networks*, 48(3):351–375, June 2005.

[97] D. L. Mills. Network time protocol version 4 reference and implementation guide. Technical report, University of Delaware, June 2006.

[98] K. Murota. *Discrete Convex Analysis*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 1987.

[99] R. Neapolitan and K. Naimipour. *Foundations of Algorithms*. Jones & Bartlett Learning, 2010.

[100] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation. *ACM SIGCOMM Comput. Commun. Rev.*, 28(4):303–314, Oct. 1998.

[101] K. Pandit, J. Schmitt, C. Kirchner, and R. Steinmetz. A transform for network calculus and its application to multimedia networking. In *Proc. SPIE/ACM MMCN*, Jan. 2006.

[102] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Trans. Netw.*, 1(3):344–357, June 1993.

[103] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple-node case. *IEEE/ACM Trans. Netw.*, 2(2):137–150, Apr. 1994.

[104] K.-J. Park, H. Lim, and C.-H. Choi. Stochastic analysis of packet-pair probing for network bandwidth estimation. *Computer Networks*, 50(12):1901–1915, May 2006.

[105] K.-J. Park, H. Lim, J. C. Hou, and C.-H. Choi. Feedback-assisted robust estimation of available bandwidth. *Computer Networks*, 53(7):896–912, May 2009.

[106] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Trans. Netw.*, 3(3):226 –244, June 1995.

[107] B. Pfaff. *Analysis of integrated and cointegrated time series with R.* Springer, 2006.

[108] M. Portoles-Comeras, A. Cabellos-Aparicio, J. Mangues-Bafalluy, A. Banchs, and J. Domingo-Pascual. Impact of transient CSMA/CA access delays on active bandwidth measurements. In *Proc. ACM IMC*, pages 397–409, Nov. 2009.

[109] J. Postel. DoD standard transmission control protocol. RFC 761, Jan. 1980. URL `http://www.ietf.org/rfc/rfc761.txt`. Obsoleted by RFC 793.

[110] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168 (Proposed Standard), Sept. 2001. URL `http://www.ietf.org/rfc/rfc3168.txt`. Updated by RFCs 4301, 6040.

[111] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. PathChirp: Efficient available bandwidth estimation for network paths. In *Proc. PAM*, Apr. 2003.

[112] J. Ridoux, D. Veitch, and T. Broomhead. The case for feed-forward clock synchronization. *IEEE/ACM Trans. Netw.*, 20(1):231–242, Feb. 2012.

[113] A. Rizk and M. Fidler. Non-asymptotic end-to-end performance bounds for networks with long range dependent fBm cross traffic. *Computer Networks*, 56 (1):127–141, Jan. 2012.

[114] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1972.

[115] M. Sargent, E. Blanton, and M. Allman. Modern Application Layer Transmission Patterns from a Transport Perspective. Technical report, Case Western Reserve University, International Computer Science Institute, May 2013. draft.

[116] H. Sariowan, R. L. Cruz, and G. C. Polyzos. Scheduling for quality of service guarantees via service curves. In *Proc. IEEE ICCCN*, pages 512–520, Sept. 1995.

[117] P. Sarolahti and A. Kuznetsov. Congestion control in linux TCP. In *Proc. USENIX Annual Technical Conference, FREENIX Track*, pages 49–62, June 2002.

[118] J. Schmitt and U. Roedig. Sensor network calculus - a framework for worst case analysis. In *Distributed Computing in Sensor Systems*, volume 3560 of *LNCS*, pages 467–467. Springer, 2005.

[119] M. Sedighizad, B. Seyfe, and K. Navaie. MR-BART: Multi-rate available bandwidth estimation in real-time. *Network and Computer Applications*, 35(2):731 – 742, Mar. 2012.

[120] H. She. *Performance Analysis and Deployment Techniques for Wireless Sensor Networks*. PhD thesis, KTH Royal Institute of Technology, Stockholm, 2012.

[121] H. She, Z. Lu, A. Jantsch, and L.-R. Zheng. Estimation of statistical bandwidth through backlog measurement. In *Proc. WoNeCa*, pages 11–14, Mar. 2012.

[122] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proc. ACM IMC*, pages 39–44, Oct. 2003.

[123] L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *Proc. IEEE ISCAS*, pages 101–104, June 2000.

[124] F. Thouin, M. Coates, and M. Rabbat. Large scale probabilistic available bandwidth estimation. *Computer Networks*, 55(9):2065–2078, June 2011.

[125] A. Undheim, Y. Jiang, and P. Emstad. Network calculus approach to router modeling with external measurements. In *Proc. IEEE CHINACOM*, pages 276–280, Aug. 2007.

[126] C. Villamizar and C. Song. High performance TCP in ANSNET. *ACM SIGCOMM Comput. Commun. Rev.*, 24(5):45–60, Oct. 1994.

[127] L. Von Bertalanffy. An outline of general system theory. *British Journal for the Philosophy of Science*, 1(2):134–165, Aug. 1950.

[128] K. Wang, F. Ciucu, C. Lin, and S. H. Low. A stochastic power network calculus for integrating renewable energy sources into the power grid. *IEEE J. Sel. Areas Commun.*, 30(6):1037–1048, July 2012.

[129] D. X. Wei, C. Jin, S. H. Low, and S. Hegde. FAST TCP: Motivation, architecture, algorithms, performance. *IEEE/ACM Trans. Netw.*, 14(6):1246–1259, Dec. 2006.

[130] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):255–270, Dec. 2002. ISSN 0163-5980.

[131] K. Wu, Y. Jiang, and D. Marinakis. A stochastic calculus for network systems with renewable energy sources. In *Proc. IEEE INFOCOM Workshops*, pages 109–114, Mar. 2012.

[132] J. Xie and Y. Jiang. Stochastic network calculus models under max-plus algebra. In *Proc. IEEE Globecom*, pages 1121–1126, Nov. 2009.

[133] P. Yang, W. Luo, L. Xu, J. Deogun, and Y. Lu. TCP congestion avoidance algorithm identification. In *Proc. IEEE ICDCS*, pages 310–321, May 2011.

[134] L. A. Zadeh. On the identification problem. *Circuit Theory, IRE Transactions on*, 3(4):277–281, Dec. 1956.

[135] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the constancy of Internet path properties. In *Proc. ACM IMW*, pages 197–211, Nov. 2001.

PUBLICATIONS

[1] R. Lübben, M. Fidler, and J. Liebeherr. Stochastic bandwidth estimation in networks with random service. *IEEE/ACM Trans. Netw.*, May 2013. accepted for publication.

[2] R. Lübben and M. Fidler. Non-equilibrium information envelopes and the capacity-delay-error-tradeoff of source coding. In *Proc. IEEE WoWMoM*, June 2012.

[3] R. Lübben and M. Fidler. On the delay performance of block codes for discrete memoryless channels with feedback. In *Proc. IEEE Sarnoff Symposium*, May 2012.

[4] R. Lübben and M. Fidler. Poster: on the capacity delay error tradeoff of source coding. *SIGMETRICS Perform. Eval. Rev.*, 39(2):72–72, September 2011.

[5] R. Lübben, M. Fidler, and J. Liebeherr. A foundation for stochastic bandwidth estimation of networks with random service. In *Proc. IEEE INFOCOM*, pages 1817–1825, April 2011.

[6] R. Lübben, M. Fidler, and J. Liebeherr. A foundation for stochastic bandwidth estimation of networks with random service. *Computing Research Repository*, abs/1008.0050, July 2010. URL `http://arxiv.org/abs/1008.0050`. Extended version of [5].

[7] M. D. Pérez-Guirao, R. Lübben, and T. Kaiser. Autonomous optimization of UWB link access. *Frequenz*, 63(9-10):196–199, October 2009.

[8] R. Lübben, G. Li, D. Wang, R. Doverspike, and X. Fu. Fast rerouting for IP multicast in managed IPTV networks. In *Proc. IEEE IWQoS*, July 2009.

[9] N. Neumann, R. Lübben, M. Arumaithurai, and X. Fu. Decoupling congestion control using traffic aggregates and middleboxes. In *Proc. IEEE ICNP*, October 2008. Poster session.

[10] M. D. Pérez-Guirao, R. Lübben, and T. Kaiser. Pulse rate adaptive multiple-access scheme for cognitive autonomous IR-UWB networks. In *Proc. IEEE ICUWB*, pages 181–185, September 2008.

[11] M. D. Pérez-Guirao, R. Lübben, T. Kaiser, and K. Jobmann. Evolutionary game theoretical approach for IR-UWB sensor networks. In *Proc. IEEE COCONET*, pages 107–111, May 2008.

[12] M. D. Pérez-Guirao, R. Lübben, Z. Zhao, T. Kaiser, and K. Jobmann. Cross-layer mac design for IR-UWB networks. In *Proc. IEEE IWCLD*, pages 113–116, September 2007.

SCIENTIFIC CAREER

Curriculum Vitae	
Name	Lübben, Ralf
Day of birth	January 9th 1981
Education	
since 03.2009	Research assistant and PhD student at the Institute of Communications Technology at the Leibniz Universität Hannover.
01/2008 – 03/2009	Research assistant at the Computer Networks Group at the University of Göttingen.
10/2005 – 11/2007	Studies of computer engineering (M.Sc.) at the Leibniz Universität Hannover.
08/2001 – 07/2005	Studies of electrical engineering and information technology (Dipl.Ing. (FH)) at the FHWT Vechta – Diepholz – Oldenburg.

COLOPHON

This thesis was typeset with $\LaTeX\,2_\varepsilon$ using Hermann Zapf's *Palatino* and *Euler* type faces. The listings are typeset in *Bera Mono*, originally developed by Bitstream, Inc. as "Bitstream Vera".

The typographic style is provided by Andre Miede and has been modified by Michael Bredel. It is inspired by *The Elements of Typographic Style* [23]. It is available for $\LaTeX$ via CTAN as "`classicthesis`".

Final Version as of February 28, 2014.