NETWORK LINK DIMENSIONING

A measurement & modeling based approach

Remco van de Meent

Graduation committee:

Chairman, secretary:	prof.dr.ir. A.J. Mouthaan
Promoters:	prof.dr. M.R.H. Mandjes
	prof.dr.ir. L.J.M. Nieuwenhuis
Assistant promoter:	dr.ir. A. Pras
Members:	prof.dr. J.L. van den Berg
	prof.dr.ir. B.R.H.M. Haverkort
	prof.dr. R.D. van der Mei
	dr. P. Owezarski
	prof. P. Thiran

ISBN 90-365-2305-2
ISSN 1381-3617 (CTIT Ph.D.-thesis series number 06-79)

Centre for Telematics and Information Technology
University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands

NETWORK LINK DIMENSIONING

A MEASUREMENT & MODELING BASED APPROACH

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. W.H.M Zijm,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 24 maart 2006 om 16.45 uur

door
Remco van de Meent
geboren op 18 maart 1977
te Venray

Dit proefschrift is goedgekeurd door:
prof.dr. M.R.H. Mandjes (promotor)
prof.dr.ir. L.J.M. Nieuwenhuis (promotor)
dr.ir. A. Pras (assistent-promotor)

Abstract

Adequate network link dimensioning requires a thorough insight into the interrelationship between: (i) the traffic offered (in terms of the average load, but also its fluctuations), (ii) the desired level of performance, and (iii) the required bandwidth capacity. It is clear that more capacity is needed when the average traffic load becomes higher, the fluctuations become fiercer, or the performance criterion becomes more stringent.

Existing approaches to network link dimensioning are often based on rules of thumb, e.g., 'take the average traffic rate at times when the network is relatively busy, and add 30% to cater for fluctuations'. Clearly, such an approach does not explicitly incorporate the fierceness of the traffic rate's fluctuations, or the desired level of performance.

A common approach to estimate the average traffic rate is as follows. A network manager regularly polls the so-called Interfaces Group MIB via the Simple Network Management Protocol (SNMP), for instance through a tool such as the Multi-Router Traffic Grapher (MRTG). This yields the average rate of the offered traffic since the last poll. The polling interval generally is in the order of 5 minutes. Evidently, the fierceness of fluctuation of the traffic rate within these 5 minute intervals is unknown to the network manager. These fluctuations may, however, be considerably large, and noticeable to users of the network. If, at timescales of say 5 seconds, more traffic is offered to a network link than it can transfer during that interval, traffic may be lost. Such loss is generally known as possibly leading to performance degradation and this may well be noticeable to a network user; for instance, entire words may be lost in a (voice) conversation. Hence, it is in the interest of network users, and for obvious business reasons also to network opera-

tors, to have sufficient bandwidth capacity available to meet the demand at timescales considerably smaller than 5 minutes.

In this thesis, we develop an alternative approach to network link dimensioning, which explicitly incorporates the offered traffic in terms of both the average rate as well as its fluctuations at small timescales, and the desired level of performance. This is expressed through mathematical formulas that give the required bandwidth capacity, given the characteristics of the offered traffic, and the performance criterion.

The characteristics of the offered traffic are described using traffic models. To find accurate traffic models, we have performed hundreds of detailed measurements of real network traffic at five different locations. These locations are chosen to differ in terms of number of users, network access technologies, link speeds, type of users, etc., as to determine which traffic models are usable in a broad range of networking environments.

We find that a Gaussian traffic model, in general, accurately describes real network traffic. A Gaussian traffic model is a model in which the traffic is described as follows: the amount of traffic $A(T)$, offered over an interval of length T is distributed according to the Gaussian (normal) distribution, parameterized through the average $\mathbb{E}A(T)$ and variance $\mathrm{Var}A(T)$. Commonly seen characteristics of Internet traffic, e.g., long-range dependency and self-similarity, fit into the framework of Gaussian traffic modeling.

The performance criterion we use throughout this thesis focuses on achieving 'link transparency': in no more than a fraction ε of intervals of length T, the offered traffic $A(T)$ should exceed the available bandwidth capacity C. In other words: $\mathbb{P}(A(T) \geq CT) \leq \varepsilon$.

For Gaussian traffic, we show that the required bandwidth capacity to meet the above performance criterion can be estimated by the following formula: $C = \mu + 1/T \cdot \sqrt{(-2\log\varepsilon) \cdot \mathrm{Var}A(T)}$, in which μ denotes the average traffic rate.

In this formula, T and ε reflect the specified performance criterion. The average traffic rate μ can be found through the traditional, relatively coarse-grained, SNMP measurements. The fluctuations of the traffic rate, represented by $\mathrm{Var}A(T)$, require measurements at timescale T. As T is expected to be small (say in the order of seconds or smaller, corresponding to the timescale that determines the performance), it seems that relatively fine-grained measurements are required to determine $\mathrm{Var}A(T)$. It is generally

not feasible to perform such measurements using SNMP. Therefore, in this thesis, we propose an alternative approach to estimate $\mathrm{Var}A(T)$.

Our approach to estimate $\mathrm{Var}A(T)$ relies on coarse-grained polling of the occupancy of a buffer in front of the network link that we want to measure (similar to the coarse-grained polling required to estimate μ). Occasional polling of the buffer occupancy yields the empirical distribution function of the buffer occupancy. We derive a formula that 'inverts' the buffer occupancy distribution to $\mathrm{Var}A(T)$. Importantly, we can estimate $\mathrm{Var}A(T)$ without requiring measurements at (small) timescale T.

Our alternative approach to network link dimensioning, especially the formulas for the required bandwidth and the 'inversion' to estimate $\mathrm{Var}A(T)$, is extensively validated through case-studies that make use of the hundreds of measurements of real traffic.

The research presented in this thesis can be used for a variety of network link dimensioning related questions; e.g., it can be used to determine how much bandwidth capacity is needed to cater growth or to meet Service Level Agreements. For example, we envision scenarios where small to medium-sized organizations want to have a 'transparent connection' to the Internet, without wasting resources on bandwidth capacity that is not needed, or to provision Virtual Private Networks to interconnect various office locations of an organization.

Contents

1 Introduction

This chapter presents the background and motivation of our research, poses the research questions to be addressed in this thesis, and outlines the thesis structure.

1.1 Background

The Internet has become a vital element of the modern world. Ranging from leisure activities such as playing online games, to mission-critical applications like email for businesses, modern society increasingly relies on the Internet. Moreover, it is expected that the Internet will play even a bigger role in the future, as it will (further) replace other types of telecommunications. For instance, for telephony we may use VoIP (Voice-over-IP) applications instead of the traditional telephony network, and (digital) television signals may be sent over the Internet as well, instead of via the traditional cable or antenna installation.

In order to support these applications, a 'proper' Internet connection (i.e., the communication channel(s) between the involved parties in a communication session) is required. Various applications may pose different requirements on the Internet connection, for example (see, e.g., [Tan02]):

- streaming (and non-interactive) multimedia, such as television signals, require a guaranteed amount of available bandwidth all the time;
- critical applications like remote surgery require a hard guarantee on the availability of the Internet connection between patient and surgeon;
- interactive, real-time applications, such as VoIP and video conferencing, require almost no delay and delay variation (jitter); and

– other applications, e.g., web browsing, do not pose hard requirements on the Internet connection, but one could say that if the Internet connection is 'not good enough', a user may get dissatisfied because of, for instance, high response times or low throughput of the traffic.

For all Internet applications, it clearly holds that sufficient *bandwidth* should be available, regardless of possible other constraints, to (i) make the application function correctly, and/or (ii) satisfy the user (which are, obviously, not necessarily unrelated goals).

In order to meet these 'performance' requirements on an Internet connection, two approaches seem viable (see, e.g., [Tan02, PvdMM05, ZOS00]). The first approach is implementing protocols to ensure certain service levels. Examples of such techniques are DiffServ [BBC+98] and IntServ [BCS94] (more on this later on in this section). The second approach does not use such network protocols, but rather relies purely on network capacity: the capacity should always be sufficient to satisfy all requests, and hence provide the service level requested. The latter approach is called *overprovisioning*, and is commonly used by network operators: a study by Odlyzko [Odl03] concluded that backbone networks are generally overprovisioned: on average only a small percentage of the available capacity is used[1]. In this thesis we consider methods related to the concept of overprovisioning for meeting the requirements of an Internet connection. The methods we consider, however, aim at achieving a higher utilization of the available capacity.

Overprovisioning has several advantages (see, e.g., [FML+03, FTD03]):

– No need for network systems and network management to support relative (to overprovisioning) complex (and therefore error-prone) techniques to ensure certain service levels;
– Traffic growth is easily catered for;
– Performance is good: no congestion, low latency; and
– Redundancy: if there are two separate physical links between two locations, and both have a utilization under 50%, each link can handle the traffic of both links, in case one of them fails (hence, graceful degradation).

[1] Note that 'average' and 'used' are not further defined in [Odl03]

An argument against overprovisioning is that of the extra costs that are presumably incurred because of the apparent waste of resources, as 'a lot' of the available capacity is not used. When protocols such as DiffServ and IntServ are used, however, a higher grade of utilization of the available resources can be used, compared to an overprovisioning situation: with protocols such as DiffServ and IntServ, all user can be assigned an amount of bandwidth capacity that is sufficient to meet their individual performance levels; in the overprovisioning situation, all users get the 'best' performance (see, e.g., [FTD03]). The low utilization grade of resources has not withdrawn large backbone network operators to apply the overprovisioning concept in their networks. For smaller parties, however, paying for more than 30 times as much transit bandwidth as the average traffic load — referring to the 3% utilization grade mentioned by Odlyzko [Odl03] — may be less attractive.

Although the idea of overprovisioning is simple, still the question remains of *how much* a network operator should overprovision its network to guarantee certain levels of service. Without sufficient overprovisioning, the performance of the network (also from a user's perspective) will drop below the required levels. Too much overprovisioning, however, does not improve performance (again, from the user's perspective) anymore, and hence, may be seen as a waste of resources. This leads to the question what the lowest capacity level is, at which additional capacity does not improve the service level (or: at which capacity level the required service level is achieved).

> **Example:** Consider the following scenario — which is a typical application scenario for the research presented in this thesis — depicted in Figure 1.1. An organization connects to the Internet through a so-called *uplink* to an Internet Service Provider (ISP) (or an ISP connects to a backbone network operator). The ISP takes care of connectivity to the rest of the Internet. Now suppose the organization does not want their uplink to be a limiting factor, i.e., it may not be a *performance bottleneck*, in the Internet communications of its employees (or clients, etc.). To meet that requirement, the bandwidth capacity of the uplink should be sufficiently large. On the other hand, however, the larger the uplink's capacity, the higher the associated costs for the organization will be in general.

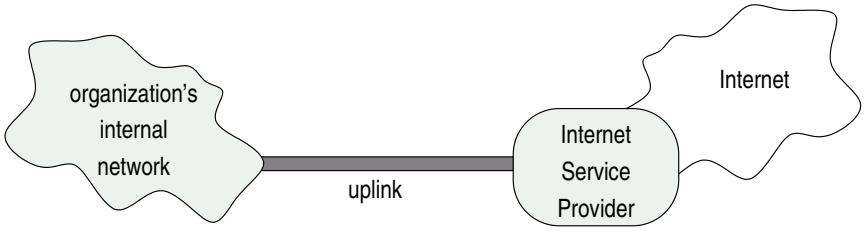


Figure 1.1: An organization is connected to the Internet via an Internet Service Provider, through an uplink

In a situation as in the example above, a tradeoff between capacity (costs) and performance is to be made: performance should be good, but there should be no needless waste of resources. We use the term *performance-aware bandwidth provisioning*, or in short *bandwidth provisioning*, to denote this concept of 'efficient' overprovisioning. Bandwidth provisioning is the topic of this thesis.

Note that, in the context of thesis, when we use the more general term (network) link dimensioning*, we refer to this (performance-aware) bandwidth provisioning idea.*

1.2 Bandwidth provisioning

The tradeoff between network link capacity, and the performance as perceived by users for traffic transmitted over that link, is illustrated in Figure 1.2. When capacity is low, the link will be a performance bottleneck for the users. When capacity is added, the performance as perceived by the users will improve - to a certain extent: at some point, the addition of even more capacity, does not, or just slightly, further improve the perceived performance. When the overprovisioning concept is applied, there is a substantial amount of spare capacity (see the indication in Figure 1.2). Clearly, bandwidth provisioning has the advantage of providing 'good performance', and does not suffer from the very low utilization grade in pure overprovisioning, in that (it is expected that) higher resource utilization grades are achieved with bandwidth provisioning.

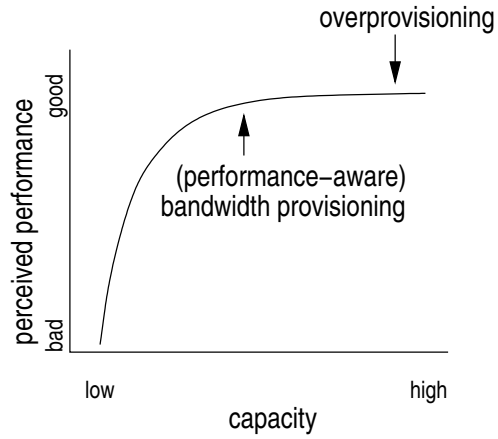


Figure 1.2: Relation between bandwidth capacity and perceived performance.

We define bandwidth provisioning as follows:

> *(Performance-aware) bandwidth provisioning is the procedure to determine the lowest required bandwidth capacity level for a network link, such that for a given traffic load, a desired performance target is met.*

Note that the term 'traffic load' refers to the characteristics of the traffic offered to the network link, e.g., the mean load and variations around this mean load. Clearly, the crucial question in the context of bandwidth provisioning is: *What is the required bandwidth capacity of a network link, given the traffic load and a desired performance target?*

In more formal terms: let A be (a statistical description of) the offered traffic, and p the desired performance target, and denote with φ the function (for simplicity we assume such a function exists) that determines the required capacity, denoted with C. In other words:

$$\varphi : (A, p) \mapsto C.$$

Figure 1.3 illustrates the relation between C, A and p, in a *capacity*, *load* and *performance* triangle. Clearly, these three quantities are related to each other, according to the following dynamics:

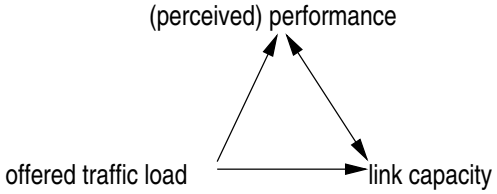


Figure 1.3: The relation between capacity, load and (perceived) performance.

- If the offered traffic load increases, and the capacity is not changed, the perceived performance may deteriorate (and vice versa);
- If the capacity increases, and the traffic load remains unchanged, the perceived performance may improve (and vice versa);
- If the offered traffic load increases, the link capacity needs to be expanded in order to keep the perceived performance unchanged (and vice versa).

Traditionally, the provisioning formula φ that is used to determine the required bandwidth, is merely a rule of thumb. For instance: 'the mean traffic rate, plus a margin of 30%'. Obviously, such a fixed margin is not universally applicable, nor does it guarantee no needless waste of resources on the one hand, and availability of sufficient resources on the other hand. One could, however, say that provisioning formulas typically are of the following generic form:

$$\varphi(A, p) := M + \alpha_p \cdot V \tag{1.1}$$

for (expected) average load M, some factor α_p that reflects the desired performance target p and some error term V to account for fluctuations of the traffic rate (which we also refer to as *burstiness* in this thesis). We get back to this at a later stage.

Let us now further explore the concept of bandwidth provisioning, by looking at the performance p and offered traffic A in more detail.

(Perceived) performance

First, (perceived) performance is a subjective issue. At what level, e.g., simply 'good' or 'bad', the performance of networking is perceived, may vary

from user to user. People who are patient may not be dissatisfied when having to wait a few seconds for a webpage to load, whereas others may be disturbed waiting 'that long'. Such difference in how various people perceive performance makes it difficult, if not impossible, to determine an objective and universal desired p.

Second, people use the Internet for all kind of different applications, e.g., e-mail, file transfer and video-streaming. Various applications may, however, lead to different perceptions of 'good' and 'bad' performance from a user's point of view. For instance, a user may not be worried when an e-mail takes a few seconds to be sent across the world, but the user will not be happy when his video-conferencing application is stuck for the same few seconds because of bandwidth shortage. Thus, a specific performance measure may depend on the actual applications being used. In general, however, the user's perception of the performance will be determined at a timescale in the order of seconds, or even less: one could argue that when browsing the web, a user would like to have 'the network' respond within a second or so after clicking on a hyperlink — if it takes longer, the user may well notice the (extra) delay and interpret such (extra) as 'bad performance'. One could further argue that when it is no human user sitting behind the computer, but computer programs using the network without human action, e.g., for data exchange between nodes in a so-called grid-network, the timescale at which performance is 'perceived' may be even orders of magnitude smaller than one second.

Third, it is often unclear how performance as perceived by a user, precisely relates to network level parameters. For some specific applications, this relation is studied (e.g., voice quality, see [Uni03, Uni96, Uni01]), but there is no general rule that describes the relation between 'user perceived performance' and 'network level performance'. It can be expected, however, that for instance for Voice-over-IP (VoIP) applications, performance is affected at a timescale of tens of milliseconds. As commonly used codecs in VoIP send packets every 20 milliseconds (see, e.g., [MTK02, ZZX01]), link overload situations at such a timescale lead to increased delay or delay variations, or even packet losses, and hence to performance degradation: the perceived performance of the voice conversation decreases. Another application that clearly benefits from good 'network level performance' at rather small timescales, is (multi-user) on-line gaming (see, e.g., [PW02]).

We recognize that there is no single performance criterion p that is accurate for all users and for all possible applications. In this thesis we mainly use a performance criterion p that we call *link transparency*. The idea behind link transparency is as follows: a network link should be 'transparent' to a user of that link, in that it may not have more noticeable (by a user) negative impact on the performance than is inherent to the link's properties (such as distance). We translate our link transparency objective to the following 'network level' performance measure: *the fraction of disjoint (time-)intervals of length T in which the offered traffic exceeds the link capacity, should be below some (small) value ε*. In other words:

$$\mathbb{P}(A(T) \geq CT) \leq \varepsilon. \tag{1.2}$$

Note that the performance criterion can be chosen by a network operator. Also, he can select the exact setting of the performance parameters (i.e., T and ε in the above target). Whereas the above criterion focuses on achieving link transparency, one may, e.g., also think of a criterion that limits the delay or delay variation (jitter) incurred by traffic on the network.

> **Remark:** In this thesis, T and ε are parameters that we set at 'reasonable values' in all examples. If other values were to be chosen (within reasonable bounds), the conclusions that are drawn throughout this thesis remain similar.

We consider a further study of the (precise) relation between network level performance and user perceived performance out of scope for this thesis.

Offered traffic load

We now look at the offered traffic load A in more detail.

A common way to determine the offered load is to poll Interface Group MIB counters via the Simple Network Management Protocol (SNMP) every 5 minutes (or even coarser); this yields the total amount of traffic sent through the network interface over this time-interval, from which the average traffic rate (per second) can be derived. The inherent drawback to this approach, however, is that the characteristics of the traffic within the 5 minute interval remain unknown.

Consider the following illustrative example:

> **Example:** The University of Twente has, at time of writing, a 1 Gbit/s uplink to SURFnet, the Dutch national research and education network. The current bandwidth utilization averaged over one week is around 400 Mbit/s in one direction, and around 250 Mbit/s in the other direction. The peak utilization, averaged over 5 minutes at the busy moments, is around twice the weekly average (in both directions). Detailed analysis of the traffic on the uplink at the busy moments has shown, however, that the 1 Gbit/s is (almost) fully used during several 1 second intervals. Hence, it is likely that the offered traffic load exceeds the available capacity of the uplink, and, thus, the performance as perceived by users is degraded. In other words, the 1 Gbit/s connection may be deemed full, which is a motivation to upgrade — even when the weekly average utilization is only some tens of percents of the current available bandwidth.

An illustration of the differences between longer term average traffic rates, and rates at more detailed timescales, is given in Figures 1.4 and 1.5. In Figure 1.4, the traffic load on an uplink is shown over a 42 hour period (the 'area' corresponds to outgoing traffic, the 'line' to incoming). The averages shown are based on a 5 minute interval. In Figure 1.5, we zoom in to the outgoing traffic in three such 5 minute interval (around 15.30h, as indicated in Figure 1.4) by plotting the traffic loads based on averages over smaller time intervals, down to 100 milliseconds. It can be seen that, whereas the 5 minute averages are around 330 Mbit/s, when zooming in to a 100 millisecond timescale, considerably higher traffic rates of up to 470 Mbit/s are achieved.

This example is illustrative to common practice, in that it shows both how traffic measurements are usually performed, as well as the mismatch between the measured offered load, and the actual traffic rates at timescales that are relevant to a user's perception of performance. Clearly, for bandwidth provisioning to be successful, such mismatches introduced by coarse measurements may not be ignored.

To quantify the fluctuations of the traffic rates at detailed timescales, network operators could resort to fine-grained measurements. A drawback,

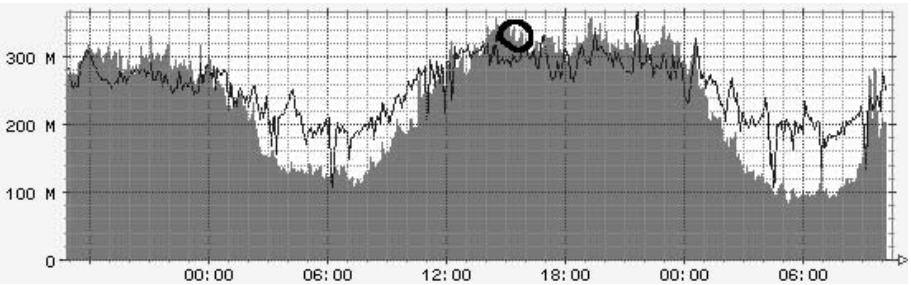
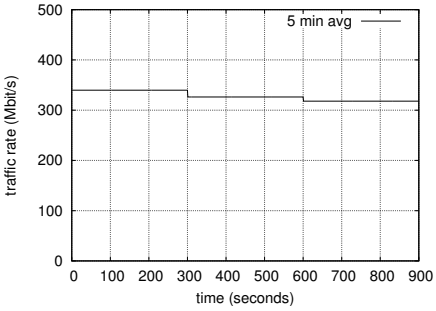
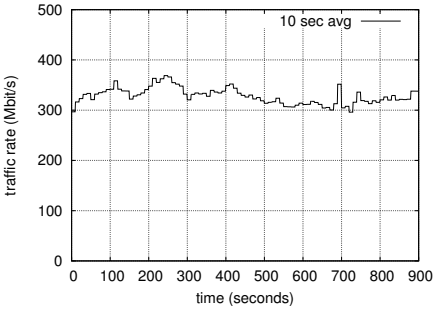


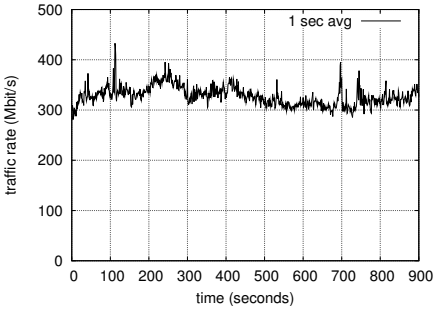
Figure 1.4: Traffic rates based on 5 minute averages



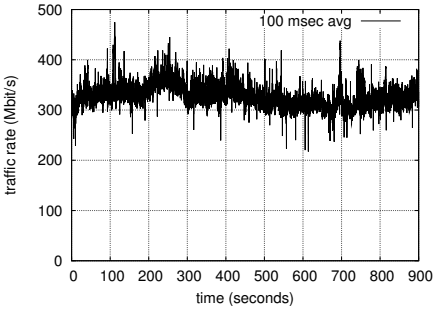
(a) 5 minutes average traffic rates



(b) 10 seconds average traffic rates



(c) 1 second average traffic rates



(d) 100 milliseconds average traffic rates

Figure 1.5: Zooming in: traffic rates at increasingly smaller timescales

however, is that their standard measurement procedure does not support such measurements: SNMP does not give reliable results at very detailed timescales (mainly due to processing time and counter update intervals, see [HvdMP02]).

In this thesis, we develop procedures for bandwidth provisioning, which do not rely on detailed timescale measurements, but still provide (statistical) guarantees at such detailed timescales.

Alternatives to bandwidth provisioning

We have introduced bandwidth provisioning as a concept to provide a desired performance level, as indicated before. Recall from Section 1.1 that there are various alternative approaches, often referred to as Quality of Service (QoS) mechanisms. A QoS-mechanism describes an application or algorithm used to guarantee the quality of the performance of a network. The two main alternative QoS frameworks that have been developed for use in the Internet are:

- *Reservation-based — Integrated Services (IntServ):* In IntServ, QoS is guaranteed by means of reservations through the use of the RSVP protocol. An application can reserve bandwidth by sending the required information to the network routers. If sufficient resources are available to fulfill the reservation along the entire path, this application is then guaranteed the specified number of resources in the network [BCS94] (along a certain path). Note that the reservations have to be kept at each network router along the path, which clearly leads to scalability concerns.
- *Prioritizing — Differentiated Services (DiffServ):* In DiffServ, a byte in the IP header of a packet holds information about the QoS of the packet, and this is used by network routers to differentiate between levels of service: packets belonging to a class with high QoS demands are prioritized over others [BBC$^+$98]. Note that DiffServ offers only 'relative' performance guarantees, whereas IntServ offers 'absolute' guarantees, as no hard reservations are made in DiffServ. Also, one needs to make sure that 'sufficient' resources are available for the various so-called traffic classes associated with the different service levels.

Significant amounts of research and standardization efforts have been put into the development of these QoS frameworks. Little is known about actual deployment of these QoS frameworks in operational IP networks, but it is generally accepted that they are currently not widely used, which may be motivated by the following arguments [FTD03]:

- Bandwidth is abundant in backbone networks. If the utilization of backbone networks is sufficiently low, prioritizing one packet over another does not make much sense as both will get an 'excellent treatment'.
- Network equipment is more complex when QoS has to be provided, compared with situations where no QoS support is required. Extra complexity may lead to higher costs, and is likely more error-prone; and
- Setup and maintenance of QoS support requires higher-trained, and thus more expensive staff.

Therefore, it may not be cost-effective to deploy QoS mechanisms in environments where abundant bandwidth is available. Conversely, one may expect that in situations where bandwidth is scarce, e.g., GPRS or UMTS access networks, QoS mechanisms may be deployed to ensure service levels.

Our bandwidth provisioning approach is related to the aforementioned QoS mechanisms, in that it does not completely replace but rather supports them. Both IntServ as well as DiffServ assume that the network provides a certain amount of bandwidth — but they do not determine how much bandwidth is actually needed for a certain traffic stream. Our bandwidth provisioning approach does exactly that: it yields the required bandwidth capacity to fulfill a desired performance level for a given traffic stream, and this can be used as input to, for instance, the reservation request in an IntServ environment, or to configure the amount of bandwidth required for a certain service level in a DiffServ environment.

1.3 Related work on network link dimensioning

A vast amount of research has been done over the last century in the area of dimensioning of telecommunication networks. In order to put this thesis

into perspective, and to further motivate the need for the research presented in this thesis, this section gives a brief overview of the major developments in the area of dimensioning of telecommunication network links.

In the traditional telephony world, the amount of offered traffic λ is denoted in (the dimensionless unit) Erlang. Because of the circuit-switched nature of those telephony networks, each call is assumed to occupy a fixed amount of bandwidth capacity. In order to provide an adequate performance level, network operators strive to keeping the probability of call blocking (i.e., a user not being able to setup a connection to the remote end) below some threshold ε. Hence, the crucial question in the context of dimensioning is: 'what is the lowest (trunk) capacity C such that the probability that calls are blocked, for given traffic load λ, is below ε?' This basic question has long been answered, as well as for many variants of this model that incorporate features such as 'automatic redialing' (which was not supported by the original model). See, e.g., [Tij94].

Later on, during the 1980s, the notion of multiservice networks emerged. Multiservice networks deliver more services than just voice calls like the traditional telephony networks, e.g., high-speed data transmission, or video conferencing. An example of a multiservice network is B-ISDN (see, e.g., [Onv94]). As various services occupy different amounts of bandwidth capacity (constant within a single call, however), the traditional assumption that each source occupies a fixed amount of bandwidth no longer holds. Therefore, so-called *multirate models* were developed, in which the total amount of offered traffic was decomposed into traffic loads λ_i ($i = 1 \ldots n$) for each service, and associated bandwidth requirements C_i. Clearly, the above dimensioning question can be altered to incorporate the idea of various services — the new target is to ensure that the probability that calls are blocked for the set of all λ_i and C_i is below ε. See, e.g., [Kau81, Rob81, Kel91, RMV96] for more information on multirate models and associated network dimensioning issues.

As it turned out in the late 1980s and early 1990s, the above assumption that sources generate traffic at a constant rate was no longer valid — instead, traffic streams are in general of a *variable bit rate* (VBR) type: the bit rate fluctuates over time [Onv94]. The notion of 'effective bandwidth' (or 'equivalent bandwidth') was introduced to overcome the problem with non-constant rates: it assigns to each traffic stream a number between the

mean and peak rate, reflecting the minimum bandwidth capacity that such a traffic stream requires to meet a certain performance criterion. In VBR models, this performance criterion is often specified at the 'packet level', e.g., the probability that a packet is lost. Following that example, the effective bandwidth, say $C_i(\varepsilon)$, incorporates some (maximum) packet loss probability ε — the lower ε, clearly, the higher $C_i(\varepsilon)$, as more capacity is occupied when the probability of packet loss decreases (i.e., the performance criterion becomes more stringent). This effective bandwidth $C_i(\varepsilon)$ is similar to the bandwidth requirement C_i in the multirate models described above. Therefore, when we assume that traffic of class i occupies $C_i(\varepsilon)$ bandwidth — guaranteeing a maximum loss probability ε at the packet-level — it is possible to compute the (maximum) blocking probability at the call-level (and, hence, to dimension this system such that this blocking probability is below a predefined level). Thus, one could say that two types of performance are offered simultaneously: at the packet-level and at the call-level. See, e.g., [Hui88, GAN91, Kel96, EM93, RMV96] for more information on VBR models and associated network link dimensioning issues. The use of VBR models is not by definition limited to, say, use in ATM networks. VBR models, however, still use the notion of calls and are not designed to deal with very heterogeneous traffic sources. Internet traffic is generally known to be of highly heterogeneous nature (thus, it may be hard to distinguish separate 'classes' of traffic with their own bandwidth requirements), 'operates' with packets instead of circuits, and tends to be extremely unpredictable. Hence, one can not straightforwardly apply VBR models to accurately capture the behavior of Internet traffic.

The fundamentally different nature of the Internet (packet-oriented), compared to the traditional telephony system, as well as ATM (both circuit-oriented), inspired researchers to come up with traffic models, performance criteria (e.g., packet delay or packet loss bounds, instead of call blocking probability) and network dimensioning frameworks that focus primarily on Internet traffic and achieving packet-level performance targets.

Studies that are closely related to the research presented in this thesis, are by Fraleigh [Fra02] and Papagiannaki [Pap03]. In those theses, dimensioning for highly aggregated network links ('backbone links') is studied, with a focus on delay-sensitive applications (such as VoIP). Therefore, they aim at dimensioning such that the delay incurred on the network (due to queuing) does not exceed a certain threshold. The results by Fraleigh and Pa-

pagiannaki are validated using traffic measurements on operational 'backbone' networks. Another dimensioning approach is by Bonald *et al.* [BOR03] — they look at data traffic at the flow-level (in fact, at TCP-connections), aiming at dimensioning such that the throughput rates of TCP-connections remain above a certain threshold. Another dimensioning study in which traffic is modeled at the flow-level is by Barakat *et al.* [BTI+03], who aim at keeping congestion below a certain threshold.

In our research, we primarily aim at achieving *link transparency*, i.e., $\mathbb{P}(A(T) \geq CT) \leq \varepsilon$. This ultimately yields different network link dimensioning rules, which we develop in Chapter 4. For T sufficiently small, our criterion also bounds the delay in the network — in our research, we assume that buffers are used to absorb traffic bursts at timescales smaller than T, thus, the impact of queuing in buffers (which contributes to delay) is also limited to small timescales. Another difference between Fraleigh's and Papagiannaki's research and ours, is that our research is applicable to network links with both low and high aggregation in terms of users, instead of only 'backbone' links. Evidently, fewer users are aggregated at the network links close to the edges of the network, which implies that it is unclear whether traffic models for 'backbone' traffic may still be used; we study this issue in detail in Chapter 3.

1.4 Goal, approach and research questions

The motivation for the research presented in this thesis follows from the discussion in the previous sections, summarized by the following observations:

- To estimate the required bandwidth for a network link, operators rely on information from coarse network traffic measurements, typically in the order of 5 minutes;
- The user of a network experiences performance at more detailed timescales, typically in the order of seconds down to 100 milliseconds or even less; and
- Fluctuations of traffic rates at such detailed timescales are nonnegligible: peak traffic rates can be up to hundreds of percents higher than the 5 minute average traffic rate [vdMPM+03, vdMPM+04].

Goal

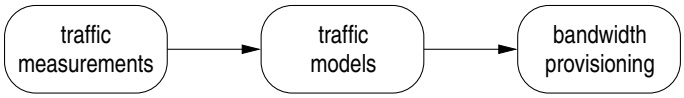
The goal of the research presented in this thesis is:

> *to develop network link dimensioning formulas that determine the minimum bandwidth needed to achieve a certain performance level; these formulas should cater for traffic peaks at small timescales, but not require traffic measurements at such timescales.*

Approach

A network link dimensioning formula (or, equivalent, bandwidth provisioning formula) gives the required bandwidth capacity level to achieve a certain performance criterion, for given input traffic — as was illustrated in Figure 1.3. The input traffic will be described through a model of the real network traffic, hence, we need to find one or more traffic models that accurately describe real traffic. Thus, in order to validate whether a traffic model does accurately describe the real traffic, measurements of real traffic are required.

Consequently, the steps that we take to achieve this objective are illustrated by the following figure, and are further explained below:



One of the novel aspects of the research presented in this thesis, is that it integrates the use of measurements and modeling for network link dimensioning. Other studies often rely on either a purely empirical approach, or just on stochastic models that are not backed up by real traffic measurements.

Research questions

In order to get an in-depth as well as broad understanding of real network traffic, and in line with the identified typical application scenario that we

have in mind (see Figure 1.1 on page 4), we observe the behavior of network traffic on the uplinks of organizations and ISPs. We rely on *measurements* to gain the desired understanding. These measurements are performed on various uplinks of multiple organizations, which have different characteristics in terms of numbers and types of users, link speeds, application usage, etc. By considering these different network environments, we increase the representativeness of our measurements (as opposed to studies that focus on just a single environment), and, hence, broaden the applicability of the results of our research.

The research questions that are associated with this step, are:

(i) How to perform measurements on a (high-speed) network link with the required (detailed) granularity?

Also, as we would ultimately like to prevent the need for such detailed measurements in practice:

(ii) Is it possible to infer detailed information about the traffic characteristics, without relying on detailed measurements?

In order to abstract from individual measurements, each with possible 'incidental behaviors'[2], we aim at finding a statistical Internet traffic model that accurately describes the real traffic. We therefore investigate existing traffic models to find a model that fits our measured traffic. Such models, for example, parameterize the real traffic with a mean traffic rate and a burstiness factor[3] (as in (1.1)).

The research question we pose is:

(iii) Which statistical traffic model(s) describe the traffic we have measured 'good enough' to rely on for use in network link dimensioning?

If we find one or more traffic models that accurately describe the real traffic, we may use these models to derive *provisioning formulas*, i.e., rules that

[2] An individual measurement may or may not be heavily influenced by events that are atypical to the regular network traffic patterns, e.g., a user downloading a DVD-image at high speed.

[3] In this thesis, whenever we mention 'burstiness', we refer to fluctuations of the traffic rates at some, typically small timescale.

estimate the required bandwidth capacity to meet a (prespecified) performance target. For instance, if the model would parameterize the traffic through a mean traffic rate and burstiness, a provisioning formula based on that model should also predict the required capacity with just these two parameters.

The research questions associated with this step are:

(iv) What is an accurate bandwidth provisioning formula for a given traffic model?

If we find the answers to these questions, we have all the ingredients required to successfully achieve the main research goal, as formulated on page 16.

> **Remark:** In order to achieve our main goal of bandwidth provisioning, we have chosen to follow an *inductive* and *empirical approach*, as indicated above. Empiricism is, in the philosophy of science, the concept in which knowledge is derived from experience of the world. According to empiricism, scientific theories are developed (and tested) through experiments and observations. Roger Bacon argued, as early as in the 13th century, that a *scientific method* consists of a repeating cycle of observation, hypothesis and experimentation, as well as the need for independent verification. Bacon's arguments built upon Aristotle's portrait of induction [Tha01]. Another important contributor to empiricism is Karl Popper, who argued that no number of positive outcomes at the level of experimental testing can confirm a scientific theory; a single genuine counter-instance, however, suffices to prove a scientific theory false. This is also known as the asymmetry between verification and falsification [Pop71].

1.5 Organization

The structure of this thesis generally follows the steps in our approach identified in the previous section. Therefore, the remainder of the thesis is organized as follows:

- *Chapter 2 (Internet traffic measurements)* provides an overview of measurement technologies, and presents the measurements that we have done as part of our research. Chapter 2 addresses research question (i).
- We investigate existing statistical Internet traffic models, such as traffic processes described by M/G/∞ input models or Gaussian arrivals, in *Chapter 3 (Traffic modeling)*. We aim at finding one or more models that accurately describe the real traffic as we have measured on various uplinks, thereby addressing research question (iii).
- In *Chapter 4 (Bandwidth provisioning rules)* we derive various formulas for bandwidth provisioning under different modeling assumptions. These formulas are of type (1.1), and are validated using measurements from various uplinks. It turns out that there are various formulas that estimate the required capacity. Chapter 4 addresses research question (iv).
- The bandwidth provisioning formulas derived in Chapter 4 suffer from the inherent drawback that they require an occasional detailed estimation of the traffic burstiness, i.e., parameter V in (1.1), which is rather involved in practice with regard to the required detailed measurements. In *Chapter 5 (Burstiness estimation)*, we develop a new approach to estimate V that eliminates the need for detailed measurements — addressing research question (ii).
- Using the bandwidth provisioning rules from Chapter 4 and the new 'indirect' approach to estimate burstiness from Chapter 5, we develop, showcase and validate (with the help of a prototype implementation) our bandwidth provisioning approach in *Chapter 6 (Validation)*. The validation studies in Chapter 6 also address research question (iv).
- *Chapter 7 (Conclusions)* summarizes the conclusions drawn in this thesis, and identifies possible directions for further work.

The structure of the thesis is depicted in Figure 1.6.

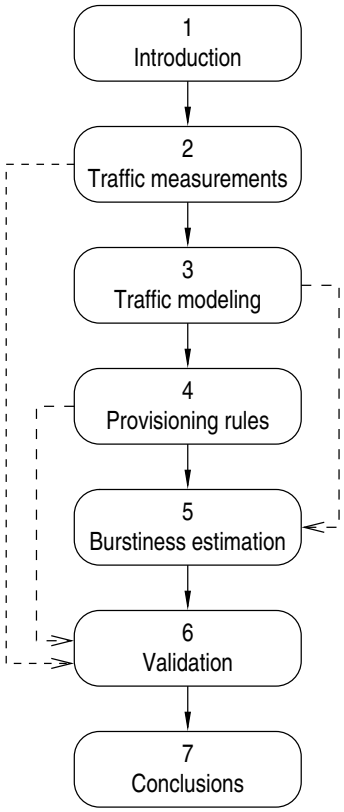


Figure 1.6: Structure of this thesis

1.6 Reading guide

The research presented in this thesis stands midway between the areas of traffic measurement procedures, traffic modeling and queuing theory, in that we combine research in these areas to achieve our objectives. An introduction into traffic measurement procedures is given in Chapter 2, relevant traffic modeling theory in Chapter 3 and relevant queuing theory in Chapter 5. In this thesis it is assumed that the reader is to some extent familiar with TCP/IP.

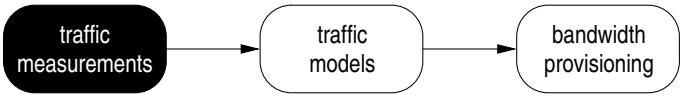
This thesis is intended for those who are interested in network link dimensioning, traffic measurements, and traffic modeling. Chapters 2, 6 and 7 are written from a more practical perspective, whereas Chapters 3, 4 and 5 of

this thesis are of a more theoretical nature, although our objective has been to keep the text accessible to a broad audience.

Parts of the following papers that are coauthored by the author of this thesis, are used in this thesis. Chapters 1 and 2 make use of [vdMPM+03]. Chapter 3 is partly based on [vdMM05] and [vdMMP06]. Chapter 4 is based on [vdBMvdM+06] and [vdMPM+04]. Chapters 5 and 6 extend the results from [MvdM05].

2 Internet traffic measurements

In Chapter 1 we introduced the concept of 'bandwidth provisioning'. Bandwidth provisioning ultimately relies on accurate information about the offered traffic. Such information can be collected through network traffic measurements, and subsequently captured in a 'traffic model', based on which the bandwidth provisioning can be done. These steps are illustrated in the picture below:



This chapter has two separate objectives, namely a discussion of existing Internet measurement technologies including the relation of the research presented in this thesis with these technologies, and an overview of the measurements we have done in this research. The organization of the present chapter is as follows:

- *Section 2.1 introduces the broad area of Internet measurements.*
- *Section 2.2 gives an overview of existing measurement technologies, and relates our research to these existing measurement technologies. Note that we elaborate only on those existing measurement technologies that are relevant to the research presented in this thesis. There are many more things to measure about the Internet (than we need for our research), and thus there are many more measurement technologies; we consider the latter technologies, however, out of scope for this thesis.*
- *Section 2.3 describes the measurements that we have performed in our research.*

– *Section 2.4 presents some traffic characteristics that result from our measurements, which give further insight into the measurements and their environments that we used in our research.*

2.1 Introduction

Internet measurements are used in practice for a variety of purposes, e.g., collecting usage information for charging and billing, detecting errors on networks, and performance monitoring. In this section, we give a list of examples that further illustrate the rationale behind Internet measurements. Note, however, that our intention is not to give an exhaustive list: there are many more reasons for performing measurements.

Internet measurements may be seen as part of network management, as they support the operation of a network. The following examples are structured according to the five so-called 'functional areas' of OSI management, i.e., FCAPS (a contraction of the initial letters of the functional areas) [Int89, Pra95], as follows:

– *Fault management,* i.e., the set of facilities that enables the detection, isolation and correction of abnormal operation. Measurements may help an operator in finding a faulty network link, for instance through the observation of counters that keep track of the number of faulty packets.
– *Configuration management* involves, among other things, the modification of network parameters, for instance link capacity assignments. Traffic measurements are supportive to a network manager who has to make a decision on how much capacity should be assigned to a certain link: he may, for instance, monitor counters that keep track of the traffic volume over a network link (see performance management, below), and use the derived traffic rates to determine the required capacity to handle the offered traffic.
– *Accounting management,* i.e., the set of facilities that allow for establishment of charges and cost identification for the use of network re-

sources such as transfer of data. Clearly, measurements are an important instrument to determine usage, and hence cost.

- *Performance management* is needed to maintain and optimize Quality of Service in a network. For instance, a network manager may keep records of packet loss through periodical monitoring of counters (i.e., measurements). Such records are called 'performance logs', and may be used for other purposes than just performance management as well (see, e.g.,, configuration management).
- *Security management* is related to traffic measurements in that measurements can be used to identify anomalies in the behavior of a network, e.g., the detection of so-called 'Denial of Service' attacks. The information that is collected through traffic measurements can subsequently be used for defense as well (say, by blocking specific traffic sources) [BP01, QZCZ02, HvdM05].

Clearly, Internet measurements are used for a wide variety of purposes. There are two fundamentally different approaches to Internet measurements, viz., *passive* and *active* measurements [BL01, CDF+00, SF01]:

- *passive measurements* are carried out by observing normal network traffic, such that this traffic is not perturbed. A common application of passive measurement is counting the number of packets and bytes traveling through Internet routers.
- *active measurements*, on the other hand, involves sending test traffic into the network. For instance, the round-trip-delay between two hosts can be measured using the PING tool, which sends out a sequence of IP packets that are echoed by the destination.

In this thesis on the topic of bandwidth provisioning, *workload characterization* (also referred to as *bandwidth utilization*) is important. The research presented in this thesis is supported by numerous passive measurements, to characterize the workload on various operational (i.e., carrying real traffic) network links — we will elaborate on this in Sections 2.3 and 2.4. We regard the workload of a network link equal to the amount of traffic (in terms of the mean rate, but also its fluctuations) offered to this link (this, indeed, neglects effects of possible buffering or loss).

The (passive) measurement of Internet traffic typically is a two-step process: (i) the observation of traffic on an Internet network link, and (ii) the inference of information from such observations. An example to illustrate this division in two steps is given below:

> **Example:** Bandwidth utilization graphs such as generated by the Multi-Router Traffic Grapher (MRTG) [Oet03] tool, are made in 2 steps. First, the amount of traffic that is sent through a network interface is tracked internally by an Internet router and represented in the Interfaces Group MIB (IF-MIB) [MK00]. Second, a network manager (e.g., the MRTG tool) polls this IF-MIB every (for instance) 5 minutes, computes the 5 minute average utilization and plots this in a graph over time.

After this broad introduction, in the next section we discuss existing Internet traffic measurement technologies.

2.2 Existing Internet traffic measurement technologies

Internet traffic measurements are widely used for a variety of purposes. One of such purposes is bandwidth provisioning, the topic of this thesis. In this section we give an overview of Internet traffic measurement technologies that are relevant for bandwidth provisioning. Even with this limited scope, there are a number of standards and tools that are relevant in the area of bandwidth provisioning. A classification of them can be done in various ways; we choose to order them by the granularity of the information on the measured Internet traffic they provide. First, however, we give a conceptual overview that fits all Internet traffic measurement technologies that will be discussed, and make some remarks on standardization.

2.2.1 Conceptual overview of the Internet traffic measurement process

Figure 2.1 illustrates how an Internet traffic measurement process generally works. An observation point observes individual packets on some network link. These packets are subsequently captured, which results in packet records that contain (part of) the captured packets. From these packet

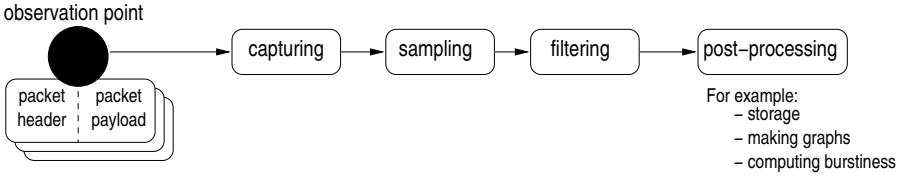


Figure 2.1: Generic Internet traffic measurement process

records, a sample is taken (possibly a trivial 1:1 sample, i.e., 'sampling everything'), and filtering is applied to remove unwanted records (might actually be 'no filtering'). The remaining packet records may now be further processed to ultimately infer the desired information (e.g., bandwidth utilization). Such further processing may also involve the storage of information on disk, e.g., 'tcpdump' [Law05b] packet capture files. Alternatively, the filtered packet records may be input to some flow classification process. A flow consists of (somehow) related packets (see Chapter 3 for a more elaborate discussion on flows). For instance, all packets belonging to a specific TCP connection may constitute a single flow. The flow classification process yields flow records, which may subsequently be sampled and filtered (analogous to the corresponding packet records functions), before further processing is done.

In short, the output of packet level processing are packet records. Tools for packet capturing and further processing interact by using common formats for packet records. The *'pcap' format* [Law05a] is a well-known example of a common format. Records can be exchanged between the various elements of the Internet traffic measurement process using Application Programming Interfaces (APIs), packet record files, packet records exchange protocols or packet records in a database. Similar considerations apply to flow records.

2.2.2 Standardization bodies and other organizations

Standardization is required for interoperability between components and tools in Internet traffic measurement. Standardization may include not only (for instance) the packet record format, but also configuration and control of the measurement process, the measurement process itself, the transmission of the measured data, etc. The obvious standardization body for Internet traffic measurements is the Internet Engineering Task Force (IETF),

Acronym	Name	URL
3GPP	3rd Generation Partnerships Project	http://www.3gpp.org
GGF	Global Grid Forum	http://www.ggf.org
IETF	Internet Engineering Task Force	http://www.ietf.org
IPDR	IPDR.org	http://www.ipdr.org
IRTF	Internet Research Task Force	http://www.irtf.org
ITU	International Telecommunication Union	http://www.itu.int

Table 2.1: Standardization bodies for Internet traffic measurements

Acronym	URL
ANS	http://www.advanced.org
CAIDA	http://www.caida.org
NLANR	http://www.nlanr.net
RIPE NCC	http://www.ripe.net/ripencc/mem-services/ttm/
SPRINT ATL	http://www.sprintlabs.com/
WAND	http://wand.cs.waikato.ac.nz

Table 2.2: Organizations and groups working on Internet traffic measurements

which develops Internet related standards and protocols. There are, however, also several other standardization bodies with activities in the area of Internet traffic measurements. They are listed in Table 2.1.

There are also a number of organizations or groups that focus on various aspects of Internet traffic measurements. We do not discuss these activities extensively here, as they are only to a limited extent related to our usage of Internet traffic measurements, i.e., bandwidth provisioning through traffic measurements and modeling. A non-conclusive list of organizations and groups is given in Table 2.2.

Note that for the research presented in this thesis, no individual organizations or institutes are of specific relevance, in that they do not specifically address the bandwidth provisioning issue. Therefore, we only refer to the URLs given in the tables for more information on these organizations and institutes. In Sections 2.2.3 — 2.2.5 relevant (passive) measurement technologies are discussed, some of them developed by the organizations and institutes mentioned above. In this discussion, a distinction is made between three levels, or granularities, of information, viz.:

– technologies that give information *per (IP) packet*;

– technologies that give information *per flow*, i.e., a string of related packets (see Chapter 3 for a more precise definition), such as all packets in a single TCP connection; and

– technologies that give *high-level*, information, such as the number of packets flowing in or out a network interface card.

2.2.3 Measurement technologies: per-packet information

Packet-level measurement technologies are developed to collect detailed information about each packet transmitted over a single network link.

A simple technology like that is a computer running the *libpcap/tcpdump* combination [Law05b] connected to a Ethernet link that carries the traffic that is to be measured. *libpcap/tcpdump* records the packets[1] received on the computer's network interface, together with the time at which each packet was received. The *libpcap/tcpdump* measurement technology has been used in various measurement studies, e.g., [PF95].

The packet records collected by the *libpcap/tcpdump* combination may be further analyzed in real time, but can also be stored to disk for later post-processing as so called *packet traces*. These are binary format packet records in the 'pcap' format.

Existing or newly developed tools can be used to access and process these traces. For instance, the *ethereal* tool [eth04] provides a (graphical) user interface to perform all kind of post-processing tasks, varying from bandwidth utilization computations to Voice-over-IP analyses, and more.

A shortcoming of *libpcap/tcpdump* based measurement systems is that the packet timestamps are only as accurate as the system clock of the computer that is used to collect the measurements. Depending on the accuracy of this clock, as well as the objective of the measurement efforts, this may or may not be a problem. With current state of the art software and hardware (off-the-shelf PCs with Linux 2.6 kernels, for example), the accuracy of the system clock is in the order of tens of microseconds.

[1] Depending on the measurement parameters, this can be limited to just the first n octets of a packet, for instance corresponding to everything 'up to' the transport layer headers; also a filter may be applied to, for example, only record packets from a particular source.

Next to this clock precision issue, it may also be required that clocks are synchronized between various installations of the measurement system (in order to compare measurements, for instance to follow a packet traversing the network). The standard protocol for network-based time synchronization, the Network Time Protocol (NTP), gives accuracy in the order of tens of milliseconds, which may not be good enough. In that case, dedicated clock synchronization hardware or software may be needed, see, e.g., [Nie91].

There is yet another uncertainty in the timestamp: the timestamp is recorded when the operating system receives the packet from the network interface — not the time at which the packet was observed on the network link. The packet spends 'some' time in the buffer of the network interface card, before the operating system's interrupt handler services the packet. The exact delay is unknown, but is generally accepted as being smaller than the timescales we aim at in the context of this thesis.

Based on *libpcap*/*tcpdump*, AT&T developed a measurement system called PacketScope [ACD+97]. Others developed alternative packet level measurement technologies, in which the packet timestamp resolution was improved by using hardware that timestamps the packets immediately when they are received at the network interface card. Examples of such systems are OC3MON, OC12MON [ACTW96] and IPMON [FML+03].

A final initiative worth mentioning is the IETF's *psamp* working group [iet04b], which strives to develop a standard way of performing packet-level sampling during measurements.

2.2.4 Measurement technologies: per-flow information

The big advantage of packet-level measurements, as discussed above, is that those technologies give detailed per-packet information. The disadvantage however, is that this results in a vast amount of measurement data. As it may not be necessary to capture traffic with such great detail, one might resort to flow-level measurement technologies, which only record information about each flow. Definitions of what a flow is may differ, but generally they are seen as strings of related packets, for instance TCP connections.

The information collected about each flow typically includes the start- and end-time of the flow, the number of packets transmitted in the flow, and the

size of the flow. Note that this amounts to much less information than is collected in packet-level measurement systems.

Examples of flow-level measurement systems include NeTraMet [Bro02] (an implementation of the IETF's real-time traffic flow measurement system [BMR99]), and Netflow [net04], a feature built in to many Cisco routers, which allows these routers to export flow information to external flow-data collection systems. The IETF started a working group in 2001, ipfix [iet04a], to standardize this information export process.

More detailed information on network traffic flows (in general) is given in Chapter 3 of this thesis.

2.2.5 Measurement technologies: high-level information

Whereas flow-level measurement technologies already abstract from per-packet information, one can even go one step further: providing high-level information on all traffic that traverses a network link.

The arguably most widely used traffic measurement system on the Internet is of this type: in the *IF-MIB* (Interface Group MIB) [MK00], counters are kept of, e.g., the total amount of data sent through a network interface. Tools such as MRTG (Multi Router Traffic Grapher) [Oet03] make use of these counters to generate average traffic throughput graphs like Figure 1.4 on page 10.

Related to the IF-MIB is the RMONMIB [Wal00], which provides a bit more detailed information on the traffic on a network link (such as counters of how many packets in a certain size-range have passed), but this is still relatively high-level information, compared to the level of detail provided by per-flow or per-packet measurement technologies.

2.2.6 Traffic measurements for bandwidth provisioning

In the previous subsections we have discussed numerous technologies for Internet traffic measurements. When discussing actual practices with network operators, it is often admitted that their bandwidth provisioning policies are based on rules of thumb. For instance, the 5 minute average bandwidth utilization is measured through SNMP (e.g., with help of MRTG), and

some margin for fluctuations at smaller timescales is added, yielding the estimated required bandwidth capacity. Thus, despite the availability of more sophisticated, and arguably more complicated, measurement technologies, operators often resort to simple tools and technologies. Hence, it would be useful to accurately estimate the required bandwidth capacity, and still only rely on simple tools and technologies. In the course of this thesis we will develop such an approach.

2.3 Traffic measurements in this study

We have performed numerous measurements on five distinct networks to support our study. These networks, discussed later on in this section, cover different network technologies, have a varying number of users as well as different types of users and applications. Also various link speeds are covered by these different networks. All in all, we believe that a broad range of realistic networking scenarios is covered by these locations, as to enlarge the applicability of our entire study.

The purpose of the present section is to describe our measurements. In Section 2.3.1 the measurement procedure followed is discussed, and Section 2.3.2 gives an introduction to the networks ('locations') on which we have performed measurements. The various locations have different characteristics, in terms of, e.g., number of users, access link speeds and type of users, in order to capture a broad range of realistic scenarios.

The next section, 2.4, provides several analyses of the measurements, to provide some fundamental characteristics of the measured network traffic, and to illustrate the diversity of the networks. We discuss the average traffic rates at each location, and actual traffic rates measured at small timescales — as we have seen in Chapter 1, there are 'spikes' in the traffic rates at small timescales, that presumably strongly influence the user's perceived performance.

2.3.1 Measurement procedure

We measure at so-called *uplinks* of the various networks. An uplink is the communication link between the local network (or 'access network') of an

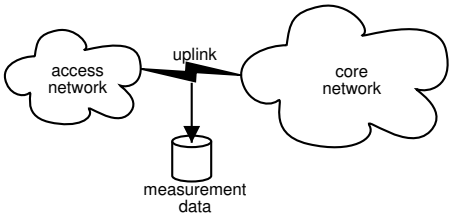


Figure 2.2: Conceptual measurement setup

organization and its 'upstream' network provider (often referred to as ISP, Internet Service Provider; referred to as 'core network'). An uplink carries the network traffic of that organization that does not stay on the local network (so-called 'transit traffic'). For instance, when a user from within that organization visits the website of another company, the associated traffic typically does not stay on the local network, and thus is transit traffic, transmitted over the uplink. Note that an uplink may carry both downstream as well as upstream traffic (this is not required, however — there may be different links for each direction).

Our measurement approach is to connect a PC running the GNU/Linux operating system with *libpcap/tcpdump*, to a network router or switch that is connected to the uplink of the organization whose link is measured. The router or switch is then configured by its network manager to electronically copy (sometimes referred to as 'mirroring') all transit traffic (in both directions) of that organization to our measurement PC. This PC receives, through a Gigabit Ethernet network interface card and the *libpcap/tcpdump* tool, the traffic, and timestamps it upon reception by the operating system. The accuracy of this timestamp (see also the discussion in Section 2.2.3) is deemed 'good enough' for our ultimate goal of bandwidth provisioning, which — in our opinion — plays at a timescale of milliseconds or larger.

libpcap/tcpdump is configured to record only the first 64 octets of every frame in packet traces, ensuring that we capture everything up to the transport layer header (this includes, e.g., TCP headers), which is sufficient for our post-processing purposes in the context of bandwidth provisioning. See Figure 2.2 for a conceptual overview of the above described measurement setup.

Component	Specification
CPU	Pentium-III 1 GHz
Mainboard	Asus CUR-DLS (64 bit 66 MHz PCI)
Hard disk	multiple, >100 Gigabyte, UDMA/66
Operating system	Debian GNU/Linux, kernel 2.4 / 2.6
Network interface	1 x Gbit/s Intel Pro/1000T
Main memory	512 MB reg. SDRAM

Table 2.3: Measurement PC Configuration

Our measurements generally consist of 15 minute windows in which all traffic is captured. At the five locations, in total, some 850 packet traces were collected. The individual packet trace sizes range from a few megabytes (per 15 minutes) to a few gigabytes, depending on the utilization of the network. In total, some 500 gigabytes of traces were collected. These were made anonymous and are available online from the following URL:

`http://traffic-repository.ewi.utwente.nl/`

The configuration of our measurement PC is outlined in Table 2.3.

The post-processing part of the measurements is mostly handled by tools we developed ourselves, in C, Perl and other programming languages. These traces access the repository, analyze the trace and ultimately yield the desired information, depending on our analysis objective.

2.3.2 Measurement locations

As outlined in the beginning of the present section, we have performed measurements at 5 different locations, covering a broad range of networking scenarios. Table 2.4 provides a short overview of these locations, on which we will elaborate below. Note that in the remainder of this thesis we will use the letters *U*, *R*, *C*, *A*, and *S* (see Table 2.4) to denote these locations. For reasons of confidentiality, we cannot disclose the real names of the organizations whose uplinks we have performed measurements on.

location	short description	no. of traces
U	university residential network	15
R	research institute	185
C	college network	302
A	ADSL access network	147
S	server-hosting provider	201

Table 2.4: Measurement locations

Location #1: *U*

On location *U* the 300 Mbit/s (a trunk of 3 x 100 Mbit/s) Ethernet link has been measured, which connects a residential network of a university to the core network of this university. On the residential network, about 2000 students are connected, each having a 100 Mbit/s Ethernet access link. The residential network itself consists of 100 and 300 Mbit/s links to the various switches, depending on the aggregation level. The measured link has an average load of about 60%. Measurements have taken place in July 2002.

Location #2: *R*

On location *R*, the 1 Gbit/s Ethernet link connecting a research institute to the Dutch academic and research network has been measured. There are about 200 researchers and support staff working at this institute. They all have a 100 Mbit/s access link, and the core network of the institute consists of 1 Gbit/s links. The measured link is only mildly loaded, usually around 1%. The measurements are from May — August 2003.

Location #3: *C*

Location *C* is a large college. Their 1 Gbit/s link (i.e., the link that has been measured) to the Dutch academic and research network carries traffic for over 1000 students and staff concurrently (during busy hours). The access link speed on this network is, in general, 100 Mbit/s. The average load on the 1 Gbit/s link usually is around 10-15%. These measurements have been done from September — December 2003.

Location #4: *A*

On location *A*, the 1 Gbit/s aggregated uplink of an ADSL access network has been monitored. A couple of hundred ADSL customers, mostly student dorms, are connected to this access network. Access link speeds vary from 256 kbit/s (down and up) to 8 Mbit/s (down) and 1 Mbit/s (up). The average load on the aggregated uplink is around 150 Mbit/s. These measurements are from February — July 2004.

Location #5: *S*

Location *S* is a hosting-provider, i.e., a commercial party that offers floor- and rack-space to clients who want to connect, for example, their WWW-servers to the Internet. At this hosting-provider, these servers are connected at (in most cases) 100 Mbit/s to the core network of the provider. The bandwidth capacity level of this hosting-provider's uplink (that we have measured) is around 50 Mbit/s. The measurements at this location have been done from December 2003 — February 2004.

2.3.3 Privacy issues with traffic measurements

As an aside, we would like to address some privacy issues related to traffic measurements.

Generally, one could say that the more detail in the measurements, the more privacy issues may need to be addressed. When only looking at high-level information, especially when the associated network link carries traffic from many users, there is little privacy-sensitive information to deduce from that high-level data. On the other hand, when individual packets are collected, it clearly is possible to find out which user visited what websites, for instance. This is because the packets that are collected contain the source and destination IP addresses, generally traceable to individual users and servers, and depending on the measurement settings, even the 'payload' of packets can be analyzed (for instance, to read email that is exchanged between users).

In our research, we have decided to make the packet traces anonymous, in the sense that it will not be possible to find out which specific user visited what website. Tools such as tcpdpriv [Ips97] or tcpurify [Bla02] can be used

to achieve this, by, e.g., randomizing the IP addresses. We also did not capture the 'payload' of the observed packets, but rather the headers (up to the transport layer).

2.4 Traffic measurements in this study: general results

In this section we present some general results that we obtained from analysis of our measurement data. The objective of the analyses presented here is to 'roughly' characterize the traffic on the uplink to the various networks. Importantly, it highlights the variety in network characteristics in terms of workload and user applications.

2.4.1 Average workload

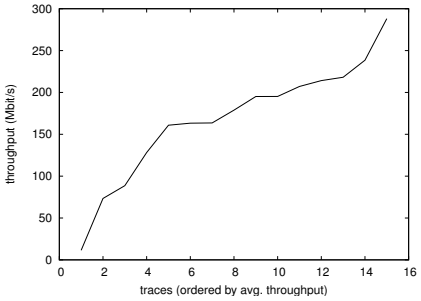
The average workload M (in this thesis used interchangeably with the terms mean load, mean rate and average throughput) denotes the average traffic offered per time interval, usually expressed in Mbit/s (i.e., million bits per second).

Let $A(s,t)$ be the amount of traffic offered in time interval $[s,t]$, with $s < t$, expressed in bits. Then the average workload in $[s,t]$ is given by

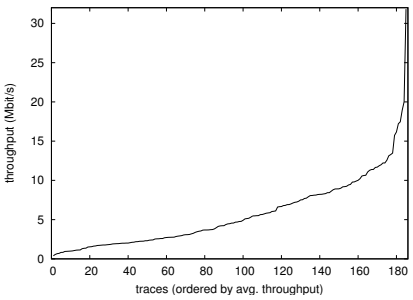
$$M = \frac{A(s,t)}{t-s}.$$

In order to get a rough idea of the workload at a particular location, we average over the mean load of all measurements (each representing 15 minutes of observation) at that specific location (also see Table 2.4). The resulting 'mean average workloads' are denoted with $M^\star$, and listed in Table 2.5 on page 39.

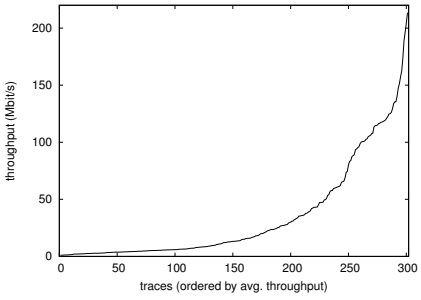
Table 2.5 also lists the standard deviation and 95th percentile of the average workloads at each location. These are an indication of the spread of the workloads at each location (i.e., 'usually the same workload' or not) and how the workload is at 'busy times' ('busy times' or 'busy hours' is referred to when those times of the day when the most traffic is transmitted are meant), respectively. The measurements have been performed at various times of the day and night, but more or less evenly spread. Note that the numbers in



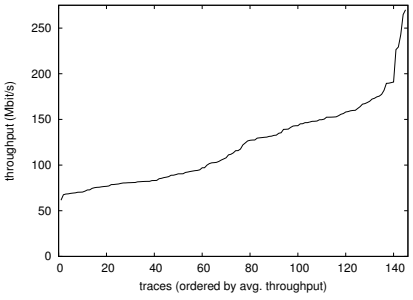
(a) Location U



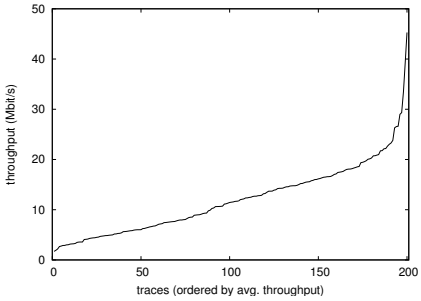
(b) Location R



(c) Location C



(d) Location A



(e) Location S

Figure 2.3: Distribution of the average bandwidth utilization figures, across the various measurement locations

Loc.	Avg. load $M^\star$ (Mbit/s)	Std. error (Mbit/s)	95th percentile (Mbit/s)
U	168.3	70.0	288.1
R	5.7	4.5	13.1
C	34.9	44.5	125.1
A	120.4	42.7	189.7
S	12.0	7.0	23.0

Table 2.5: Average bandwidth utilization at our measurement locations

Table 2.5 represent characteristics *between* various traces at each location, not characteristics *within* specific traces.

> **Remark:** It is often noted in traffic measurement studies that the offered traffic exhibits a 24 hour periodic, in that (for an office environment) the traffic volume increases when people start working, peaks at some hours during the work-day, and decreases in the evening and night. This phenomenon can also be seen in Figure 1.4 on page 10.
>
> The detailed measurements that we have performed all have duration of 15 minutes, and hence, do not exhibit this 24 hour periodic characteristic. A noteworthy advantage of such 'short duration' measurements is, however, that the traffic may be assumed 'stationary' in the measurement interval, in that the mean load, averaged over sufficient long time intervals, is more or less constant. More on this follows in the next subsection.

Figure 2.3 graphically shows the distribution of the measured average throughput at each measurement location. On the horizontal axes, the individual measurements are listed, ordered by the average work load. The vertical axes represent the actual throughput. The shape of the graph is an indication of 'how common' a measured workload is. For instance, at location R a (relatively) high workload is rare, whereas this is not that exceptional at location A.

2.4.2 Throughput at various timescales

Common practice is to measure the average workload over time intervals in the order of minutes. Figure 1.5 showed that at smaller timescales, the fluc-

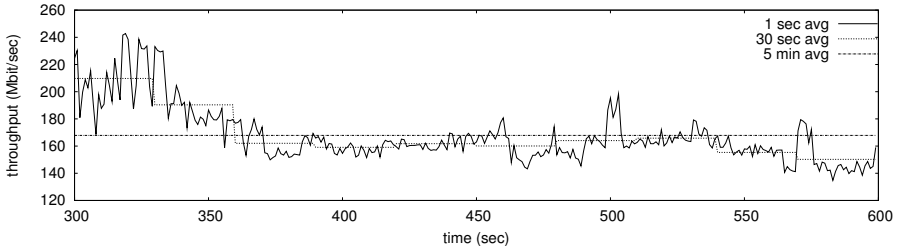


Figure 2.4: Traffic rates at location U for T = 5 minutes, 30 seconds and 1 second

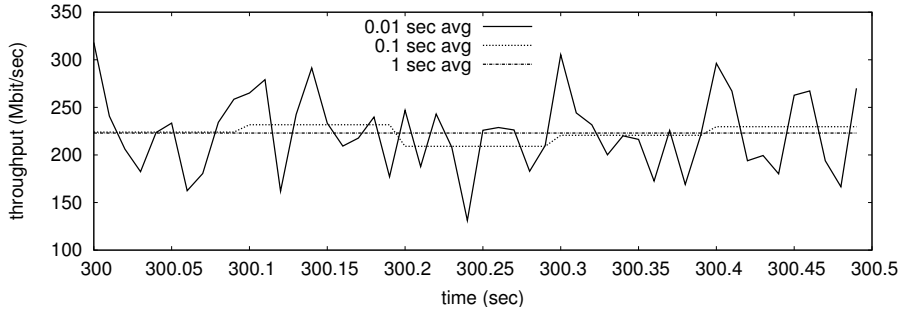


Figure 2.5: Traffic rates at location U for T = 1 second, 100 milliseconds and 10 milliseconds

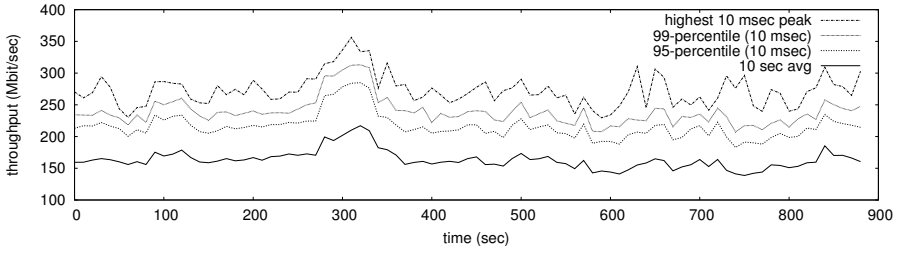


Figure 2.6: Traffic rates at location U: percentiles

tuations (i.e., burstiness) around the average workload figure may be significant. We will now further quantitatively investigate these fluctuations.

Figure 2.4 shows, for a trace taken from location U , the difference between common 'MRTG statistics' (i.e., 5 minute averages) and measurements at smaller timescales. The time-granularity T is increased from 5 minute throughput averages, to 30 second averages and finally 1 second averages. Note that the 5 minute average is around 170 Mbit/s. From the picture it is clear that, within that interval, the average throughput in the first minute is considerably higher than the 5 minute average. This is true for both the 30 seconds as well as the 1 second averages. Some of the measured 1 second average throughput values are even 40% higher than the traditional 5 minute average value. It should be noted that all measurements span 15 minutes; for visualization reasons, the graphs show only part of that interval.

Figure 2.5 zooms in on the first half second of the measurement of Figure 2.4. Time-granularity is further increased from 1 second, to 100 ms and finally 10 ms. Note that each 10 millisecond interval still contains hundreds of packets. The graph shows that the 100 ms averages are relatively close to the 1 second average throughput. This is not a general rule, however; other measurements on the same network have shown differences of up to tens of percents. It is interesting to see spikes of over 300 Mbit/s for the 10 ms averages — almost twice the value of the 5-minute average.

To gain even more insight into the fluctuations at small timescales, Figure 2.6 compares the 10 second average throughput values of a same example trace as above, with the 10 millisecond averages, by plotting the highest 10 millisecond figure within each interval of 10 seconds, as well as the 95th and 99th percentile of the 10 millisecond averages. Clearly, it is 'not uncommon' that traffic rates at small timescales are considerably higher than at larger timescales, which is, again, an important thing to keep in mind for bandwidth provisioning.

Similar figures can be made for the other measurement locations. At a later stage in this thesis, we will use detailed workload figures for the other locations as well.

2.5 Concluding remarks

In this chapter we have first outlined how Internet traffic measurements are, generally, organized (Section 2.1). Second (Section 2.2), we have given an overview of existing measurement technologies, which are somehow related to the concept of bandwidth provisioning.

Let us recall from Chapter 1 that for bandwidth provisioning, there are two parameters to be found through measurements: mean load and some notion of burstiness. An important conclusion of the overview of existing measurement technologies is that (i) existing technologies allow both parameters to be found, but that (ii) the existing measurement technologies to estimate the burstiness are — apparently — too involved for wide-scale use in practice. In the course of this thesis we develop a novel measurement approach to estimate the burstiness which is less involved than the existing techniques.

The second part of this chapter has been dedicated to the measurements that we have performed to assess the bandwidth provisioning problem. In Section 2.3 we have described our measurement procedure, which comprises capturing packet headers from 'network uplinks'. We have performed this measurement procedure at 5 distinct locations: a residential network of a university, a scientific research institute, a large college, an ADSL network access infrastructure and a server hosting provider. By choosing various networks with different characteristics in terms of number of users, network access technologies, etc., we — to some extent — reduce the chance that the remainder of our research (which uses our measurements for modeling and verification purposes) would significantly be affected by 'coincidences'. Furthermore, at each measurement location, we have measured a number of times. In total, some 850 packet header traces were collected, each comprising 15 minutes of network traffic.

In Section 2.4 we have presented some preliminary analyses of the measurements at the five different locations, giving an indication of the average traffic load and burstiness. Clearly, these are important factors in bandwidth provisioning, as we have seen in Chapter 1 as well. For now, they also serve as an indicative characterization of the traffic at the various network links.

In the next chapter we will continue with more in-depth analyses, ultimately yielding statistical models that describe the network traffic.

3 Traffic modeling

Traffic modeling has a long history and serves a variety of purposes, one of them being bandwidth provisioning. Traffic modeling is an important tool to represent real Internet traffic in a 'mathematical' way, in that a traffic model captures the statistical characteristics of the real traffic. One can think of characteristics such as mean traffic rate, fluctuations of the traffic rate, correlation structures, packet sizes, data transfer times, inter-packet spacing, etc.

In this chapter we first briefly discuss the current state of affairs in traffic modeling. Second we use traffic modeling to represent the real traffic that we have measured (see Chapter 2). The resulting traffic models will form the basis of our solution to the bandwidth provisioning procedure that will be further developed in the subsequent chapters, as is outlined in the picture below:

The organization of the present chapter is as follows:

- *Section 3.1 describes the state of the art in traffic modeling. As this is a broad and widely studied research area, we limit ourselves to a discussion on traffic modeling approaches that are relevant within the context of this thesis. In our discussion, we choose to make a distinction between two different approaches in traffic modeling: modeling of 'traffic flows' in which traffic of individual users is separately captured, and modeling 'aggregate traffic' in which all traffic on a network link is described (and hence, traf-*

fic flows are superimposed). We call the former 'flow-based modeling' and the latter 'black-box modeling'.

- *In Section 3.2 we apply flow-based modeling to represent real Internet traffic. We choose the (flow-based) $M/G/\infty$ input model, as it is commonly used in literature (see e.g., [PM97, ACFG04]). Using data sets from our measurements, we assess the fitting of real traffic using a $M/G/\infty$ input model. We also address (practical) limitations of the application of this model.*
- *In Section 3.3 we apply black-box modeling to represent real Internet traffic, and we choose to use a Gaussian model to represent the aggregate traffic. This model is also commonly used in literature (see e.g., [KN02, Fra02]). Likewise the previous section, we investigate fitting a Gaussian model to real traffic, and discuss the limitations of this model.*
- *Section 3.4 concludes this chapter.*

3.1 Overview of the traffic modeling research area

This section starts off with a short overview of the history of traffic modeling in telecommunication networks, from the traditional telephony networks to the modern Internet. As it turns out, traditional (telephony) modeling cannot be used to describe Internet traffic, and therefore various alternative models have been developed, which we introduce to the extent necessary for the remainder of this thesis.

3.1.1 Traffic modeling in early telecommunication networks

In the early days of telecommunications, i.e., the beginning of the 20th century, the Danish mathematician A.K. Erlang was hired by the Copenhagen Telephone Company. While working there he was presented with the classical problem of determining how many circuits are needed to provide an acceptable telephone service. Erlang is nowadays recognized as the founder of the fields of queuing theory and traffic engineering.

Throughout the 20th century, the use of teletraffic theory[1] has arguably been one of the most successful applications of mathematical techniques in industry [WP98]. One of the main reasons behind the success of (the application of) teletraffic theory, is the 'static' nature of the traditional telephony network, as well as the ever-present notion of 'limited variability'. These are characteristics of a 'homogeneous system', in that one may talk about 'typical users' and 'generic behavior', and hence, averages adequately describe the system's performance. Thus, rather simple and efficient rules for the number of circuits required, as function of the amount of traffic offered, could be developed, such that the call blocking probability is below some set level (see, e.g.,, [Tij94], and Section 1.3 of this thesis).

It is generally accepted that there are 'universal laws' governing (traffic on) traditional telephony networks. The most significant of which is that calls (presumably) arrive according to a *Poisson* process, when there is sufficient (circuit-)aggregation. The presumption (or 'law') goes as follows[2]: call arrivals are mutually independent, and the call interarrival times are all exponentially distributed with one and the same parameter λ (this is widely described in literature — for an early overview see e.g., [Jen48]).

The Poisson law, or in fact all teletraffic theory, was originally based on empirical studies on the public telephony networks. It has remained valid for at least fifty years [WP98], and so has another 'invariant' of telephony traffic: call durations (also referred to as holding times or sojourn times) are often assumed exponentially distributed.

Changes came from the 1980s onward, with the rise of data-traffic on the (in the past) primarily voice telephony networks. Starting with fax transmission and later wide-spread Internet access, the nature of the traffic on the telephony network changed. 'Data calls' tend to be significantly longer and more variable than voice calls. (Telephony) network operators could no longer rely on the traditional 'laws' and so-called multirate and later on variable bit rate (VBR) models were developed to cope with the more sophisticated telecommunication networks (e.g., ATM) — see literature such

[1] Teletraffic theory (originally) encompasses all mathematics applicable to design, control and management of the public switched telephone networks, i.e., statistical inference, mathematical modeling optimization, queuing and performance analysis [WP98].

[2] This may be true in most cases — however, calls to a power utility during a power failure would not be considered Poisson.

as [Kau81, Rob81, Kel91, RMV96] for more information on multirate and VBR models.

3.1.2 Traffic modeling in IP networks

By now, it is clear that the (modeling) rules change when computers do the talking instead of humans. Voice traffic has the property that it is relatively homogeneous (and predictable). In contrast, data traffic is more variable, and connections range from very short to very long, at very low to very high rates. The design principle of modern data/IP networks is also different from traditional voice/telephony networks: 'packet-switched' as opposed to 'circuit-switched'. Instead of having an end-to-end circuit reserved for the duration of a call, modern data/IP networks are based on individual packets that are self-contained, and routers only look at such packets to determine the destination. Consequently, routers do not need to keep track of each currently active connection.

The shift from circuit-switched to packet-switched networks has numerous implications. For example, networks are more efficient, as at any time capacity is available, newly arriving packets can benefit from that capacity. Also, packet networks are more robust, as they transparently route around failures in the network.

There are, however, also disadvantages to packet-switched networks. For instance, there is no admission control, which reduces the 'control' over the performance. A result of the lack of admission control is that links can become overloaded because packets arrive for transmission at an (aggregated) rate that may exceed the available transmission capacity. Such packets will be buffered, awaiting transmission. If the excess rate is sustained, and the buffer is full, packets will even be dropped, which is generally considered as undesired. The situation when the offered load of a data communication path exceeds the capacity is called congestion. The protocols in use on the Internet to transmit data, are designed (specifically TCP, the Transmission Control Protocol, but also new protocols like DCCP [KHF03]) to include end-to-end congestion control mechanisms, that automatically decrease the rate at which traffic is transmitted (by the source) when congestion is detected along the path, thus limiting packet drops.

An important consequence of the use of congestion control mechanisms like described above, is that characteristics of the traffic at any time may be influenced by conditions in the past. Such possible influences are, clearly, important for traffic modeling: Internet traffic may, thus, have (possibly complicated) *correlations* across time. Correlations can also be caused by the network application that generates the traffic (e.g., a web-browser, at the request of a user, typically 'fires' a burst of traffic on the network to fetch a web-page consisting of text and pictures, etc.). A good traffic model thus has to support such correlation structures in order to accurately represent real traffic.

Of course, researchers attempted to fit the traditional telephony models on data traffic. A few measurements studies sufficed, however, to discover that data traffic is highly variable, or *bursty*. It was also observed that traffic bursts in networks occur not only at a single timescale, but at many different timescales , see e.g., [LTWW94, Pax94]. Obviously, such multi-timescale burstiness was not present in the world of traditional telephony traffic modeling.

Let us now look more at the underlying mathematical ideas. Traditional telephony traffic is characterized by limited variability: the traffic arrival processes are independent or have temporal correlations that decay exponentially fast. Also, one can talk about 'typical' users and 'generic' behavior, where just averages accurately describe the system. The statistics for data/IP traffic, however, is one of high variability. Informally, high variability is a phenomenon where a set of samples takes values that vary over orders of magnitude (and, generally, most samples taking a low value, and a few samples taking very large values). For such a set of samples, describing this set with just the average value is largely uninformative.

Leland *et al.* [LTWW94] observed that Internet traffic variability is invariant to the timescale — the power-law behavior in time or space makes (some) statistical descriptors of data/IP traffic processes *fractal*-like, also known as *self-similar*. In the present context, we mean with fractal-like that a property of a traffic process is exhibited at multiple timescales (for a more precise definition, we refer to [WP98]).

A self-similar process can be described using *heavy-tailed distributions*. Self-similar processes are said to exhibit *long-range dependency* (LRD) — see, e.g., [WAL04, WP98] (and Appendix A to this thesis).

With the above described mathematical ideas in mind, researchers have come up with various statistical models to describe IP traffic. We will not discuss all these models in this thesis, but see e.g., [RMV96, TGV00] for an overview. We would like to point out though, and elaborate on, two fundamentally different approaches in IP traffic modeling:

- *black-box modeling*, which captures the statistical features of the superposition of many individual *flows*, and
- *flow-based modeling*, which considers the statistical features of individual *flows*

In the above definition, the notion of *flow* plays a crucial role. A flow can be described as follows. A stream of traffic is aggregated into flows that are coherent strings of packets. There is no fixed (technical) definition of which strings of packets should be considered a single flow, but some example definitions are given: a flow consists of all packets that

- belong to the same TCP connection or UDP stream,
- are exchanged between two IP addresses or (sub)networks,
- stem from the same traffic source (e.g. a single user),
- originate from the same (sub)network,
- etc.

In the next subsections we further elaborate on black-box and flow-based modeling, and make a comparison. The focus is on two models that are commonly used: *Gaussian models* for the black-box modeling approach, and $M/G/\infty$ *input models* for the flow-based modeling approach. Definitions of these models are given in the respective subsections.

3.1.3 Black-box modeling

In the IP world, various statistical models have been proposed to characterize traffic streams. The simplest model assumes Poisson arrivals of packets, but such a model has the undesirable feature that it fails to incorporate the correlations between packet arrivals observed in real traces. For this reason, the model with (a superposition of) ON/OFF sources is an attractive alternative: a broad variety of correlation structures can be modeled by

choosing appropriate distributions for the ON- and OFF-times — see, e.g., [BD98, LTWW94, PE95, Soh93].

More recently, the attention has somewhat shifted to *Gaussian traffic models* and multi-fractal analysis. With $A(s,t)$ denoting the amount of traffic arriving in $[s,t]$, a Gaussian model with stationary increments is such that $A(s,t)$ only depends on the interval length $t-s$. More specifically, $A(s,t)$ follows a Normal law, with mean $\mu\cdot(t-s)$ (for some mean $\mu>0$) and variance $v(t-s)$ (for some non-negative function $v(\cdot)$), for any s,t such that $s\leq t$.

It is clear that the Gaussian model is in some sense an 'artifact', as it, at least in principle, allows for negative input. This can be seen as follows. The probability for negative input ($A(s,t)<0$) is given by

$$\mathbb{P}\big(A(s,t)<0\big)=\Phi\left(\frac{-\mu\cdot(t-s)}{\sqrt{v(t-s)}}\right),$$

where Φ is the probability distribution function of a standard-normal random variable. When $\mu\cdot(t-s)$ is substantially larger than the standard deviation $\sqrt{v(t-s)}$, however, the above probability is very small, thus, it is highly unlikely that over an interval of length $t-s$ the increment is negative.

The further development of Gaussian models[3] was triggered by a number of measurement studies performed in the early 1990s, such as the famous Bellcore measurements [LTWW94]. These studies revealed extreme complexity and self-similarity in Ethernet traffic. Clearly, such phenomena on the link layer may relate to characteristics of traffic when regarding the higher layers in the protocol stack. For instance, Paxson and Floyd showed that [PF95] wide-area TCP traffic could also be modeled through a self-similar process.

A simple model with long-range dependency is a self-similar process characterized by a slowly (hyperbolically) decaying autocorrelation function. A stochastic model, advocated by Norros in [Nor94, Nor95], that has many desirable properties (e.g. long-range dependency) is a self-similar Gaussian process: *fractional Brownian motion* (fBm). In recent years the fBm model (and other Gaussian models) found wide-spread use as a reference model for IP traffic. In fBm, traffic arrives according to a Gaussian process with

[3] Note that, although the application of Gaussian models in telecommunication network traffic modeling was started only relatively recently, the Gaussian distribution was already developed by the German mathematician Gauß in the 19th century.

variance function $v(t) = \sigma^2 t^{2H}$, in which $H \in [0,1]$ is the so called Hurst parameter. With $0.5 < H \leq 1$, fBm exhibits long-range dependence.

Apart from the above (more or less) empirically derived motivations for Gaussian traffic models, there is also the argument of the Central Limit Theorem (CLT): by this theorem, the sum of a large number of 'small' independent (or weakly dependent), statistically more or less identical, random variables (users) has an approximately normal (i.e., Gaussian) distribution. Thus, one can expect that an aggregated traffic stream consisting of many individual communications may be modeled by a Gaussian stochastic process. The CLT argumentation does, however, not apply to all timescales: at the timescale of transmission of (minimum size) packets, the traffic stream is always ON/OFF (either there is transmission at link speed, or silence) — which is obviously not Gaussian. Thus, apart from the number of users (referred to as 'vertical aggregation'), there should also be sufficient aggregation in time ('horizontal aggregation'). The necessity for some aggregation in both directions for traffic to be Gaussian, was first pointed out by Kilpi and Norros in [KN02].

3.1.4 Flow-based modeling

A variant of the (packet level) ON/OFF model as described in the previous subsection, are so-called *flow level traffic models*.

The distinction between flow-based and the (packet-level) aggregate traffic modeling is illustrated in Figure 3.1 on page 54: in the top figure, the amount of traffic arriving per time interval is plotted for an aggregate traffic stream; below, it is shown how three individual flows contribute to this aggregate traffic stream. Note that in the bottom figure, the individual flows are stacked (thus cumulative). Obviously, in order to model individual flows, we need parameters to describe the properties of all these individual flows — opposed to modeling the aggregate where we only need to have parameters to describe the aggregated traffic.

Flow-based modeling can, mathematically, be described as follows:

- flows arrive according to some random process, for which one usually assumes a Poisson process with intensity λ;
- a flow stays in the system for some random duration D (e.g.,, this could be determined by the flow size), and the durations of flows are i.i.d.[4]; and
- during the time a flow stays in the system, the so-called sojourn time, the flow generates traffic at some (bit-)rate r.

A widely used flow-based traffic model is the so-called $M/G/\infty$ input model, see, e.g., [PM97], which is the Kendall notation for the flow-based model with the three properties described above.

There is vast body of literature on this topic. An overview of some approaches is given in, for instance, [TGV00, Ch.3] and references therein.

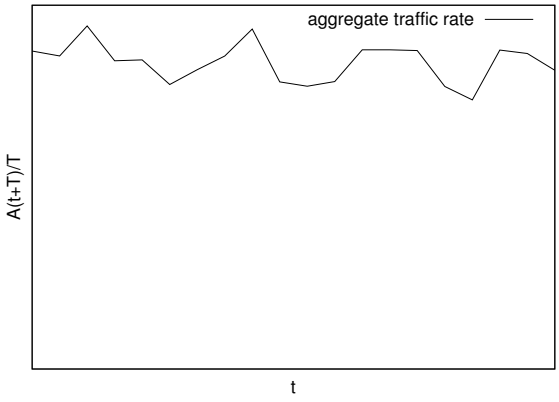
> **Remark:** Note that we do *not* consider the actual queuing and transmission of the offered traffic in this thesis, nor do we study we effects of any closed loop control (such as TCP mechanisms) on this traffic — there is a lot of literature available on those topics, however. In this thesis, we rather focus on (modeling) the arrival process, and its impact on dimensioning issues.

3.1.5 Flow-based versus black-box modeling

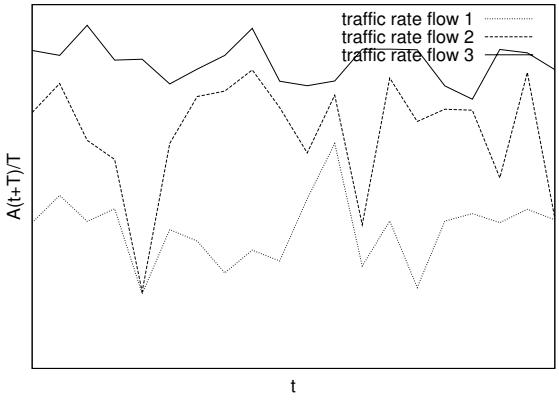
In the previous subsections we have introduced two different modeling approaches, namely black-box and flow-based models. We now briefly discuss some advantages and disadvantages of both approaches.

The most prominent advantage of using flow-based models, in which (through the modeling of flows) the characteristics of traffic of an individual user is considered, is that it facilitates sensitivity analysis. For instance, it enables the assessment of the effect of the migration of (a part of) the

[4] In probability theory, a sequence or other collection of random variables is independent and identically distributed (i.i.d.) if each has the same probability distribution as the others and all are mutually independent (definition from *Wikipedia*)



(a) Traffic rate of an aggregate stream over time



(b) Stacked (cumulative)traffic rates of three flows constituting the aggregate stream over time

Figure 3.1: Traffic rates: aggregate v. flow-based

user population from a 'slow' access technology to a 'faster' one: what is the impact on the bandwidth needed? Also the effect of a change in the flow-size distribution could be quantified[5]. Black-box models, in which aggregate traffic streams are modeled, however, do not model 'individual user behavior', but rather attempt to find an accurate statistical description of the *aggregate* of all users. Hence, a sensitivity analysis as described above for flow-based models, is not possible for black-box models. A commonly used subclass of black-box models are the *Gaussian* models.

A disadvantage of flow-based modeling over black-box modeling, in practice (see Section 3.2), is that the properties of various flows may differ significantly — also called heterogeneity — which can make it hard to capture the various flows in the same model. Heterogeneity between flows can have several causes. In the first place, the end-users use different applications, which are characterized by different bandwidth consumption patterns. For instance: streaming applications could use a constant bit rate (possibly well below the access rate), whereas file transfers are based on TCP (and grab as much bandwidth as possible, constrained by the access rate, the maximum window size, and the bottleneck elsewhere in the network). A second (perhaps more important) cause of heterogeneity lies in the fact that the bottleneck for different flows could be at different links or routers somewhere else in the network. For instance, two downloads from different servers at different locations in the network, could result in very different transmission rates.

To cope with the problems related to the heterogeneity described above, one could opt for splitting the user population in several subclasses, with their own characteristics. For instance, flows with a size (in bits) smaller than f could behave very differently from flows bigger than f (cf. the notion of 'mice' and 'elephants' as in, e.g., [GM01, ACFG04]). But of course, then this parameter f should also be chosen and tuned, so this leads to similar problems. In addition, when the set of applications changes (which happens every now and then), the parameter has to be tuned again.

Concluding, both models have their *pros* and *cons*, and hence, there is no reason to rule out one of them at this point. As discussed before, in the

[5]For example, if newer encoding standards for multimedia content result in different filesizes than before, this may well lead to a change in the flow-size distribution of Internet traffic

context of the research presented in this thesis, traffic models are used to resemble real Internet traffic. The next steps therefore are to assess the feasibility of modeling real traffic — as gathered through the measurements presented in Chapter 2 — through two models that are commonly used to model Internet traffic: $M/G/\infty$ (as a flow-based input model) and Gaussian (black-box). This is the topic of the next two sections.

3.2 Applying flow-based modeling to real traffic

In this section we will demonstrate how to apply flow-based modeling to describe real Internet traffic. As said before, a prominent advantage of flow-based modeling is that it allows sensitivity analysis of 'user-level parameters' — an attractive feature for bandwidth provisioning purposes: as illustrated by the following practical example:

> **Example:** In The Netherlands, various DSL service providers are active on the market. Probably as part of their effort to gain as much market-share as possible, the DSL service providers increased the access link speed (i.e., the bandwidth capacity of the link between the ADSL modem at the customer's premises and the DSLAM[6]) their customers a number of times — a quadruple increase over a period of about 2 years (early 21st century). These access links are generally seen as the 'bottlenecks' for the user's data traffic.
>
> In a flow-based model, the access link speed could thus determine the rate r at which traffic is generated. Obviously, an increase in r due to an access link speed upgrade, has impact on the characteristics of the traffic on the network. A relevant question for a DSL service operator could be: how much extra capacity would be required in its backbone to facilitate an increasing r. If a flow-based model is used to model the users' traffic, with the r just a parameter in the model, the required extra backbone capacity could be estimated beforehand.

[6] DSL access multiplexer; the device is installed at a telecommunication company's site, and separates the voice and data components from the subscriber lines and aggregates the data for sending over the company's network.

We choose to use the $M/G/\infty$ input model as the flow-based model, because of its widespread use in the research community. For an $M/G/\infty$ input model to be applicable to real traffic, the model parameters need to be known: the traffic rate r, the distribution of the duration D, and the arrival rate λ. In this section, we assess whether it is possible to estimate such parameters from the real traffic that we have measured on various Internet links (see Chapter 2). The approach taken in this assessment is as follows:

We first investigate whether traffic *within a single flow* is transmitted at a (more or less) *constant rate* (Section 3.2.1). Secondly (Section 3.2.2), we compare traffic rates *among various flows*, and assess whether each flow generates traffic at the same, i.e., *fixed, rate* as other flows. Motivated by measurements that show that most of the traffic is generated by only a limited number of large flows (so-called 'elephants'), we then repeat this exercise only taking into account these large flows. As access rate limitations are imposed on the user's transmission rate rather than the transmission rate of an individual flow, we also aggregate all (concurrent) flows as generated by the same user and investigate whether these aggregated traffic flows streams exhibit homogeneity. Based on these observations we argue that the model with a constant and deterministic rate does not apply, and therefore we shift our attention to the model with a constant but random transmission rate (Section 3.2.3).

3.2.1 Are traffic rates fixed and constant?

Various definitions of a *flow* of network traffic are in use, as discussed before. At this point, we define a flow following the common 5-tuple definition: a flow comprises all packets with the same:

- source IP address,
- destination IP address,
- transport protocol (e.g. TCP or UDP),
- source port, and
- destination port,

as long as the 'gap' between such packets does not exceed some predefined time interval[7]. For instance, in this definition all IP packets within the same TCP connection belong to a single flow.

Furthermore, in the remainder of this thesis we use the following properties of flows:

- the *flow size* is the amount of data transmitted as part of that flow, denoted in bytes or bits;
- the *flow duration* or *flow length* is the time difference between the first and the last packet of the flow, in seconds;
- the *traffic rate* is the rate at which traffic is generated (in a flow); when talking about the *flow rate*, this is averaged over the entire flow, i.e., flow duration divided by flow size.

Figure 3.2 shows four (relatively large) flows, picked from the approximately 60000 flows in a random packet trace taken from the traces collected at measurement location *R* (see Chapter 2 for descriptions of the various measurement locations). The slopes of the lines indicate the rate of the flows; the traffic rate of the 'fastest' flow is about eight times as high as the rate of the 'slowest'. The traffic rate within the flows, however, appears more or less constant (given the fairly straight lines). The differences in traffic rates may be explained by heterogeneity, for instance because the various flows stem

[7] We choose 20 seconds for the 'gap', as this turned out, from experiments, to be reasonable in that, for instance, TCP connections were almost always completely captured (and not more than that) through this definition.

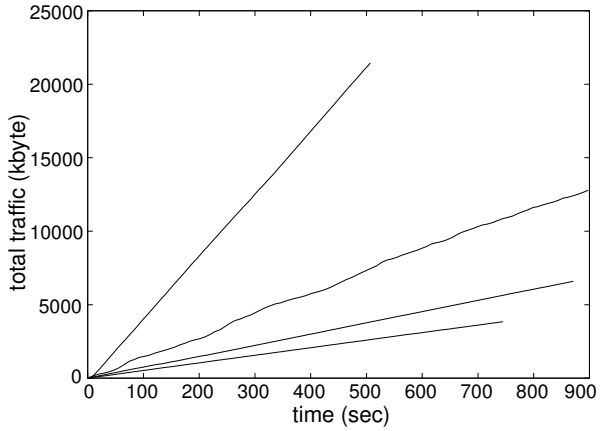


Figure 3.2: Accumulated traffic within four flows over time

from different applications with varying demands from the network, or occur at different bottleneck links (that limit the rate) on the network.

> **Remark:** For instance, a bit speculatively, the top flow transmits over 20 megabyte in over 500 seconds, which corresponds to some 320 kbit/s, a bit-rate that may be used in high quality MP3 music streams. Another possible explanation for the different traffic rates may be that various end-to-end paths may have different bottlenecks that restrict the transmission rate.

From the example flows given in Figure 3.2, we may conclude that it is not accurate to assume the traffic rate r to be a *fixed* value (in terms of $M/G/\infty$ modeling). They do, however, look *constant within* a single flow.

3.2.2 Are traffic rates fixed?

In this subsection we further compare traffic rates *between* flows. We assume, motivated by the discussion in the previous subsection, that traffic rates within single flows are constant — every flow itself may have a different traffic rate though. Figure 3.2 already gave the impression that traffic rates *between* flows may not be fixed, as the slopes for the different flows were not equal — we further study this in the present subsection by looking at all flows within a trace.

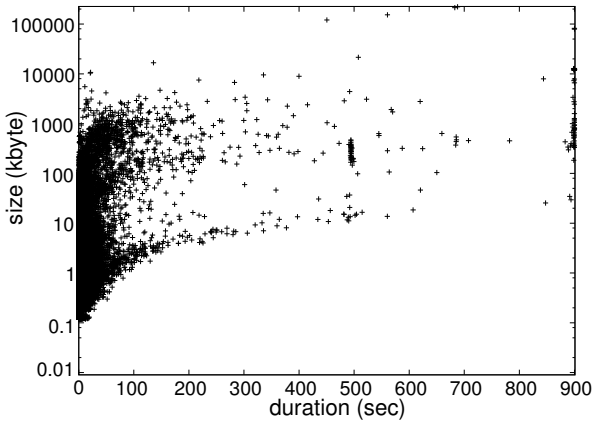


Figure 3.3: Flow durations compared to flow sizes

To investigate the variations of traffic rates between flows, we plot the duration of every flow against its size — their ratio is the traffic rate. As the traffic rate *within* a single flow appeared *constant* in Figure 3.2, we choose to represent each flow with a single point in Figure 3.3, instead of the lines used in Figure 3.2.

Figure 3.3 shows all flows (according to the above 5-tuple definition) in the trace from location *R*. If the traffic rates would be fixed (i.e., all flows generate traffic at the same rate), the points in Figure 3.3 should be on one line. Clearly, however, various flows of the same size, may take longer or shorter to complete. Similarly, the duration of a flow does not provide us with any information on the flow's size. In fact, we observe an extreme heterogeneity.

One may wonder whether this heterogeneity may be caused by the mice-elephants dichotomy, or by the fact that we should aggregate flows per user. We investigate both options.

The widely used assumption that Internet traffic is heavy-tailed, motivates our choice to 'zoom in' on larger flows. Therefore we first investigate which percentage of the flows cause what fraction of the traffic. Figure 3.4 clearly shows that only a small percentage of all flows accounts for most of the traffic. Hence, we decide to 'zoom in' on the approximately 3000 largest flows (corresponding to 95% of all traffic, and about 5% of all flows). For this subset of all flows, the duration-size pair of each flow is plotted in Figure 3.5; again, the spread of the *(duration,size)*-tuples suggests great heterogeneity.

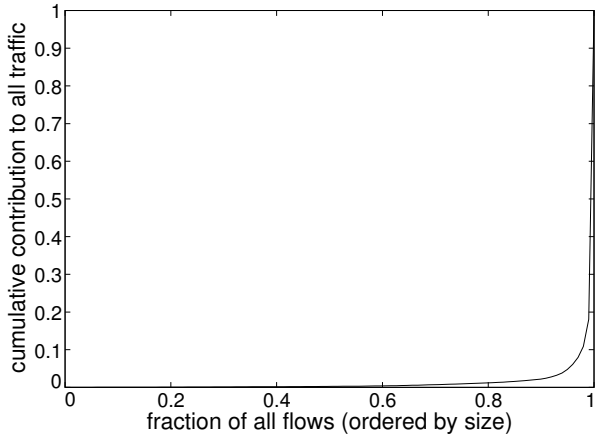


Figure 3.4: Cumulative distribution of flows and their contribution to all traffic

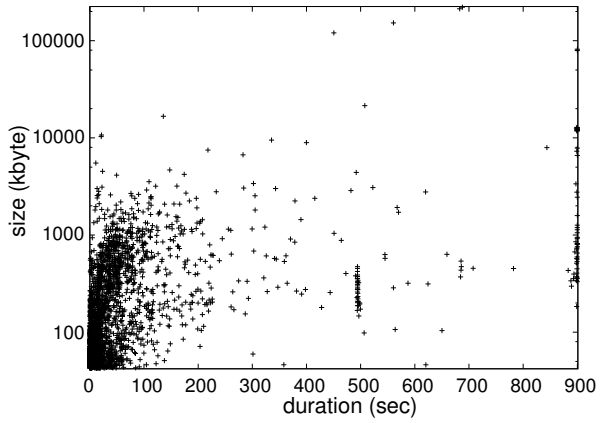


Figure 3.5: Flow durations compared to flow sizes (5% largest flows)

A single user can have multiple flows generating traffic concurrently, e.g., he may be browsing the web while a file download is going on in the background. These flows may interact with each other with regard to the rate at which each flow generates traffic when (partly) following the same Internet path; in any case, they share the access line. Such interaction may affect the 'homogeneity' in terms of rates of individual flows. Therefore, we aggregate flows generated by the same user that are 'overlapping in time', and with this new definition of flow we again plot the duration-size tuples. Note that we left out the 95% smallest flows, like above, because of their negligibly small contribution to the total traffic. With the new definition of flow,

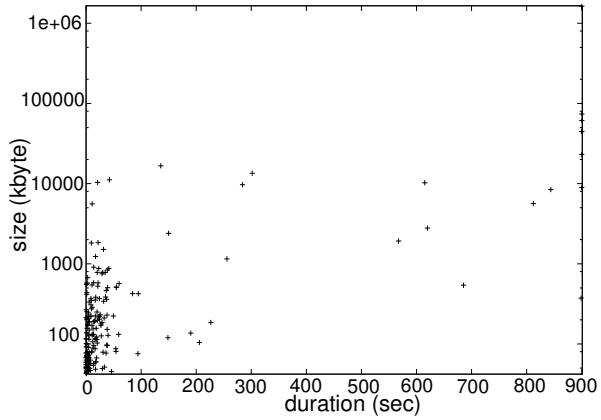


Figure 3.6: Flow durations compared to flow sizes (5% largest flows, aggregated per source IP address)

some 160 'aggregated flows' remain. Figure 3.6 shows the resulting *(duration,size)*-tuples. The cloud in Figure 3.6 is not as dense as before, however, but the 'spread' is still considerable.

From the discussion above we conclude that traffic rates in this trace from location R are not fixed between flows; this conclusion remains valid when only 'elephants' are considered, and when flows are aggregated per user.

This conclusion is also supported by analysis of an extensive number of other traces, taken from the same and other networks; although the specifics of the achieved rates, and spread of flow sizes and durations differ, the clouds suggest strong rate heterogeneity.

3.2.3 Traffic rate as a random variable?

The previous section showed that one cannot assume that the traffic rate is fixed between flows, although they appear constant within a single flow. Therefore, we now consider another option: the transmission rates are constant within flows, but the value R of this rate is random. In this subsection, we thus try to 'fit' real traffic with an $M/G/\infty$ input model with random rate R.

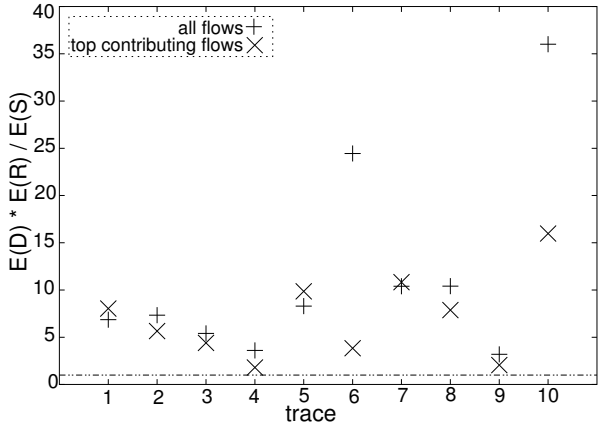


Figure 3.7: Ratios $\mathbb{E}D \cdot \mathbb{E}R/\mathbb{E}S$

For every single flow, the rate R is determined by its size and duration, according to $S = R \cdot D$, where S denotes the flow's size and D its duration. We first investigate whether rate R and D are independent.

Note that $\mathbb{E}S = \mathbb{E}(R \cdot D)$. If R and D are independent random variables, $\mathbb{E}(R \cdot D)$ equals $\mathbb{E}R \cdot \mathbb{E}D$. Hence, a necessary condition for independence is:

$$\frac{\mathbb{E}D \cdot \mathbb{E}R}{\mathbb{E}S} = 1. \tag{3.1}$$

We compute $\mathbb{E}D$, $\mathbb{E}R$, and $\mathbb{E}S$ for 10 different traces from location R, and plot the resulting fraction as in (3.1) in Figure 3.7. The same fraction is also plotted for the set of all flows that together constitute 95% of all the traffic in every trace, leaving the majority of flows (i.e., the smaller flows) out ('top contributing flows'). Clearly, in only 2 of the 10 traces the resulting fraction comes close to 1. Therefore, from Figure 3.7 we conclude that R and D are *not* independent in these traces of real traffic. This implies that we cannot fit the $M/G/\infty$ input model with random rates to our measured real traffic.

3.2.4 Discussion

From the discussion in Sections 3.2.1 — 3.2.3 we conclude that the traces considered do not simply fit in the framework of the $M/G/\infty$ input model. A few remarks are appropriate here:

Of course one could split the aggregate traffic stream into various smaller 'sub-streams', on the basis of size, application, etc., and then attempt to 'fit' these sub-streams with $M/G/\infty$ input models. Such an approach was pursued in recent studies on flow-based modeling, see, e.g., [ACFG04]. There it was found that it is possible to describe real traffic using an $M/G/\infty$ input model, but the accuracy of the fit is at the expense of the number of sub-streams, and the corresponding tuning parameters (for instant the threshold that distinguishes the mice from the elephants). We remark that a lot of effort is put into grouping flows that are similar, for instance elephants, together.

The extra effort that is required to estimate the modeling parameters accurately, may be unattractive to network operators. As Willinger and Paxson argue [WP98], for a traffic model to be widely used in practice, it should fit in many environments, and the model parameters should be relatively easy to guess or estimate. Also, when the nature of the traffic changes, for instance because of new popular applications, the estimation of the parameters has to be redone, which may, again, require significant effort.

An other important remark is that we have not succeeded in recognizing the access rate in our traces: the transmission rates (which were, as said before, constant during the flow's holding time) are apparently limited by other bottlenecks than the access rate. As a consequence, it appeared infeasible to do sensitivity analysis of the required bandwidth as a function of the access rate (at least in the cases that we investigated).

We thus have seen that the strong heterogeneity of the traffic in our traces appears to be a key problem to flow-based modeling. We now shift our focus to black-box modeling, which captures the statistical features of the aggregated traffic of many individual flows.

3.3 Applying black-box modeling to real traffic

In this section we turn to a black-box model, i.e., describing a traffic *aggregate*. We focus on the case of Gaussian traffic (see Section 3.1.3).

We first describe how to investigate whether traffic is Gaussian or not, following a procedure similar to the one discussed in [KN02]. In fact, it investigates whether the amount of traffic A arriving in intervals of length T

is normally distributed with mean μT and some standard deviation being the square-root of the variance of the amount of traffic per interval $\big($i.e., $v(T) := \mathrm{Var}(A(T))\big)$. In other words:

$$A(T) \stackrel{?}{\sim} \mathrm{Norm}\big(\mu T, v(T)\big).$$

Note that μT and $v(T)$ completely describe the Gaussian (normal) distribution at timescale T, thus an estimate of those two parameters suffices to describe the entire distribution at timescale T.

We investigate if a Gaussian model accurately describes the traffic aggregate in our traces, starting with a timescale of $T = 1$ second (Section 3.3.1).

Second (Section 3.3.2), we determine a simple quantitative measure for the 'goodness-of-fit' (i.e., 'normality'), dubbed the linear correlation coefficient. There are various alternative methods for determining goodness-of-fit, e.g., the *de facto* standard Kolmogorov-Smirnov test. As it turns out, different tests for normality sometimes give different results — as an aside, we assess whether the method we use gives similar results as the standard Kolmogorov-Smirnov test.

Furthermore, we repeat the procedure for many of our measurement traces in order to get an understanding of how Gaussian traffic generally is.

Third, we assess Gaussianity at other timescales than $T = 1$ second (Section 3.3.3). In particular we wonder what the minimum timescale is at which traffic is Gaussian — this relates to the 'horizontal aggregation' requirement mentioned in Section 3.1.3.

And finally, fourth, we investigate (Section 3.3.4) the minimally required aggregation in terms of users (traffic sources) ('vertical aggregation' in Section 3.1.3) for traffic to be Gaussian, extending the study by Kilpi and Norros [KN02]. Based on Central Limit Theorem type of arguments, it is expected that the more traffic sources are aggregated, the 'more Gaussian' the traffic tends to be.

3.3.1 A procedure for testing Gaussianity

In this section, we investigate whether the traffic in our traces is accurately described by a Gaussian process:

$$A(T) \stackrel{?}{\sim} \text{Norm}\left(\mu T, v(T)\right).$$

Note that literature suggests that this may be true for T not too small [FTD03, KN02]. We choose $T = 1$ second to start with, motivated by our expectation that timescales of roughly this order are relevant for performance as perceived by end-users of interactive applications like web-browsing. Later on, in Section 3.3.3, we will investigate other timescales.

The estimates $\hat{\mu}$ and $\hat{v}(T)$ of the average and (sample) variance of the traffic rates in our traces can straightforwardly determined:

$$\hat{\mu} = \frac{1}{nT} \sum_{i=1}^{n} A_i,$$

and

$$\hat{v}(T) = \frac{1}{n-1} \sum_{i=1}^{n} (A_i - \hat{\mu})^2,$$

where A_i denotes the amount of traffic offered in an interval of length T, and n the number of slots. Given our typical measurement interval of 900 seconds, we use $T = 1$ second and $n = 900$ slots at this point.

> **Remark:** We note that the convergence of the estimator of the sample variance could be rather slow when traffic is long-range dependent [Ber92, Ch. I]. Although our traffic likely is long-range dependent, 900 samples turns out to be sufficient to determine an accurate estimate.

We find that, for an example trace from location *R*, $\hat{\mu} = 18.9$ Mbit/s and $\hat{v}(1 \text{ sec}) = 24.3$ Mbit^2.

We use a so-called quantile-quantile plot (Q-Q plot) for visualizing the degree of Gaussianity of the traffic — Q-Q plots can be used to determine if two sets of data come from a common distribution. One of these two sets of data could be a reference distribution. As we assess the Gaussianity of the

traffic, we use the normal distribution as the reference distribution. A Q-Q plot presents the pairs

$$\left(\Phi^{-1}\left[\frac{i}{n+1}\right], s_{(i)}\right),$$

where n is the number of samples, $s_{(1)}, \ldots, s_{(n)}$ are the order statistics, and Φ^{-1} is the inverse of the normal cumulative distribution function with mean $\hat{\mu}$ and variance $\hat{v}(T)$. When the resulting points in a Q-Q plot lie (roughly) on the diagonal, the distributions are the same. Hence, the closer the point-pairs are to the diagonal in our Q-Q plots, the more Gaussian the distribution of $A(T)$ is. Additionally, one can also see from a Q-Q plot in what part(s) of the 'spectrum' a sample distribution does or does not match with the reference distribution. For instance, suppose that at the right-hand side of a Q-Q plot the point-pairs are above the diagonal. This is an indication that 'spikes' in the traffic rates are higher than would be expected from the reference distribution with such mean and variance (in case a Gaussian distribution is used as the reference distribution to compare the samples with).

Figure 3.8 shows the comparison between the traffic trace from location R and the Gaussian traffic model. Visually, the traffic seems to be 'fairly Gaussian', as most point-pairs are close to the diagonal. Note, however, how the Gaussian model fails to capture the head and tail of the distribution of $A(T)$: at both the left- and right-hand side of the graph in Figure 3.8, the point-pairs are above the diagonal. Because the point-pairs at the right-hand side are above the diagonal, 'spikes' in the traffic rates are higher than expected by this Gaussian model, which in turn motivates somewhat conservative bandwidth provisioning: the model underestimates the peak traffic rates.

3.3.2 Testing the goodness-of-fit

In order to get a quantitative measure of *goodness-of-fit*, we use the *linear correlation coefficient* as in [KN02, JSPA05]:

$$\gamma = \frac{\sum_{i=1}^{n}(s_{(i)} - \hat{\mu})(x_i - \overline{x})}{\sqrt{\sum_{i=1}^{n}(s_{(i)} - \hat{\mu})^2 \cdot \sum_{i=1}^{n}(x_i - \overline{x})^2}},$$

where the 'average model quantile' $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, and $x_i = \Phi^{-1}\left[\frac{i}{n+1}\right]$. Note that $-1 \leq \gamma \leq 1$, and $\gamma = 1$ means that the empirical distribution is identical to the reference distribution.

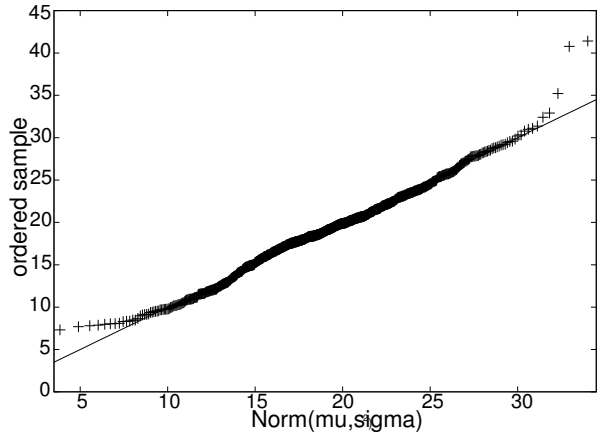


Figure 3.8: Q-Q plot — trace from location R compared to Gaussian

Remark: The linear correlation coefficient γ gives a quantitative measure for the goodness-of-fit. In this thesis, this is equivalent to the 'Gaussianity' of network traffic, as we only use this measure to compare a traffic trace with a Gaussian distribution. It is noted however, that the goodness-of-fit is not a test, in that it determines *whether or not* a traffic trace *is* Gaussian — an actual test is, e.g., the Kolmogorov-Smirnov test discussed below. We remark that we consider a traffic trace to be 'more Gaussian' when the value of γ is closer to 1 (when compared to another traffic trace). In the course of this thesis, we use rather subjective measures such as 'fairly Gaussian' to indicate that γ is 'quite close' to 1, but we intentionally do not propose precise thresholds as in 'for γ between 0.95 and 0.98, the traffic is fairly Gaussian'.

We find that $\gamma = 0.994$ for the trace from location R used in Figure 3.8, supporting the earlier 'fairly Gaussian' characterization, at timescale $T = 1$ second.

One may wonder how representative the example trace from location R is. Therefore we now look at all traces collected from our measurement locations. For each of these traces, we compute the goodness-of-fit γ, and we plot the results as to see how common certain values of γ are. The outcome is presented in Figure 3.9.

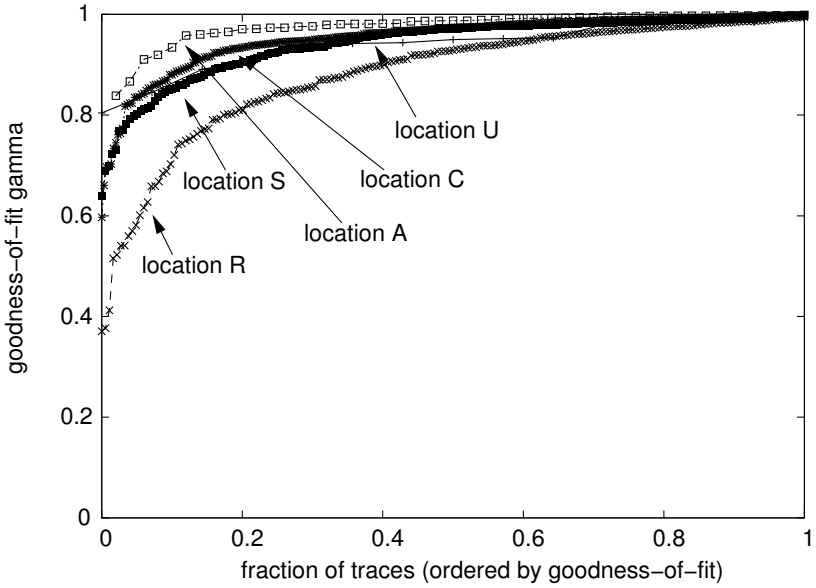


Figure 3.9: Distribution of the determined linear correlation coefficient γ over all measurements

From Figure 3.9 we may conclude that, for all locations except *R*, in about 80% of the cases γ is above 0.9, suggesting fairly Gaussian traffic. For location *R*, we will later see that the somewhat reduced Gaussianity is likely caused by the fact that there are fewer users active at the same time.

Comparison with Kolmogorov-Smirnov test

We have introduced γ (linear correlation coefficient) as a simple goodness-of-fit measure to compare an empirical distribution to a model (e.g. Gaussian) distribution. As we remarked, the linear correlation coefficient is not a real 'test' for Gaussianity, as it does not either accept or reject the hypothesis that an empirical distribution is Gaussian. In this section, we compare the linear correlation coefficient measure with a common (real) test for Gaussianity: the *Kolmogorov-Smirnov* (K-S) test , see e.g., [DS86]. The goal is to see whether or not a low γ value for a trace corresponds to the K-S test rejecting the hypothesis that the trace is Gaussian, and a γ value close to 1 corresponds to the K-S test not rejecting this hypothesis (also see remark below).

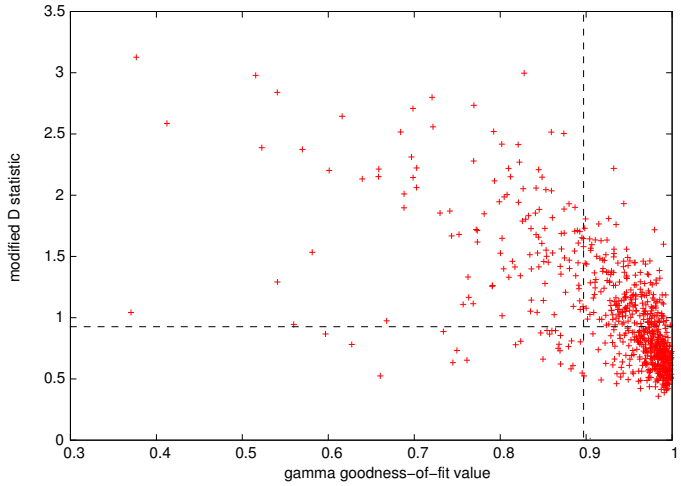


Figure 3.10: Comparison between γ and D (Kolmogorov-Smirnov) statistic, using all traces.

The Kolmogorov-Smirnov test can be described as follows: the cumulative frequencies of both the sample distribution as well as the model distributions are calculated. The greatest discrepancy between the observed (sample) and expected (model) distribution function is compared with a threshold (which is tabulated in statistical textbooks, e.g., [DS86]). If the discrepancy is above the threshold, one has to conclude that the observations do not stem from the model distribution.

> **Remark:** It is noted that the Kolmogorov-Smirnov statistical tests for normality (as well as the alternative Anderson-Darling test) are not able to guarantee that a sample is indeed normally distributed — they are only able to reject the hypothesis that a sample is normally distributed; if the test does not reject this hypothesis, one may not conclude that the sample is normally distributed.

We have compared the values of γ that are computed from hundreds of our traces, with the outcome of the Kolmogorov-Smirnov test (modified for estimations of mean and variance, see [DS86, Sect.4.8]). The results are depicted in Figure 3.10. It shows that, in about 80% of the cases, thus, roughly speaking, if the correlation coefficient is high (say, $\gamma > 0.9$, as indicated by the vertical line in the Figure 3.10), then the Kolmogorov-Smirnov test (at

significance level 0.05, with the corresponding threshold indicated by the horizontal line in Figure 3.10) does not reject the hypothesis that the underlying distribution of $A(T)$ is normal. In other words, the methods are in line with each other, and, as a consequence, it seems justifiable to use the 'easy' goodness-of-fit test based on γ, rather than the Kolmogorov-Smirnov test.

3.3.3 Time aggregation and Gaussianity

As pointed out, the use of Gaussian traffic models requires 'some' aggregation in time and number of users. In the present and the next section, we investigate 'how much' aggregation is required in terms of time and number of users, for traffic to be (close to) Gaussian.

We already investigated Gaussianity of traffic at a fixed timescale of T = 1 sec. In this section we will look into Gaussianity at other timescales, ranging from T = 5 msec to T = 5 sec. The choice for this range of timescales T is motivated by our expectation that these dominate the 'user-perceived performance level', and hence should be used for provisioning purposes.

An important question here is whether a computed value of γ at a given timescale gives a clear indication of γ at another timescale. Or in other words: if traffic is fairly Gaussian at a certain timescale, does that say anything about Gaussianity at other timescales? Suppose that a particular traffic stream exhibits strong Gaussianity at a timescale of, say, 5 seconds, and that such characteristic typically would be constant across timescales. If this is true, then, after having verified Gaussianity at a timescale τ, one could also assume Gaussianity at other timescales. Of course it would be tempting to also assume traffic to be Gaussian at smaller — possibly harder to measure — timescales as well, but, as remarked earlier, this reasoning could be dangerous (as, at very small timescales, traffic is certainly *not* Gaussian).

First, we look at an example with only a few traces. We determine γ at various timescales; the results, with five traces from measurement location R, are plotted in Figure 3.11. The impression from the examples in Figure 3.11 is that, as reflected by the more or less horizontal lines, the Gaussianity is quite constant over different timescales.

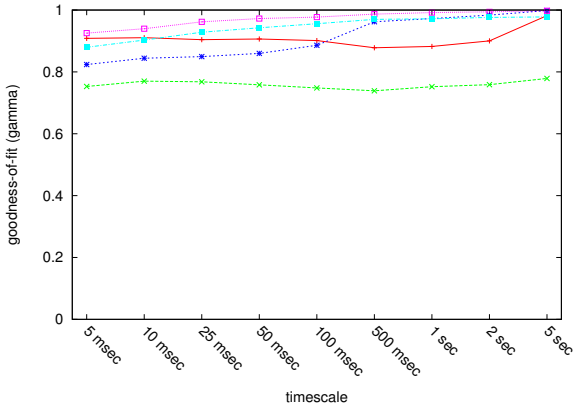


Figure 3.11: Comparing Gaussianity at different timescales, for 5 example traces from location *R*.

Next, we investigate this for all traces. We introduce ν_γ as measure of the 'variation of γ'. More precisely, we define ν_γ as the square root of the sample variance of the γ_τ values at all assessed timescales $\tau_1, \ldots, \tau_n \in T$:

$$\nu_\gamma := \sqrt{\hat{\mathrm{Var}}(\gamma_{\tau_1}, \gamma_{\tau_2}, \ldots, \gamma_{\tau_n})},$$

where we choose $T = \{5\,\text{msec}, 10\,\text{msec}, 25\,\text{msec}, 50\,\text{msec}, 100\,\text{msec}, 500\,\text{msec}, 1\,\text{sec}, 2\,\text{sec}, 5\,\text{sec}\}$. The interpretation is that when ν_γ is low, the traffic is (more or less) equally Gaussian (or non-Gaussian) across multiple timescales.

We have computed ν_γ for all traces at all 4 measurement locations. After ordering them from low to high values of ν_γ, they are plotted in Fig. 3.12. Clearly, ν_γ is small in most cases: in over 95% of the traces, ν_γ is below 0.05. Thus we may conclude that γ is quite constant over different timescales; in other words: traffic that exhibits Gaussian characteristics at one timescale, is likely to be Gaussian at other timescales as well (for the timescales that we investigated, at least).

Finally we have computed the 'average Gaussianity' of all traces at various timescales, i.e., the average value of γ for all traces at a particular location, for various timescales. These are plotted in Figure 3.13, together with error bars that represent the standard deviation of the computed γ values at a specific timescale.

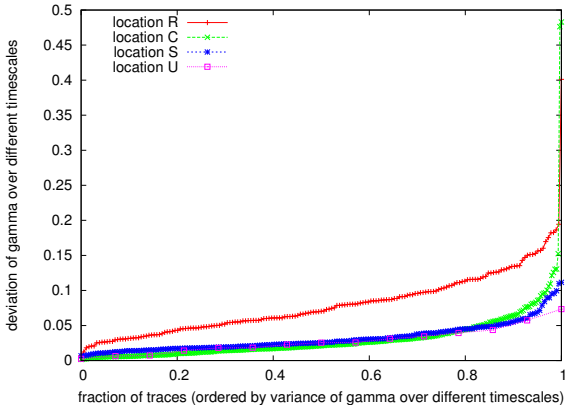


Figure 3.12: v_γ: goodness-of-fit γ over different timescales

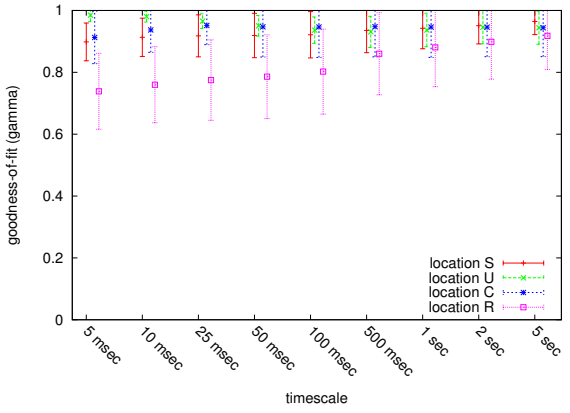


Figure 3.13: 'Average Gaussianity' at different timescales

3.3.4 User aggregation and Gaussianity

We discussed the impact of the horizontal aggregation, i.e., the timescale, on the Gaussianity of network traffic. Now we will look into the effect of the vertical aggregation, i.e., the number of users whose traffic is aggregated.

When traffic of a 'sufficiently large' number of users is aggregated, the resulting traffic mix exhibits strong Gaussianity [KN02]. We now investigate in further detail what 'sufficiently large' means. We do this by comparing the Gaussianity of network traffic as function of the number of users involved.

We rely on the same traces as used earlier. In these traces the traffic of a large number of users was aggregated, however. Therefore, we took from these traces a subset of all packets, namely just the packets that relate to a random subset of users. In this way we can investigate the Gaussianity of a traffic in which just a fraction p of all users is aggregated (as a function of this p). Our procedure to reduce the number of users involved is as follows:

We process the trace per packet; when a new IP address within the local network address range is found, with a probability p all of this IP address' traffic in the trace will be taken into account, with a probability $1-p$ traffic of this user is not selected, thus reducing the number of users as desired. The experiment is repeated with the same p, evidently leading to different results due to the random nature of the selection process, as well as with different p. The experiments yield input to our 'Gaussianity quantification procedure' described earlier: for various numbers of 'active users', a Gaussianity figure is computed.

The number of 'active users' is defined as follows. Per T, e.g. 1 second, it is observed how many distinct IP addresses (within the local network address range) send or receive traffic in that interval. The number of active users per experiment is then the average number of distinct IP addresses over all intervals (which is evidently not necessarily an integer number).

It is assumed that the traffic of the users that are not taken into account, does not influence the characteristics of the traffic of the users whose traffic *is* taken into account; this can be justified by the relatively high degree of overprovisioning of the measured network links.

Figure 3.14 shows for locations R (top) and S (bottom) how the number of active users relates to the Gaussianity of the network traffic, taking only a few example traces into account. We have limited ourselves to locations R and S, as these are the locations with the least number of users. As could be expected, Gaussianity increases with the number of active users. Also, for a given number of active users, there is typically quite some variation in the Gaussianity (between traces as well as within the same trace but for different experiments, i.e. with different subsets of selected users). Notably, there are cases when only a few users are active on average, but still the Gaussianity is almost 1. In these cases, a small number of users were dominating the trace and happened to be selected, and apparently their traffic plus the 'noise' of

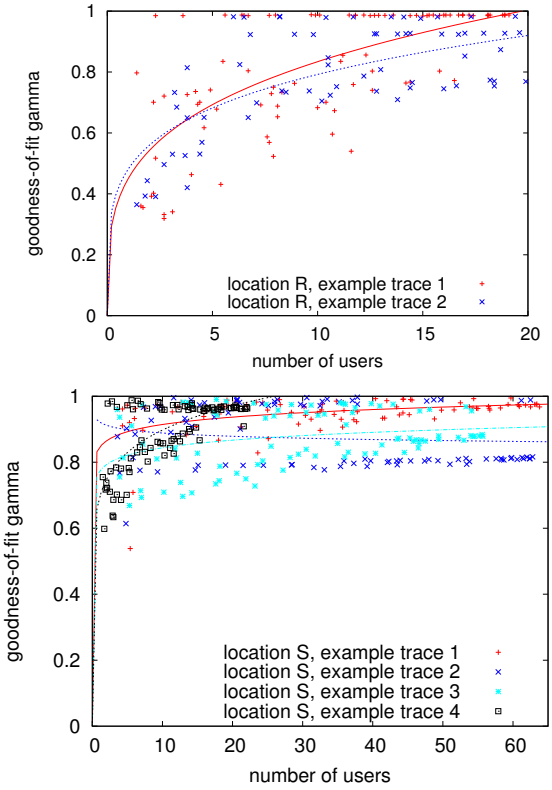


Figure 3.14: Gaussianity compared to number of active users

the others is Gaussian. Note that analysis of the measurements at the other locations gives similar results.

Figure 3.14 gives a first impression on the relation between the number of active users and the Gaussianity of the resulting aggregated traffic. For illustration purposes we have added (least squares) fits for the data plots; the fits are based on the formula $\alpha \cdot N^{\beta}$, where N denotes the number of users and α and β are (scaling) parameters. Next, we want to get a more thorough expression of this relation.

We compute the γ values for various experiments (as described above), and aggregate the results in two dimensions: (i) the number of users involved is grouped per 5, and (ii) the γ values are averaged and plotted together with an error bar indicating the standard deviation. The results are plotted in Figure 3.15. The top picture shows the result for location *R*; below for loca-

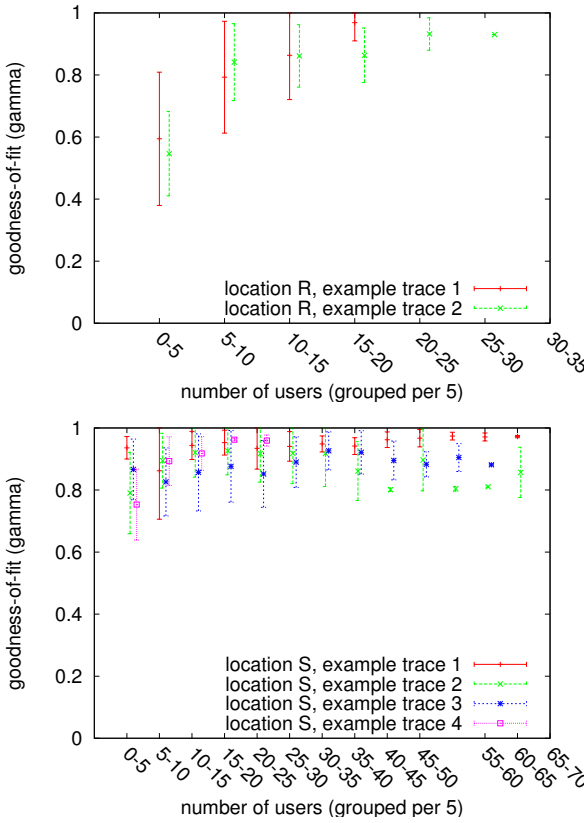


Figure 3.15: Gaussianity compared to number of active users (grouped)

tion *S*. As the primary interest here is on the lower side of the spectrum of the number of users (as, for large numbers of users, we already know traffic is quite Gaussian), we have limited ourselves here to the two locations with the least number of users.

From Figure 3.15 it can be seen that, as expected, an increase in the number of users involved tends to increase the Gaussianity. It is not possible, however, to give a hard number saying 'above N users, traffic may be assumed Gaussian'. It seems justified to claim that 'only a few tens of users' make the resulting traffic fairly Gaussian (at this timescale). These results are in line with Fraleigh [Fra02] and Kilpi and Norros [KN02], although it seems that, for our traces, even less (user) aggregation is required than in theirs for traffic to be safely assumed Gaussian — obviously, many facets are involved here.

3.4 Concluding remarks

In this chapter we have discussed network traffic modeling in both the traditional telephony network as well as the modern Internet. It was shown that models for telephony do not hold for the current Internet. A main cause of this is the high variability of Internet traffic compared to the limited variability of (voice) telephony traffic, in terms of both time and offered traffic. The high variability of Internet traffic has led to the popular belief that it is self-similar (fractal-like), long-range dependent and heavy-tailed.

The role of the present chapter in this thesis is to determine traffic models that can be used to represent real traffic, as a foundation for our further research on bandwidth provisioning. We have made a distinction into two modeling approaches, viz. flow-based modeling that considers individual flows (such as TCP connections), and black-box modeling in which the superposition of many flows is considered. We have compared both modeling approaches, and found that both have their own merits as well as drawbacks.

A merit of flow-based modeling is the sensitivity analysis for flow parameters, i.e., it can be determined what the influence of a change in for instance the distribution of flow durations, is on the resulting traffic stream. Trying to fit a flow-based model, i.e., the $M/G/\infty$ input model, on real traffic that we have measured on various networks, however, turned out to be rather cumbersome. This is caused by the strong heterogeneity that we observed in our analysis of the real traffic. We note that, by grouping flows that are more or less similar, leading to reduced heterogeneity, Azzouni *et al.* were able to model a traffic stream using the $M/G/\infty$ approach [ACFG04]. A practical drawback of such grouping, however, clearly is the increased number of parameters that need to be estimated (as for every group, the model parameters need to be found), as well as the effort that comes with the grouping itself.

Black-box modeling does not consider individual flows, and hence does not allow for the above mentioned sensitivity analysis. Focusing on the commonly used Gaussian (black-box) model, an advantage of this model is that there are only two parameters that need to be estimated: the mean offered traffic rate and variance of the offered traffic at the desired timescale. After fitting the Gaussian traffic model on numerous traffic traces, we may

conclude that real traffic can be accurately represented through a Gaussian model, for timescales as low as 10 milliseconds and traffic aggregated from some tens of users or more.

Thus, we believe that the Gaussian model is to be preferred in operational environments. The main reason for this choice is that Gaussian models seem to hold under many circumstances, and it is relatively easy to estimate the model parameters. Nevertheless, $M/G/\infty$ modeling is also attractive, but is more difficult to use in practice.

In the next chapter, we will use the Gaussian and $M/G/\infty$ input models to develop various bandwidth provisioning rules that are based on these models.

4 Bandwidth provisioning rules

Various traffic models were introduced in Chapter 3, and it was shown that these models (to a certain extent) accurately describe real network traffic. In the present chapter, these traffic models are used in derivations of various mathematical 'bandwidth provisioning formulas'. Such formulas determine the (estimated) required bandwidth capacity, based on the traffic model and its parameter estimates, which together represent the actual traffic. Of course, the required capacity also heavily depends on the chosen performance criterion, and hence, this criterion is also part of a bandwidth provisioning formula — recall from Section 1.2 that our intention is to achieve link transparency, i.e., the link should not be the 'bottleneck', which is achieved by choosing C such that $\mathbb{P}(A(T) \geq CT) \leq \varepsilon$.

The figure below illustrates the position of the present chapter in this thesis.

This chapter is organized as follows:

- *In Section 4.1 a generic bandwidth provisioning formula is derived with minimal modeling assumptions on the offered traffic — in fact, no other modeling assumptions but stationarity are made.*

  Next, we use the two traffic models introduced in Chapter 3 to further detail the generic provisioning formula:
- *In Section 4.2 it is assumed that the traffic may be described by an $M/G/\infty$ input model. This is used to determine a*

bandwidth provisioning formula that is parameterized using $M/G/\infty$ modeling parameters (traffic rate r, etc.).

– *In Section 4.3 a bandwidth provisioning formula is derived based on the assumption that the traffic is accurately described via a Gaussian process — giving a provisioning formula that is parameterized through the mean traffic rate μ and its variance $v(\cdot)$ at timescale T (note that the choice of T depends on the specified performance criterion).*

  In order to validate the provisioning formulas found above, we use our traffic traces to get an idea of how much capacity would be required to cater for the traffic in those traces.

– *In Section 4.4 we compare the required capacity as estimated through analysis of the traces, with the estimated figure from the provisioning formulas derived in this chapter. As it turns out from the analysis of the traces, there may exist alternative provisioning formulas, or better, rules of thumb, that may estimate the required bandwidth capacity about equally well. We will briefly discuss these alternatives as well.*

Note that in the previous chapter we have found that a Gaussian traffic model is very attractive in practical environments. Therefore, importantly, the bandwidth provisioning rule for Gaussian traffic that we develop in Section 4.3, is attractive to use in practical situations. Nevertheless, in cases where alternative models are more attractive, one could resort to, e.g., a provisioning rule based on $M/G/\infty$ modeling, as also presented in this chapter.

4.1 Provisioning formula for general traffic

The typical network environment that we focus on in the research presented in this thesis, is a network link with a considerable amount of users (e.g., the link between an enterprise and the backbone of its ISP) that carries mostly TCP traffic (e.g., from web-browsing). For such environments, it is important that the network link is more or less 'transparent' to the users, in that

the users should not (or almost never) perceive any degradation of their performance due to a lack of bandwidth.

Clearly, this objective will be achieved when the link rate is chosen such that only during a small fraction of time, denoted with ε, the aggregate rate of the offered traffic (measured on a sufficiently small timescale T), exceeds the link rate. Here ε and T are performance-parameters: the smaller they are chosen, the more stringent the performance criterion — see Chapter 1.

Reiterating from Chapter 1, the objective 'link transparency' can be stated as follows: the fraction (corresponding to 'probability') of sample intervals of length T in which the aggregate offered traffic rate $A(T)/T$ exceeds the available link capacity C, should be below ε, for pre-specified values of T and ε. In other words, the performance criterion is, cf. (1.2):

$$\mathbb{P}\big(A(T) \geq CT\big) \leq \varepsilon$$

This criterion can be seen as a *statistical guarantee* on the performance: it is accepted, with a certain (small) probability ε, that the offered traffic per time interval exceeds the bandwidth capacity.

For bandwidth provisioning purposes, the crucial question is: *what is, for given T and ε, the (minimally) required bandwidth capacity $C(T,\varepsilon)$ to meet the above performance criterion?* In the remainder of the present section as well as the next two sections, we will derive formulas to find this $C(T,\varepsilon)$ (for some given traffic models).

General traffic. Based on the Markov inequality $\mathbb{P}(X \geq a) \leq \mathbb{E}(X)/a$ for a non-negative random variable X, we have, by putting $X = \exp(\theta A(T))$, for $\theta \geq 0$, the following upper bound on the 'overflow probability':

$$\begin{aligned} \mathbb{P}\big(A(T) \geq CT\big) &= \mathbb{P}\left(e^{\theta A(T)} \geq e^{\theta CT}\right) \\ &\leq \mathbb{E}e^{\theta A(T) - \theta CT}. \end{aligned}$$

Note that $\mathbb{E}e^{\theta A(T)}$ is the so-called moment-generating function for $A(T)$. Thus, it implicitly captures the entire 'distribution' of $A(T)$: the average amount of traffic, variances, etc. See also Appendix A to this thesis for more information on a moment-generating function.

Because the above holds for all non-negative θ, we can choose the tightest upper bound[1]:

$$\mathbb{P}\big(A(T) \geq CT\big) \leq \inf_{\theta \geq 0} \Big(\mathbb{E}e^{\theta A(T) - \theta CT}\Big). \tag{4.1}$$

This bound, which is also known as the Chernoff bound, is usually quite tight. It is rather implicit, however, as it involves the computation of the entire moment generating function $\mathbb{E}\exp\big(\theta A(T)\big)$ and an optimization over θ.

We can now derive a formula that calculates the minimally required C, as follows. Of course, to meet our criterion it suffices that the right-hand side of (4.1) is below or equal to ε:

$$\inf_{\theta \geq 0} \Big(\mathbb{E}e^{\theta A(T) - \theta CT}\Big) \leq \varepsilon.$$

Clearly, there should be at least one value for θ that meets the above:

$$\exists \theta : \mathbb{E}e^{\theta A(T) - \theta CT} \leq \varepsilon.$$

Now we take the logarithm of both sides of the equation, and rewrite to isolate C, to arrive at:

$$\exists \theta : C \geq \frac{\log \mathbb{E}\exp\big(\theta A(T)\big) - \log \varepsilon}{\theta T}.$$

Thus, for our performance criterion to hold, it suffices to choose C as follows:

$$C(T, \varepsilon) \geq \inf_{\theta \geq 0} \frac{\log \mathbb{E}\exp\big(\theta A(T)\big) - \log \varepsilon}{\theta T}. \tag{4.2}$$

We refer to (4.2) as the *generic bandwidth provisioning rule*. For *any* traffic arrival process — as long as it is completely known — (4.2) gives the minimally required capacity to meet our 'link transparency' performance criterion (1.2), as long as the traffic is stationary[2]. We note that, very roughly

[1] In this and other expressions, inf (for infimum) may be read as minimum, and sup (for supremum) as maximum.

[2] Stationarity may be described as follows: with $A(s,t)$ denoting the traffic arrived between times s and t, the stationarity assumption is that the distribution of $A(s+\delta, t+\delta)$ does not depend on δ but only on the interval length $t-s$. In this thesis we use the abbreviation $A(t) := A(0,t)$.

speaking, if there is no stationarity, there basically is no way to accurately model the traffic at all, let alone to do proper bandwidth provisioning.

Assuming realistic traffic[3], one can also see in (4.2) that $C(T,\varepsilon)$ decreases in T and ε, corresponding to the intuitive argument that less bandwidth capacity is required when the performance criterion is loosened.

Performance criterion (1.2) that we used so far does not take into account the option of *buffering* packets. As an aside to the main text of this thesis, which focuses on achieving link transparency through the performance criterion (1.2), we mention the following alternative performance criterion. This alternative criterion, which takes buffering into account, is as follows:

$$\mathbb{P}(Q > B) \leq \varepsilon, \tag{4.3}$$

where B denotes the size of the buffer (or queue) of the network link. The associated bandwidth provisioning question is 'what is the required bandwidth capacity C such that (4.3) holds?' Note that B/C could be interpreted as an upper bound on the delay incurred by packets that go through the buffer.

One can derive a formula to estimate C to meet performance criterion (4.3), which turns out to be similar to the previous derivation for the 'link transparency' criterion (1.2), as follows (see also [AMN02, FTD03]).

To start, note that the probability of buffer 'overflow', i.e., $\mathbb{P}(Q > B)$ is equivalent to the probability that more traffic is offered over an interval than can be handled by the network link, plus an amount of data B in the buffer:

$$\mathbb{P}(Q > B) \;\equiv\; \mathbb{P}\big(\exists T \,:\, A(T) \geq CT + B\big).$$

Following the 'principle of the largest term', we can approximate the right-hand side of the above equivalence:

$$\mathbb{P}\big(\exists T : A(T) \geq CT + B\big) \;\approx\; \sup_{T \geq 0} \mathbb{P}\big(A(T) \geq CT + B\big).$$

[3] Realistic in the sense that it has properties as generally seen in Internet traffic: stationarity, long-range dependence, self-similarity, increasing $v(\cdot)$, etc. For instance, we rule out cases in which $v(\cdot)$ is locally decreasing, such as in the so-called Brownian bridge, where $v(t) = t(1-t)$.

We then use the Chernoff bound again, viz.:

$$\mathbb{P}\big(A(T) \geq CT + B\big) \leq \inf_{\theta \geq 0} \Big(\mathbb{E} e^{\theta A(T) - \theta CT - \theta B} \Big).$$

The remainder of the derivation of C is analogous to the earlier derivation, and leads to the following formula for required bandwidth which ensures that $\mathbb{P}(Q > B) \leq \varepsilon$:

$$C \geq \sup_{t \geq 0} \inf_{\theta \geq 0} \frac{\log \mathbb{E} \exp\big(\theta A(t)\big) - \log \varepsilon - \theta B}{\theta t}. \tag{4.4}$$

Note that the above required bandwidth formula (4.4) to meet performance criterion (4.3) requires the entire moment generating function $\mathbb{E} \exp\big(\theta A(t)\big)$ to be known, for all $t \geq 0$. This is a similar requirement as for the required bandwidth formula (4.2) to meet our 'link transparency' criterion (1.2): in the latter case, the moment generating function of $A(t)$ has to be known at timescale $t = T$. We stress that the estimation of the moment generating function may be demanding and not straightforward to do.

By imposing additional structure on the input traffic $A(\cdot)$, however, we may simplify the estimation of the required bandwidth. Clearly, if we have a well-defined $A(\cdot)$, and the moment-generating function of $A(\cdot)$ can be determined, the above formulas for bandwidth provisioning might be simplified. A traffic model may be seen as a well-defined $A(\cdot)$; we have discussed the modeling of real traffic in Chapter 3. For both discussed models, i.e., $M/G/\infty$ and Gaussian input, we derive more simple provisioning formulas (that do not require the entire moment-generating function to be estimated) in the next two sections. Note that both models (or classes of models, in fact) are rich in the sense that they allow for any possible correlation structure in $A(\cdot)$.

4.2 Provisioning formulas based on $M/G/\infty$ input

In order to derive the minimally required bandwidth for $M/G/\infty$ input, we determine the (log-)moment generating function in the generic bandwidth provisioning formula (4.2). Clearly, if we have the moment generating function, we can also simplify (4.4), but this is left out of this thesis.

Recall the following about the $M/G/\infty$ input model:

- flows arrive according to a Poisson process with intensity λ;
- a flow stays in the system for some random duration distributed as the random variable D (i.i.d.);
- during the flow's sojourn time, traffic is generated at some rate r.

Let the mean flow duration $\mathbb{E}D$ be denoted by δ. Now the mean (aggregated) input rate μ equals $\lambda\delta r$.

By choosing D appropriately, a broad range of input (correlation) structures may be covered. For instance, a heavy-tailed distribution corresponds to long-range dependent traffic.

Now let us work further on determining the sought (log-)moment generating function $\log\mathbb{E}\exp\left(\theta A(t)\right)$:

First, denote by $F_D(\cdot)$ the distribution function of D, and by $F_{D^r}(\cdot)$ the distribution function of the residual flow length. The corresponding densities are denoted by $f_D(\cdot)$ and $f_{D^r}(\cdot)$. The relations between these functions are given by:

$$f_{D^r}(t) := \frac{1-F_D(t)}{\mathbb{E}D}, \quad \text{and}$$
$$F_{D^r}(t) := \int_0^t f_{D^r}(s)\,ds = \int_0^t \frac{1-F_D(s)}{\mathbb{E}D}\,ds.$$

Then, with $A(t)$ the amount of traffic generated by a single $M/G/\infty$ input in an interval of length t, we have to distinguish between flows that are already active at the start of the interval, and flows that newly arrive during the interval, as these groups clearly do not equally contribute to $A(t)$:

- The number of flows that were already active at the start of the interval has a Poisson distribution with mean $\lambda\delta$. Their residual duration has density $f_{D^r}(\cdot)$; with probability $(1-F_{D^r}(t))$ they generate traffic during the entire interval;
- The number of flows that arrive during the interval has a Poisson distribution with mean λt. Given that the number of these arrivals is a non-negative integer, their arrival epochs are i.i.d. random variables, uniformly distributed over the interval (with density $1/t$). Their duration has density $f_D(\cdot)$.

Finally, straightforward computations (see [MSS05] for details) now yield the desired log-moment generating function:

$$\log \mathbb{E} e^{\theta A(t)} = \lambda \delta (M_t(r\theta) - 1) + \lambda t (N_t(r\theta) - 1), \tag{4.5}$$

with

$$\begin{aligned} M_t(r\theta) &:= \int_0^t e^{r\theta x} f_{D^r}(x) dx + e^{r\theta t}(1 - F_{D^r}(t)), \quad \text{and} \\ N_t(r\theta) &:= \int_0^t \int_u^t \frac{1}{t} \cdot e^{r\theta(x-u)} \cdot f_D(x-u) dx du \\ &\quad + \int_0^t \frac{1}{t} \cdot e^{r\theta(t-u)} \cdot (1 - F_D(t-u)) du. \end{aligned}$$

The expression (4.5) for the (log-)moment-generating function appears rather complex. It only requires, however, knowledge of r, the flow duration distribution D and the flow arrival intensity λ, to use it and thus be able to compute the minimally required bandwidth capacity. Note that finding proper estimates for these variables may be too complicated (if possible at all) for use in practical environments — see the discussion in Chapter 3.

If one would be able to find the above mentioned estimators, it is straightforward to determine $M_t(r\theta)$ and $N_t(r\theta)$, use these in (4.5), and then find the minimally required capacity according to (4.2), assuming an $M/G/\infty$ input model to represent the input traffic.

Next, we use the generic bandwidth formula (4.2) and assume a Gaussian model to represent the input traffic, to derive a bandwidth provisioning formula for Gaussian traffic.

4.3 Provisioning formulas based on Gaussian input

Recall the notion of a 'black-box' model, in that it abstracts from modeling individual users or flows. A commonly used subclass of black-box models are the Gaussian models. Assuming that the traffic aggregate $A(T)$ contains contributions of many individual users, in many situations it is justified to assume that $A(T)$ is Gaussian if T is not too small, see e.g., [FTD03, KN02], as well as the discussion in Section 3.3 of this thesis.

In other words, $A(T) \sim \text{Norm}\big(\mu T, v(T)\big)$ (and note that μ denotes the long-term average offered traffic rate). In the next few steps, we derive an expression for the log-moment generating function for this Gaussian model, to plug into the generic bandwidth provisioning formula (4.2).

An alternative way of looking at the distribution of the offered traffic, equivalent to the above mentioned $A(T) \sim \text{Norm}\big(\mu T, v(T)\big)$, is the following: $A(T)$, i.e., the amount of traffic offered over an interval of length $T > 0$, is distributed as $\mu \cdot T + \sqrt{v(T)} \cdot U$, where U follows the standard normal distribution. Then, by using the definition of the integral of the normal distribution function (which obviously integrates to 1), we derive

$$\begin{aligned}
\mathbb{E}e^{\theta U} &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\theta x} e^{-\frac{1}{2}x^2}\, dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\theta)^2} e^{\frac{1}{2}\theta^2}\, dx \\
&= e^{\frac{1}{2}\theta^2},
\end{aligned}$$

and inserting this into $A(T) \sim \mu \cdot T + \sqrt{v(T)} \cdot U$, we obtain

$$\mathbb{E}e^{\theta A(T)} = e^{\theta \mu T + \frac{1}{2}\theta^2 v(T)}.$$

We conclude that the log-moment generating function in the generic bandwidth provisioning formula (4.2) evidently is given by:

$$\log \mathbb{E}e^{\theta A(T)} = \theta \mu T + \frac{1}{2}\theta^2 v(T). \tag{4.6}$$

The bandwidth provisioning formula for Gaussian input for performance criterion (1.2) can now be determined by inserting (4.6) into (4.2):

$$\begin{aligned}
C(T,\varepsilon) &\geq \inf_{\theta \geq 0} \frac{\theta \mu T + \frac{1}{2}\theta^2 v(T) - \log \varepsilon}{\theta T} \\
&= \mu + \inf_{\theta \geq 0} \left(\frac{\frac{1}{2}\theta v(T)}{T} - \frac{\log \varepsilon}{\theta T} \right),
\end{aligned}$$

which reduces to the following formula for the minimally required bandwidth capacity after optimizing over θ, assuming Gaussian input:

$$C(T,\varepsilon) = \mu + \frac{1}{T}\sqrt{(-2\log\varepsilon)\cdot v(T)}. \tag{4.7}$$

The following observations can be made from expression (4.7):

- The required capacity depends on the 'long-term' mean traffic rate μ, added with a term that incorporates *burstiness*, i.e., the variance $v(T)$ of the traffic arrivals. Note that this is in line with the observations in Chapter 1, especially (1.1). Also, it follows from (4.7) that the larger $v(T)$, thus the burstier the traffic, the more capacity is required to cater for the total traffic.
- The required capacity increases when the performance criterion becomes more stringent: if the 'overflow probability' ε gets smaller, C increases; and also, as $v(T)$ cannot increase faster than quadratically, $\sqrt{v(T)}/T$ decreases in T, or in other words, when a smaller interval length T is taken, more capacity is required. These properties clearly correspond to one's intuitive expectation.
- μ being the 'long-term' mean traffic rate, it can easily be estimated through coarse measurements (like the traditional SNMP measurement approach). Estimating $v(T)$, however, is more involved as it requires measurements on the typically small timescale T (say 1 second or less). This may not be a problem when detailed information about the traffic is available, as is the case in our measurements, but this may not be realistic in practice. In Chapter 5 we will describe a novel method to estimate $v(T)$ for small T, *without* requiring such detailed measurements.

In Section 4.4.2, and extensively in Section 6.2, we will investigate if the required bandwidth formula for Gaussian traffic (4.7), indeed accurately estimates the bandwidth that is required to handle a traffic stream modeled through a Gaussian distribution with mean μ and variance $v(T)$.

> **Remark:** As an aside to the main text of this thesis, we return to provisioning under an $M/G/\infty$ modeling regime, as described

in Section 4.2. Let us *assume* that the input traffic stream follows a Gaussian distribution, in that it is completely described by its first two moments, i.e., mean and variance. Furthermore, assume that the correlation structure of this input traffic follows an $M/G/\infty$ model (with arrival rate λ, flow duration distribution D, $\delta = \mathbb{E}D$, and traffic rate r as before). We can then derive an explicit required bandwidth formula as follows.

First, note that the mean traffic rate is given by $\mu \equiv \lambda \cdot \delta \cdot r$. Secondly, we derive the variance function $v(t)$, as follows. Recall the (log-)moment generating function for the $M/G/\infty$ input model, i.e., (4.5), and compute the second moment (i.e., the variance) by taking the second derivative of the log moment-generating function (with respect to θ) and then substituting 0 for θ. This yields

$$\begin{aligned} v(t) = \lambda\delta \left(\int_0^t x^2 f_{D^r}(x)dx + t^2(1-F_{D^r}(t)) \right) \\ + \lambda \Big(\int_0^t \int_u^t (x-u)^2 f_D(x-u)dxdu \\ + \int_0^t (t-u)^2(1-F_D(t-u))du \Big). \end{aligned}$$

Now rewrite the above expression for $v(t)$ as follows: $v(t) \equiv \lambda \cdot \beta(t,r,D)$, for some function $\beta(\cdot)$ that depends on t, r and D (and, importantly, *not* on λ). Then, by inserting the above definitions for μ and $v(t)$ into the required bandwidth formula for *Gaussian* traffic (4.7), we derive the following required bandwidth formula for the so-called 'Gaussian counterpart' of $M/G/\infty$ input traffic:

$$\begin{aligned} C &= \mu + \frac{1}{t}\sqrt{\left(-2\log\varepsilon\right) v(t)} \\ &= \lambda\delta r + \frac{1}{t}\sqrt{\left(-2\log\varepsilon\right)\cdot\lambda\cdot\beta(t,r,D)} \\ &= \mu + \alpha\cdot\sqrt{\mu}, \quad \text{with } \alpha := \frac{1}{t}\sqrt{\left(-2\log\varepsilon\right)\frac{\beta(t,r,D)}{\delta r}}. \end{aligned}$$

Hence, α depends exclusively on the distribution of the flow duration D, the traffic rate r and the desired performance criterion. Importantly, it does *not* depend on the flow arrival rate λ. A possible practical interpretation of the latter is as follows: when more traffic sources (users) are added, the required bandwidth 'scales' with the aggregated traffic rate (assuming the behavior of individual users does not change) — for more information, see [vdBMvdM$^+$06].

Although the above required bandwidth formula looks nice and simple, it is, however, only applicable in environments where it is indeed possible to properly estimate the distribution of D and r, or to estimate α directly through, for instance, empirical determination (we will do this in Section 4.4). We have seen in Chapter 3 that an easy estimation of D and r is not generally possible, thus we do not advocate this simple formula; we rather prefer a formula based on models that *are* (more or less) generally applicable, such as Gaussian traffic models.

This concludes the more theoretical part of the present chapter. Next, we will validate the bandwidth provisioning formulas derived so far, by comparing their outcome (while estimating the required model parameters) with what can be seen in real traffic. Recall that we actually assumed (and also concluded) that our traffic models are not a perfect fit for real traffic. As a consequence, it is interesting to see to what extent the derived provisioning formulas correspond to the real demands, given that those formulas are based on models that are not perfectly matching the real traffic.

4.4 Empirical validation and alternative formulas

In this section we validate whether the derived provisioning formulas match with measurements of real traffic. In particular, we investigate whether the following formulas hold:

(i) The bandwidth provisioning formula based on the 'Gaussian counterpart' of $M/G/\infty$ input:

$$C = \mu + \alpha\sqrt{\mu}.$$

This is done through measurements taken from an ADSL access network where the access link speeds were, indeed, much smaller than the backbone capacity, yielding a relatively small factor α. See Section 4.4.1.

(ii) The bandwidth provisioning formula based on Gaussian input:

$$C = \mu + \frac{1}{T}\sqrt{(-2\log\varepsilon)\cdot v(T)}.$$

The approach taken here is that μ and $v(T)$ are actually estimated from the packet traces — a rather cumbersome process in real environments, because of the required measurement effort to determine $v(T)$ for small T. See Section 4.4.2.

In the validations introduced above, in total hundreds of packet traces are used. We relate the mean traffic rate in each trace with the peak traffic rate found within the same trace. While doing so, it *appears* that alternative provisioning formulas might be valid as well, from a purely empirical perspective. Some considerations on this topic are presented in Section 4.4.3.

4.4.1 Validation of first bandwidth provisioning formula

The approach taken in the validation of the $M/G/\infty$ input model based provisioning formula is as follows, based on our measurements on various networks:

The packet traces are analyzed, to find rates at which traffic is sent across the link, on a per-second basis. Hence, for every 5 minute interval, 300 traffic rate figures are collected. The average rate of these 300 numbers, equal to the 5 minute average traffic rate, is computed and referred to as μ. In addition we determine p, which denotes the 99th percentile of the 1 second traffic rates.

Note that the above procedure is closely related to our performance criterion (1.2), when T is chosen 1 second, and ε is chosen 1%. One could, then, say that the link capacity should be equal to, (or just above,) the determined peak rate p.

This procedure is repeated for all the packet traces at the various locations, resulting in numerous mean and corresponding peak traffic rate combina-

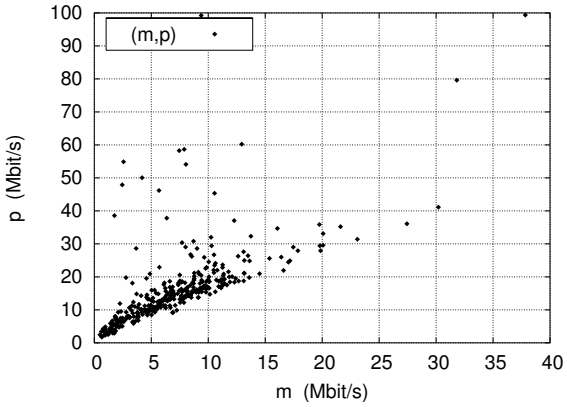


Figure 4.1: Mean rate μ vs. peak rate p for network R (outliers not removed)

tions: (μ, p)-tuples. These tuples are then plotted — see Figure 4.1 for the measurements taken at location R.

Figure 4.1 shows a dense cloud for relatively small values of μ. For larger values of μ, there is less information because such relatively high (sustained) traffic rates are, apparently, less common. It can also be observed that a group of some 10 tuples falls outside (i.e., above) the cloud. The tuples above the cloud are caused by 5 minute intervals in which there are a few seconds with a large throughput rate, which is possible by relatively small flows that have a large throughput rate. Such flows are possible because of the high access link speed at this specific measurement location. These relatively excessively high peak rates are called 'outliers', and ignored in the sequel.

Since the context of our study is bandwidth provisioning, we are not so much interested in finding an α that, for given μ, estimates the *average* p, but rather fits the *maximum* value of p. If 'outliers' are ignored, the cloud in Figure 4.1 is fitted in this respect with $\mu + 4.86 \cdot \mu$. It is noted, however, that this fit deviates from the cloud considerably, because of the 'spreading' of the cloud. Therefore, one could argue that the $C = \mu + \alpha \cdot \mu$ rule would not give an accurate estimation of the required bandwidth capacity (at least in this example).

4.4.2 Validation of second bandwidth provisioning formula

The approach taken in the validation of the Gaussian-model-based provisioning formula is as follows (again, based on our measurements on various networks):

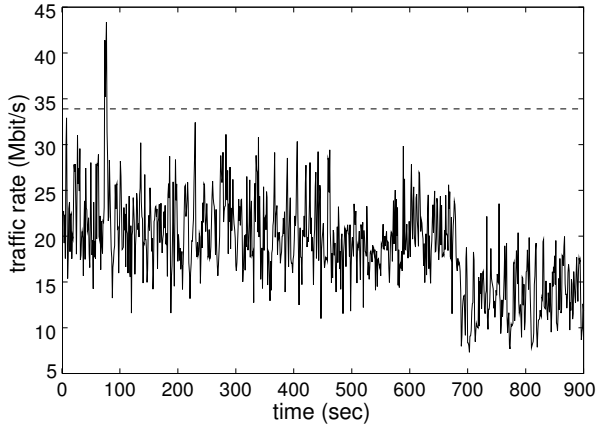
We choose to set the performance criterion parameters $T = 1$ second, and $\varepsilon = 1\%$ — the approach is similar for other settings of these parameters. Now the packet traces are analyzed, and the 15 minute average throughput rate is determined, yielding μ. Furthermore, the variance $v(\cdot)$ of the offered traffic is calculated, based on the traffic offered in each second of the 15 minute interval: $v(1\,\text{sec})$.

The estimates for μ and $v(1\,\text{sec})$ are inserted into the Gaussian provisioning formula (4.7), yielding the estimated required bandwidth capacity C (for the given packet trace, and the choices for T and ε).

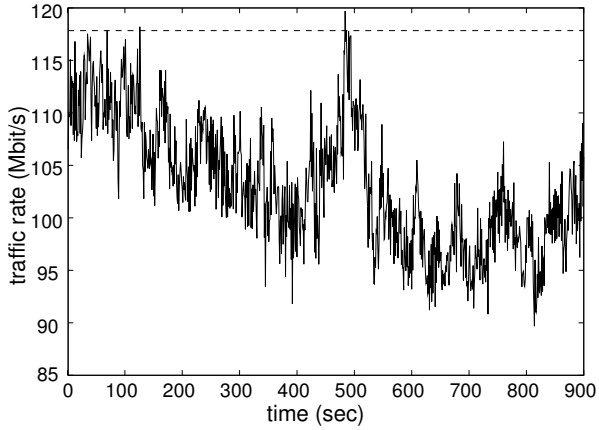
Now we can compare the estimated C with the actual packet trace. We do this by plotting the traffic rates (per second) in one graph together with the estimated C — see Figure 4.2, in which we did this, as an example, for two of our traces (each at a different location).

Figure 4.2 clearly shows that, for these example traces, the Gaussian bandwidth provisioning formula predicts the required bandwidth capacity level quite accurately: sufficiently high to cater for most of the 1 second intervals, in accordance with the prespecified performance criterion (*i.e.*, $T = 1\,\text{sec}, \varepsilon = 1\%$). One might argue that C is slightly over-estimated — this is likely caused by the fact that a bound rather than an exact value is given for the 'overflow probability'. In the context of bandwidth provisioning, however, a slight over-estimation is favorable over under-estimation, as the latter may lead to performance degradation. From a 'Service Level Agreement (SLA) perspective', performance degradation may lead to violation of the SLA, which a network operation will probably try to prevent. Thus, we believe that the apparent slight over-estimation of the required bandwidth capacity is not a problem.

A more extensive validation of the bandwidth provisioning formula for Gaussian traffic, in which hundreds of traffic traces are used, is given in Section 6.2.



(a) Location *R:* $C \approx 34$ Mbit/sec



(b) Location *A:* $C \approx 118$ Mbit/sec

Figure 4.2: Comparing the estimated bandwidth capacity figure C with the actual traffic rates ($T = 1$ sec, $\varepsilon = 0.01$).

4.4.3 Possible alternative provisioning formulas: empirical rules of thumb

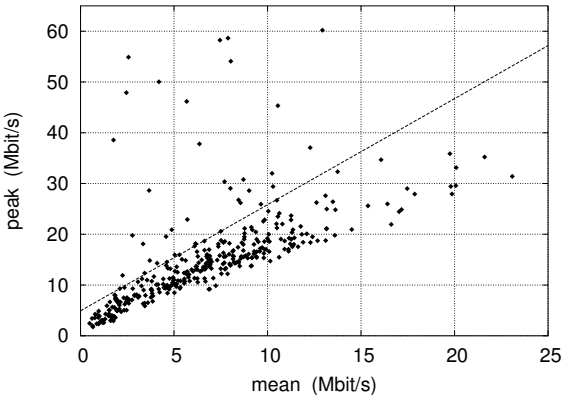
In the previous subsections we have shown both the $M/G/\infty$-based as well as the Gaussian-based provisioning formulas to give reasonably good results in terms of estimating the required bandwidth capacity, although we prefer the Gaussian-based provisioning formula as it is expected to have broader applicability.

As a 'bonus' to this chapter, we now return to the 'mean compared to peak' traffic rate framework from Section 4.4.1. By looking at the cloud of (μ, p)-tuples, one may wonder whether the $p = \mu + \alpha\sqrt{\mu}$ formula is the only adequate rule to describe the relation between mean and peak traffic rates. In this section we investigate another possible relation.

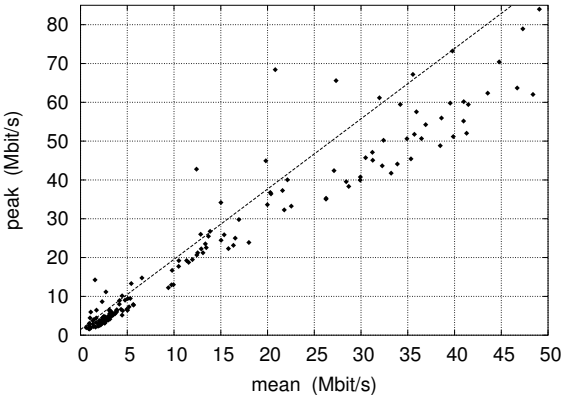
Clearly, any provisioning formula (and hence, any formula relating the long-term mean traffic rate with the peak traffic rate on smaller timescales) would have to incorporate the (relatively) long-term average traffic rate, and then add something to cater for fluctuations (peaks) on smaller timescales. In other words: $p = \mu + X$. Here, X could be various things, e.g., a constant, or a function of (for instance) the long-term average traffic rate μ.

In Figure 4.3, we have plotted (for locations R and C) the respective clouds with (μ, p)-tuples, and added a 'fit' corresponding to the following rule: $p = \mu + \alpha \cdot \mu + \gamma$. Here, the 'baseline' μ can be recognized, plus a function of μ, plus a constant. As Figure 4.3 shows, such a rule appears to be as valid as the previously derived theoretical formulas. Obviously, other alternative formulas could be thought of as well; see [vdMPM+04] for more examples.

It is not possible to give a definitive judgment on what provisioning rule would be the best — even after quantifying their quality (by assessing the 'relative overestimation') there is no clear winner. In practical environments, network managers should therefore look at formulas and/or graphs which suit their needs best.



(a) location *R*, $p = 2.09 \cdot \mu + 4.92$



(b) location *C*, $p = 1.81 \cdot \mu + 1.45$

Figure 4.3: mean rate μ compared with peak rate p: predict p with a multiple of mean μ plus a constant

4.5 Concluding remarks

In this chapter we have derived various bandwidth provisioning rules:

- A formula that gives the (minimally) required bandwidth capacity with no assumptions on the input traffic process but stationarity:

$$C = \inf_{\theta \geq 0} \frac{\log \mathbb{E} \exp\left(\theta A(T)\right) - \log \varepsilon}{\theta T}.$$

  The disadvantage of this formula is that the entire traffic arrival process $A(T)$ has to be known. When additional modeling assumptions are imposed, in that the arrival process is properly parameterized, one can come to more explicit results, like below.

- When Gaussian input traffic is assumed, the above generic formula reduces to:

$$C = \mu + \frac{1}{T}\sqrt{(-2\log \varepsilon) \cdot v(T)}.$$

- When $M/G/\infty$ modeling can be used to accurately describe the input traffic process, the following bandwidth provisioning formula can be derived, using the 'Gaussian counterpart' of the $M/G/\infty$ input model:

$$C = \mu + \alpha\sqrt{\mu},$$

  for an α that depends on just the per-flow characteristics (i.e., rate r and the distribution of the flow duration D, but not on the arrival rate λ), and the desired performance.

  Importantly however, this formula has limited use, because as we have shown in Chapter 3 of this thesis, it is not generally possible to (practically) describe the traffic process within the framework of an $M/G/\infty$ input model. Hence, α would have to be estimated empirically (requiring detailed traffic measurements).

The bandwidth provisioning formula for Gaussian traffic relies on the (long-term) average traffic rate, which is easy to estimate, and the variance of the offered traffic (as well as the performance criterion). Estimation of the variance is cumbersome, as it requires knowledge of the traffic process at (small) timescale T. In the next chapter, however, we present an indirect

approach to determine the sought $v(T)$ *without* requiring measurements at such small timescale T.

5 Burstiness estimation

Burstiness, i.e., the fluctuations in traffic rates, and the average traffic load, are together the dominant factors in determining the required bandwidth capacity for a network link — as was shown in the previous chapters.

Burstiness has to be determined at small timescales that correspond to the perception by a network user of the service level, perhaps 1 second but usually (considerably) smaller. Directly measuring burstiness at such small timescales may be too large a measurement effort. In this chapter, we propose an indirect method for estimating the burstiness, which does not rely on measurements at such small timescales. Instead, our indirect method relies on taking snapshots of the amount of data (also referred to as 'occupancy' or 'contents') in a buffer in front of the 'to-be-sized' network link.

This chapter is organized as follows:

- *Section 5.1 further details the problem of estimating burstiness.*
- *Section 5.2 introduces the theoretical framework upon which we build our indirect method to estimate the burstiness: Gaussian queues. Particular attention is paid to some dimensioning formulas for various types of network resources, which can be derived using the preliminaries on Gaussian queues, additionally to the formulas derived in Chapter 4.*
- *The theoretical framework and associated provisioning formulas lead to our indirect method for determining burstiness. This is explained in Section 5.3.*

– *The indirect method for determining burstiness assumes Gaussian traffic. In Chapter 3 it was shown that although traffic is generally 'fairly Gaussian', it is certainly not '100% Gaussian'. The impact of such a deviation from Gaussian traffic on our burstiness estimation is investigated in Section 5.4, together with an assessment of possible estimation errors.*
– *Section 5.5 concludes.*

5.1 Introduction

As we discussed in Chapter 1, there are in principle three prerequisites for bandwidth provisioning:

1. the traffic offered (in terms of the average load, but also its fluctuations, i.e., burstiness), described through a traffic model;
2. a performance target to be met; and
3. (provisioning) formulas computing the required bandwidth capacity for a given traffic stream and performance target.

In Chapter 4 we have derived and discussed various provisioning formulas for different traffic models, one of them based on Gaussian traffic modeling. In Chapter 3 we have found that real Internet traffic may, in general, fairly well be described through a Gaussian traffic model. Given the performance criterion $\mathbb{P}(A(T) \geq CT) \leq \varepsilon$, recall that the required bandwidth formula for Gaussian input is, cf. (4.7) on page 88:

$$C(T,\varepsilon) = \mu + \frac{1}{T}\sqrt{\left(-2\log\varepsilon\right)\cdot v(T)},$$

with μ denoting the average traffic rate, $v(T)$ the burstiness at timescale T (equal to the variance of the amount of traffic offered per time interval of length T), and T and ε corresponding to the performance criterion of 'link transparency'.

Thus, for proper bandwidth provisioning with prespecified T and ε, and assuming Gaussian traffic, a network manager should estimate μ and $v(T)$.

As we will see in Section 5.2.2, provisioning using some other performance criteria would even require knowledge of the entire variance curve $v(\cdot)$.

As argued earlier, μ can be determined by standard coarse-grained traffic measurements, e.g., polling the IF-MIB counters via SNMP every 5 minutes.

Determining the traffic's burstiness $v(T)$ is more involved. The standard way, i.e., estimating $v(T)$ by measuring the amount of traffic per (disjoint) time interval of length T and then determining the (sample) variance, is what we refer to as the 'direct approach'[1]. An important drawback of this approach, however, is that it requires measurements at timescale T. As T is presumed to be rather small (e.g., at most 1 second, but usually smaller), and accurate SNMP measurements can hardly (if at all) be done at such small timescales, the measurement effort to estimate $v(T)$ may be too substantial to be feasible in practice.

In the present chapter we develop a procedure to estimate $v(\cdot)$ in an indirect way, i.e., not directly from the traffic stream. We propose to do so by coarse-grained sampling of the buffer occupancy, estimating the buffer content distribution and 'inverting' this into the variance curve. Our procedure does, indeed, not only give the variance at timescale T but instead the burstiness at all timescales (of course up to some finite time-horizon). Importantly, this procedure eliminates the need for traffic measurements at small timescales. In this respect, we remark that our proposed procedure is rather counter-intuitive: without doing measurements of the traffic at timescale T, we are still able to accurately estimate $v(T)$. Apparently, the occupancy of the buffer provides enough information to estimate the full curve $v(\cdot)$.

5.2 Gaussian queues

In this section we review some basic principles of Gaussian traffic, and recapitulate the main fundamental (large-deviations) theory for queues with Gaussian input. These principles and fundamental theory are used to develop our novel approach to estimate the burstiness of network traffic. As an aside, we also derive, for this Gaussian setting, a number of provision-

[1] It is noted that the convergence of this estimator could be prohibitively slow when traffic is long-range dependent [Ber94, Ch. I]

ing rules for different network resources. These formulas motivate the need for estimating specific traffic characteristics, viz., the mean rate μ and the entire variance function $v(\cdot)$.

5.2.1 Many sources asymptotics

Consider n independent, statistically identical Gaussian sources. It is assumed that the traffic pattern generated by an individual source corresponds to a Gaussian process with stationary increments[2]. Thus, each source is completely characterized by its mean traffic rate $\overline{\mu}$, and its variance function $\overline{v}(t)$, for $t \geq 0$[3].

Now suppose that the n sources feed into a queue with capacity C_q[4], and apply the scaling $C_q \equiv nc_q$. Consider the probability that the queue length (or buffer content — we use these terms interchangeably) exceeds some threshold nb, i.e., $\mathbb{P}(Q_n > nb)$. It is well-known (see, e.g., [DM03a]) that the stationary queue length, say Q_n has the same distribution as the maximum of the corresponding 'free-buffer process':

$$\mathbb{P}(Q_n > nb) = \mathbb{P}\left(\sup_{t>0}\left(\sum_{i=1}^{n} A_i(t) - nc_q t\right) > nb\right), \tag{5.1}$$

with $A_i(t)$ denoting the amount of traffic generated by the ith source in an interval of length $t \geq 0$, $\mathbb{E}A_i(t) = \overline{\mu}t$, and $\mathrm{Var}A_i(t) = \overline{v}(t)$.

Now, we use the above in the following fundamental result that can be found in, e.g., [AMN02]:

[2]The term 'stationary increments' means, in the present context, that the distribution of the offered traffic $A(T)$ is fixed for each interval of length T. Hence, the distribution only depends on the length T of the interval, and not on 'the position' of the interval. In other words: for any t_0, $A(t_o + T) - A(t_0)$ has the same distribution as $A(T)$.

[3]Note that in the sequel, μ refers the aggregated mean rate of multiple sources, whereas the overlined version refers to the mean rate of a single traffic source, and likewise for v.

[4]The capacity C_q of a queue refers to the rate at which the queue is drained, i.e., C_q bits per second. In this thesis, unless otherwise noted, we assume that the queue itself has an infinite size.

Lemma 5.1: *To exclude certain degenerate cases, suppose that there is an $\alpha < 2$ such that $\overline{v}(t)/t^{\alpha} \to 0$ for $t \to \infty$. Then, for any $b > 0$, and $c_q > \overline{\mu}$,*

$$\begin{aligned} I(b) &:= -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}(Q_n \geq nb) \\ &\equiv -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\exists t > 0 : \sum_{i=1}^{n} A_i(t) - nc_q t \geq nb\right) \\ &= \inf_{t>0} \frac{(b + (c_q - \overline{\mu})t)^2}{2\overline{v}(t)}. \end{aligned}$$

In Lemma 5.1, $I(b)$ can be seen as the exponential decay rate of the probability that the queue length Q_n exceeds some threshold nb, as a function of b. In other words, $\mathbb{P}(Q_n \geq nb) \approx e^{-n \cdot I(b)}$. Lemma 5.1 serves as a source of inspiration in the remainder of this section.

Hence, Lemma 5.1 shows that there is some relation between the probability distribution of the queue length, and the variance of the traffic. We will further detail this relation, and exploit it in our methodology to indirectly estimate the variance, which will be further described in the remainder of this section and in Section 5.3.

Lemma 5.1 holds for the system 'scaled by n', but gives rise to an approximation for the 'unscaled' situation, see also for instance [AMN02], [FTD03, Eq. 3]. The 'unscaled' situation corresponds to our notion of traffic streams with 'some' aggregation (in terms of users). With $B \equiv nb$, consider the probability that the buffer content Q exceeds B, i.e., $\mathbb{P}(Q > B)$. We 'unscale' by specifying $\mu \equiv n\overline{\mu}$ as the aggregate mean, and $v(t) \equiv n\overline{v}(t)$ as the aggregate variance[5]. We then rewrite Lemma 5.1 as follows, to get the variant of this lemma for aggregated traffic. First define

$$I(B) := \inf_{t>0} \frac{\left((C_q - \mu)t + B\right)^2}{2v(t)}. \tag{5.2}$$

[5] It is noted that these definitions are in line with the 'black-box paradigm' used throughout this thesis.

In the next few steps, we will develop the relation between $\mathbb{P}(Q > B)$ and $I(B)$. We start with rewriting the right-hand side of (5.1) for the unscaled situation:

$$\mathbb{P}\left(\sup_{t>0}\left(A(t) - C_q t\right) > B\right).$$

It is noted that the above supremum over t can also be seen as a union of events:

$$\mathbb{P}\left(\sup_{t>0}\left(A(t) - C_q t\right) > B\right) \equiv \mathbb{P}\left(\bigcup_t \left\{A(t) - C_q t > B\right\}\right).$$

We may now apply the 'principle of the largest term', which says that the probability of a union of (rare) events can be accurately approximated by the probability of the most likely event. Evidently, we arrive at:

$$\sup_{t>0} \mathbb{P}\big(A(t) > C_q t + B\big). \tag{5.3}$$

We can now perform the same procedure as followed in Chapter 4 on page 81, i.e., applying 'Chernoff'. 'Chernoff' gives an upper bound on (5.3), but the upper bound in many cases is remarkably accurate. We thus arrive at:

$$\mathbb{P}\left(\sup_{t>0}\left(A(t) - C_q t\right) > B\right) \approx \sup_{t>0} \inf_{\theta \geq 0} \mathbb{E} e^{\theta A(t) - \theta C_q t - \theta B}.$$

Now recall from (4.6) on page 87 that $\mathbb{E}\exp(\theta A(t)) = \exp\left(\theta \mu t + \frac{1}{2}\theta^2 v(t)\right)$. We can exploit this to compute the above infimum over $\theta \geq 0$ explicitly. Start with inserting $A(t)$ as in (4.6) into the above Chernoff bound, which gives after some rearranging of terms[6]:

$$\inf_{\theta \geq 0} \mathbb{E} e^{\theta A(t) - \theta C_q t - \theta B} = \exp\left(-\sup_{\theta \geq 0}\left(\theta\left(B + C_q t\right) - \theta \mu t - \frac{1}{2}\theta^2 v(t)\right)\right).$$

[6]Besides rearranging, also the infimum over θ is changed to a supremum over θ. This is possible, as by definition the following holds for a non-empty set of values S: $\inf(S) = -\sup(-S)$.

To compute the supremum, we now have to maximize a quadratic function of θ, and the first order condition (i.e., take the derivative of this function with respect to θ, and setting this equal to zero) reads

$$B + C_q t - \mu t = \theta v(t),$$

such that the optimizing θ is $\left(B + C_q t - \mu t\right) / v(t)$. Plugging this into our objective function, we arrive at:

$$\inf_{\theta \geq 0} \mathbb{E} e^{\theta A(t) - \theta C_q t - \theta B} = \exp\left(\frac{-\left(B + C_q t - \mu t\right)^2}{2v(t)} \right).$$

This finally leads to the following approximation of the overflow probability:

Approximation 5.1 *For any $B > 0$, and $C_q > \mu$,*

$$\begin{aligned} \mathbb{P}(Q > B) \approx e^{-I(B)} &= \sup_{t>0} \exp\left(\frac{-\left(B + (C_q - \mu) t\right)^2}{2v(t)} \right) \\ &= \exp\left(- \inf_{t>0} \frac{\left(B + (C_q - \mu) t\right)^2}{2v(t)} \right). \end{aligned} \tag{5.4}$$

Hence, Approximation 5.1 gives the buffer content distribution $\mathbb{P}(Q > B)$ as a function of the burstiness of the traffic as in $v(t)$, and other variables. The question that remains is whether it is possible to 'invert' this function, i.e., to write $v(t)$ as a function of $\mathbb{P}(Q > B)$, etc. We will show, in Section 5.3, that such an 'inversion' is indeed possible, and gives an 'indirect' way to estimate the burstiness of network traffic.

5.2.2 Provisioning formulas

One of the major tasks in network management is the provisioning of resources: choose the link capacity and/or buffer size such that some prespecified performance criterion is met. In this thesis we so far focused on achieving 'link transparency'. Approximation 5.1 provides us with a tool to dimension, next to the discussed link transparency, other resources as well.

Note that Section 5.2.2 is to be seen as an aside from the main text of this thesis. We recall provisioning for link transparency, and discuss provisioning of a buffered link and just a buffer itself:

Dimensioning for 'link transparency'

Suppose the goal is to provision the link such that the probability of exceeding the capacity C_q for a period of length T is smaller than ε, with $B = 0$ (as the performance criterion does not take into account presence of a buffer). Hence we have to find the smallest $C_q = C_q(T, \varepsilon)$ such that

$$\exp\left(-\frac{\big((C_q - \mu)T\big)^2}{2\nu(T)}\right) \leq \varepsilon,$$

cf. Approximation 5.1. It is readily checked that this yields

$$C_q(T, \varepsilon) = \mu + \frac{1}{T}\sqrt{\big(-2\log\varepsilon\big)\,\nu(T)},$$

cf. (4.7) on page 88.

> **Remark:** It is noted that, in practice, there will always be a buffer. The above provisioning rule for an unbuffered link should be seen in the light that in such cases a buffer is to be used as a last resort — the objective is to (almost) never use it, but only as a last resort. This holds for timescale T; hence, one could say that in these cases, a buffer is used to absorb bursts of traffic at timescales smaller than T.

Dimensioning of a buffered resource

In the setting of provisioning rule (4.7) we considered an unbuffered resource. In practice, however, network elements are often equipped with a queue, to absorb traffic rate fluctuations. If the router has a queue of size B, and suppose we wish to provision the capacity, we have to find the minimal

$C_q = C_q(\varepsilon)$ such that (5.4) in Approximation 5.1 is below ε. Hence, we are searching for:

$$\min\left\{C_q \mid \forall\, t>0 : \exp\left(-\frac{\left(B+(C_q-\mu)\,t\right)^2}{2\,v(t)}\right) \le \varepsilon\right\} \ .$$

To find this minimal C_q, we rearrange terms as follows:

$$\min\left\{C_q \mid \forall\, t>0 :\ \left(B+(C_q-\mu)\,t\right)^2 \ge -2\cdot v(t)\cdot\log\varepsilon\right\},$$

and, hence,

$$\min\left\{C_q \mid \forall\, t>0 : C_q \ge \mu+\left(\frac{1}{t}\sqrt{\left(-2\log\varepsilon\right)v(t)}-\frac{B}{t}\right)\right\}.$$

Clearly, the contribution of μ to the minimal C_q is not influenced by t. Thus, we find that:

$$C_q(\varepsilon) = \mu+\inf_{t>0}\left(\frac{1}{t}\sqrt{\left(-2\log\varepsilon\right)v(t)}-\frac{B}{t}\right). \tag{5.5}$$

Like in the case of the unbuffered link, the bandwidth required decreases in ε. Moreover, it also decreases in B: the larger the queue, the better traffic fluctuations can be absorbed by the buffer, and hence less link capacity is needed.

It is noted that (5.5) requires knowledge of the entire variance function $v(\cdot)$, instead of only the variance at timescale T as in the case of the unbuffered link — this is caused by the fact that the timescale with the largest contribution to the buffer occupancy is not fixed.

Buffer dimensioning

Similarly to the procedure above, the minimum required buffer $B = B(\varepsilon)$ can be determined:

$$B(\varepsilon) = \inf_{t>0}\left(\sqrt{\left(-2\log\varepsilon\right)v(t)}-\left(C_q-\mu\right)t\right) \ . \tag{5.6}$$

Note that $B(\varepsilon)$ decreases in ε and C, as can be expected. Also, again, knowledge of the entire variance function is required.

We finish this section on Gaussian queues with an example, in which we use artificial traffic to derive explicit provisioning figures using the formulas presented above, to demonstrate how the formulas work. We focus on the relevant case of fBm traffic (see also Chapter 3), i.e., Gaussian traffic with $v(t) = \sigma^2 t^{2H}$ (take for ease $\sigma = 1$).

Example: *(Provisioning for fBm traffic)* Straightforward computations give for (4.7):

$$C_q(T,\varepsilon) = \mu + \frac{\sqrt{-2\log\varepsilon}}{T^{1-H}} \ .$$

When computing $C_q(\varepsilon)$ in (5.5), the optimizing t is given by

$$B^{1/H}(1-H)^{-1/H}\sqrt{-2\log\varepsilon}^{-1/H} \ ,$$

yielding

$$C_q(\varepsilon) = \mu + \sqrt{-2\log\varepsilon}^{1/H}\left(\frac{1-H}{B}\right)^{1/H-1} H \ .$$

In buffer provisioning rule (5.6) the optimizing t is given by

$$(\delta H)^{1/(1-H)}(C_q-\mu)^{-1/(1-H)} \ ,$$

such that

$$B(\varepsilon) = \left(\frac{\sqrt{-2\log\varepsilon}H}{(C_q-\mu)^H}\right)^{1/(1-H)} \frac{1-H}{H} \ .$$

An important conclusion from this section is that the above provisioning formulas indicate that it is of crucial importance to have accurate estimates of the average traffic rate μ, as well as the variance curve $v(\cdot)$ (i.e., $v(t)$ as a function of $t \geq 0$); having these at our disposal, we can find the required bandwidth capacity or buffer size.

In the next section we present a method to find $v(\cdot)$ that *does not rely on detailed measurements at (small) timescale T*.

5.3 An indirect method to estimate burstiness

This section presents a powerful alternative to the 'direct method' of determining burstiness; we refer to it as the *inversion approach* (or *indirect method*) as it 'inverts' the buffer content distribution to the variance curve. We rely on the many-sources framework from Section 5.2.1.

5.3.1 Derivation of the inversion formula

Recall the 'unscaled' variant of Lemma 5.1, i.e., cf. (5.2):

$$I(B) := \inf_{t>0} \frac{\big((C_q - \mu)\, t + B\big)^2}{2\nu(t)},$$

where $I(B)$ can be seen as the exponential decay rate of the probability that the queue length Q exceeds some value B, i.e., $\mathbb{P}(Q > B)$: Approximation 5.1 says that $\mathbb{P}(Q > B) \approx \exp(-I(B))$.

Now define t_B as the most likely timescale (or epoch) at which such an overflow occurs, for a given buffer size $B > 0$:

$$t_B := \arg\inf_{t>0} \frac{(B + (C_q - \mu)\, t)^2}{2\nu(t)}.$$

It was noted in, e.g., [AMN02, Man04, MK01], that t_B is not necessarily unique. Define the set $\mathcal{T}$ as follows, containing all t_Bs:

$$\mathcal{T} := \{t > 0 \mid \exists B > 0 : t = t_B\}.$$

The next step is to hypothesize that we may 'invert' the above formula in which $I(B)$ is defined as a function of variance $\nu(t)$ (and others), in such a way that $\nu(t)$ is written as a function of $I(B)$. The following theorem gives, for any $t > 0$, an upper bound on the variance $\nu(t)$, for given $I(B)$, and presents conditions under which this upper bound is tight.

Theorem 5.2:

1. *For any* $t > 0$,

$$v(t) \leq \inf_{B>0} \frac{\left(B + (C_q - \mu)\, t\right)^2}{2I(B)} \quad . \tag{5.7}$$

2. *There is* equality *in* (5.7) *for all* $t \in \mathcal{T}$
3. *If* $2v(t)/v'(t) - t$ *grows from 0 to* ∞ *when* t *grows from 0 to* ∞, *then* $\mathcal{T} = (0, \infty)$.

PROOF: Clearly, due to (5.2), for all $B > 0$ and $t > 0$, we have that

$$I(B) \leq \frac{\left(B + (C_q - \mu)\, t\right)^2}{2v(t)}.$$

Hence also, for all $B > 0$ and $t > 0$:

$$v(t) \leq \frac{\left(B + (C_q - \mu)\, t\right)^2}{2I(B)} \quad ,$$

which implies claim 1 immediately.

Now consider a $t \in \mathcal{T}$. Then there is a $B = B_t > 0$ such that

$$I(B) = \frac{\left(B + (C_q - \mu)\, t\right)^2}{2v(t)}.$$

We thus obtain claim 2.

Now consider claim 3. We have to prove that for all $t > 0$ there is a $B > 0$ such that $t = t_B$. Evidently, t_B solves $2v(t)(C_q - \mu) = (B + (C_q - \mu)\, t)\, v'(t)$, or, equivalently,

$$B = B_t := \left(2\, \frac{v(t)}{v'(t)} - t\right)(C_q - \mu). \tag{5.8}$$

Hence, it is sufficient if B_t in the right hand side of (5.8) grows from 0 to ∞ when t grows from 0 to ∞.

Remarkably, it apparently holds that if one knows the probability distribution of the queue length, one can derive the probabilistic properties of the input process to that queue. Thus, Theorem 5.2 gives, loosely speaking, that

for Gaussian sources the buffer content distribution uniquely determines the variance function. This property is exploited in the following heuristic, for which we coin the term 'inversion formula':

Approximation 5.3 *The following estimate of the function $v(t)$ (for $t > 0$) can be made using the buffer content distribution:*

$$v(t) \approx \inf_{B>0} \frac{(B + (C_q - \mu)\, t)^2}{-2 \log \mathbb{P}(Q > B)}. \tag{5.9}$$

Hence, if we can estimate $\mathbb{P}(Q > B)$, then 'inversion formula' (5.9) can be used to retrieve the variance; notice that the infimum can be computed for any t, and consequently we get an approximation for the entire variance curve $v(\cdot)$ (of course up to some finite horizon). These ideas are exploited in the procedure described in the next section.

5.3.2 Procedure to estimate burstiness through the buffer occupancy

In this section, we show how the theoretical results of the previous subsection can be used to estimate $v(\cdot)$. We first propose an algorithm for estimating the (complementary) buffer content distribution $\mathbb{P}(Q > B)$ (in the sequel abbreviated to BCD), such that, by applying Approximation 5.3, the variance curve $v(\cdot)$ can be estimated.

Inversion procedure

The inversion procedure consists of two steps: (1) determining the BCD, and (2) 'inverting' the BCD to the variance curve $v(\cdot)$ by applying Approximation 5.3. We propose the following algorithm:

Algorithm 5.4

1. *Collect 'snapshots' of the buffer contents: $q_1, \ldots, q_N$; here q_i denotes the buffer content as measured at time $\tau_0 + i\tau$, for some $\tau > 0$. Estimate the BCD by the empirical distribution function of the q_i, i.e., estimate $\mathbb{P}(Q > B)$ by*

$$\phi(B) = \frac{\#\{i : q_i > B\}}{N}.$$

2. *Estimate $v(t)$, for any $t \geq 0$, by*

$$\inf_{B>0} \frac{(B + (C_q - \mu)\, t)^2}{-2 \log \phi(B)}. \tag{5.10}$$

In the above algorithm, snapshots of the buffer content are taken at a constant frequency. To get an accurate estimate of the BCD, both τ and N should be chosen sufficiently large. We come back to this issue in Section 5.4. Notice that we chose a fixed polling frequency (i.e., τ^{-1}) in our algorithm, but this is not strictly necessary; the BCD-estimation procedure obviously still works when the polling epochs are not (exactly) equally spaced. One could also argue that a specific polling scheme may be preferable, e.g., if the polling epochs are Poisson-distributed, this would eliminate possible effects of period traffic on the resulting BCD-estimation (because of Poisson-Arrivals-See-Time-Averages (PASTA) arguments).

5.3.3 Demonstration of the inversion procedure

We now demonstrate the inversion approach of Algorithm 5.4 through a simulation with synthetic input, i.e., traffic generated according to some stochastic process. Concentrating on slotted time[7], we focus on the (practically relevant) case of fBm input[8], but we emphasize that the procedure could be followed for any other stochastic process. The simulation of the queue fed by the synthetic input yields an estimate for the BCD; this estimated BCD is then 'inverted' to obtain the estimated variance curve $v(\cdot)$. Finally we compare, for fBm, our estimation for $v(\cdot)$ with the actual variance curve, yielding a first impression of the accuracy of our approach (more detailed validation follows in the next section on error analysis, and in Chapter 6).

We have simulated fBm by a *fractional Brownian motion simulator* [Die] (based on Davis and Harte's circulant method, see [DM03b, Die] for more information). The traffic stream is fed into a queue with link rate C_q. The buffering dynamics are simulated follows:

[7] Slotted time means that time is divided into discrete intervals.

[8] See Chapter 3, page 51, for a description of fBm

Algorithm 5.5 *Simulation of the buffer dynamics:*

1. *Using the fBm simulator we generate fBm input, with a specific Hurst parameter $H \in (0,1)$. This yields a list $A_1, \ldots, A_Z$, for some $Z \in \mathbb{N}$, where A_j denotes the amount of traffic offered in the jth slot.*
2. *The list $A_1, \ldots, A_Z$ is used to simulate the buffer dynamics. This is done recursively:*
$$Q_{j+1} := \max\{Q_j + A_j - C_q, 0\},$$
where Q_j denotes the amount of contents in the buffer at the beginning of slot j.
3. *The buffer content Q_j is observed every τ slots, which results in $N = Z/\tau$ snapshots q_i of the buffer content. These snapshots are used to estimate $\mathbb{P}(Q > B)$, as described in Algorithm 5.4.*

In (standard) fBm, the average traffic rate μ equals 0. Trivially, fBm with non-zero 'drift' μ can be simulated by replacing the list $A_1, \ldots, A_Z$ by $A_1 + \mu, \ldots, A_Z + \mu$.

In this demonstration of the inversion procedure, we generate an fBm traffic trace with Hurst parameter $H = 0.7$ (motivated by our earlier observation that Hurst values between 0.6 and 0.8 are generally found for several types of real traffic) and length $Z = 2^{24}$ slots. The link capacity C_q is set to 0.8, and we take snapshots of the buffer content every $\tau = 2^7 = 128$ intervals.

We can then estimate the BCD. A plot is given in Figure 5.1; for presentation purposes, the (natural) logarithm of the BCD, i.e., $\log \mathbb{P}(Q > B)$, is plotted.

It can be seen that the BCD in Figure 5.1 is 'less smooth' for larger values of B. This is due to the fact that large buffer levels are rarely exceeded, leading to less accurate estimates.

Using the estimated BCD, the variance $v(t)$ for t equal to the powers of 2 ranging from 2^0 to 2^7 is estimated. The resulting variance curve is shown in Figure 5.2 ('inversion approach'). The minimization (over B) in (5.10) was done by straightforward numerical search techniques.

To get an impression of the accuracy of the inversion approach, we have also plotted in Figure 5.2 the variance curve as can be estimated directly from the synthetic traffic trace (i.e., the 'direct approach'), as well as the real variance function for fBm traffic, i.e., $v(t) = t^{2H}$.

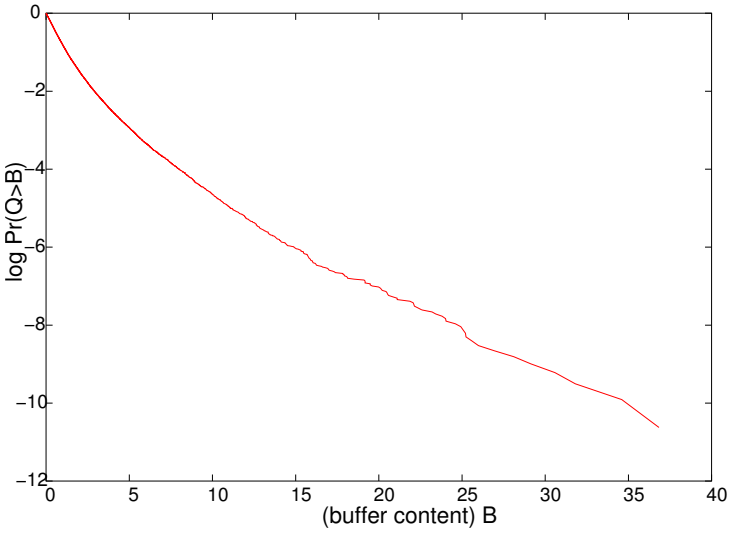


Figure 5.1: Sample buffer content distribution

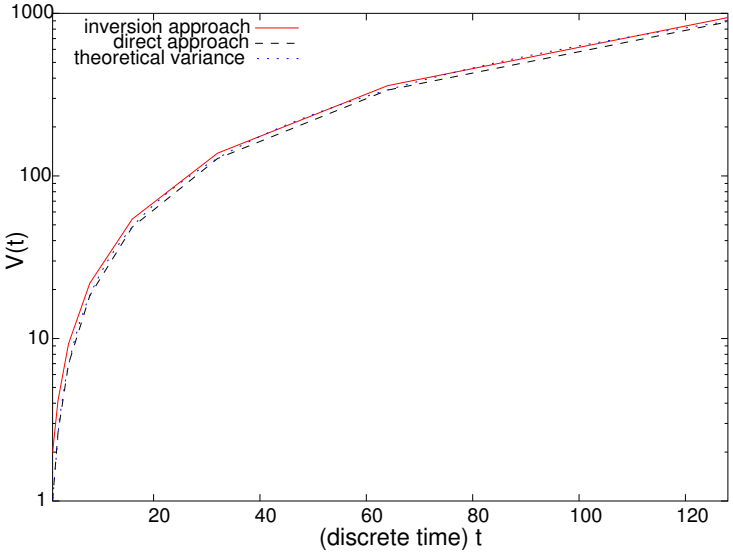


Figure 5.2: Sample variance curves

Figure 5.2 shows that the three variance curves are remarkably close to each other. This confirms that the inversion approach is an accurate way to estimate the burstiness. We note that the graph shows that the inversion approach slightly overestimates the variance; hence, when using this curve for provisioning purposes, for instance by applying rules (5.5) or (5.6), this would result in conservative numbers. An extensive validation of the use of the inversion approach for provisioning purposes is performed in Chapter 6 of this thesis.

5.4 Error analysis of the inversion procedure

In the previous section the inversion approach was demonstrated. It was shown to perform well for fBm with $H = 0.7$, under a specific choice of N and τ. Evidently, the key question is whether the procedure still works under other circumstances. To this end, we first identify the three possible sources of errors:

- The inversion approach is based on the *approximation* (5.4).
- $\mathbb{P}(Q > B)$ is *estimated*; there could still be an estimation error involved. In particular, we wonder what the impact of the choice of N and τ is.
- The procedure *assumes* perfectly Gaussian traffic, although real network traffic may not be (accurately described by) Gaussian (as also seen in Chapter 3).

We will now quantitatively investigate the impact of each of these errors on our 'indirect approach'. These investigations are performed through simulation as outlined in the previous section.

5.4.1 Approximation of the buffer content distribution

In (5.4) an approximation of the BCD is given. As the inversion approach is based on this approximation, evidently, errors in (5.4) might induce errors in the inversion. This motivates the assessment of the error made in (5.4).

We first determine the infimum in the right-hand side of (5.4), which we consider as a function of B. In line with the previous section, we choose

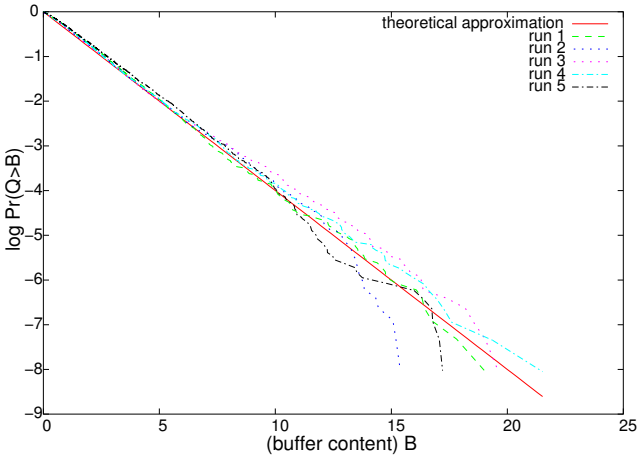


Figure 5.3: $\mathbb{P}(Q > B)$ and theoretical approximation ($H = 0.5$)

fBm input: $\mu = 0$ and $v(t) = t^{2H}$. Straightforward calculations now reveal that we can rewrite (5.4), viz.:

$$\log \mathbb{P}(Q > B) \approx -\frac{1}{2} \cdot \left(\frac{B}{1-H}\right)^{2-2H} \cdot \left(\frac{C_q}{H}\right)^{2H} .$$

We verify how accurate the approximation is, for two values of H: the purely Brownian case $H = 0.5$, and a case with long-range dependence $H = 0.7$. Several runs of fBm traffic are generated (with different random seeds), 2^{24} slots of traffic per run. We then simulate the buffer dynamics. For $H = 0.5$ we choose link rate $C_q = 0.2$, for $H = 0.7$ we choose $C_q = 0.8$; these choices C_q are such that the queue is sufficiently often non-empty (in order to obtain a reliable estimate of the BCD); see also Chapter 6 for more discussion on choosing an appropriate C_q.

Figures 5.3 and 5.4 show for the various runs the approximation of the BCD, as well as their theoretical counterpart. It can be seen that, in particular for small B the empirically determined BCD almost perfectly fits the theoretical approximation[9].

[9] It is noted that the simulator uses slotted time, whereas (5.3) involves an optimization over continuous time. This actually means that we should take the minimum over $t \in \mathbb{N}$ in (5.3), instead of $t \in \mathbb{R}$. Computation of the minimum over $\mathbb{N}$ shows that this does not have significant impact.

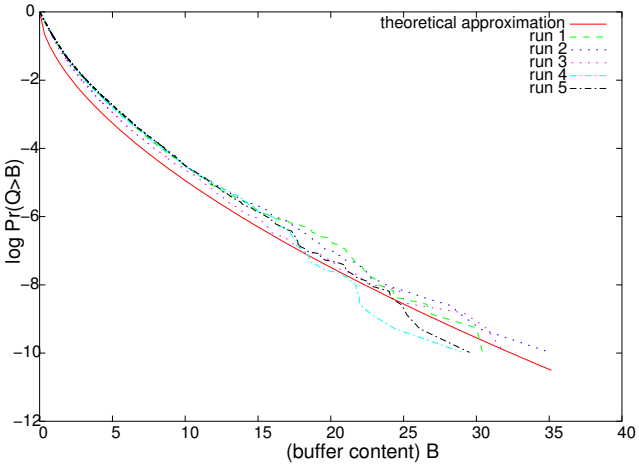


Figure 5.4: $\mathbb{P}(Q > B)$ and theoretical approximation ($H = 0.7$)

5.4.2 Estimation of the buffer content distribution

A second possible error source in our inversion approach, relates to the estimation of the BCD. As we estimate the BCD on the basis of snapshots of the buffer content, there will be some error involved. The impact of this error is the subject of this subsection. It could be expected that the larger N (more observations) and the larger τ (less correlation between the observations), the better the estimate.

First, we investigate the impact of N. The simulator is run as in previous cases (with $H = 0.7$), with the difference that we only use the first x% of the snapshots samples to determine $\mathbb{P}(Q > B)$. Figure 5.5 shows the estimation of the buffer content distribution, for various x ranging from 0.1 to 100. The figure shows that, in particular for relatively small B, a relatively small number of observations suffice to get an accurate estimate of the BCD.

Second, we investigate the impact of the interval length between two consecutive snapshots τ. One might expect that the more often the buffer occupancy is polled, the closer the resulting buffer content distribution would look like the theoretical approximation. Note, however, that when the snapshots are taken close together, the observations may be highly correlated due to the long-range dependence of the simulated fBm traffic, which might (negatively) affect the accuracy of the estimate. Figure 5.6 shows the determined BCD for τ ranging from observing every 32 to every 8192 slots. It can

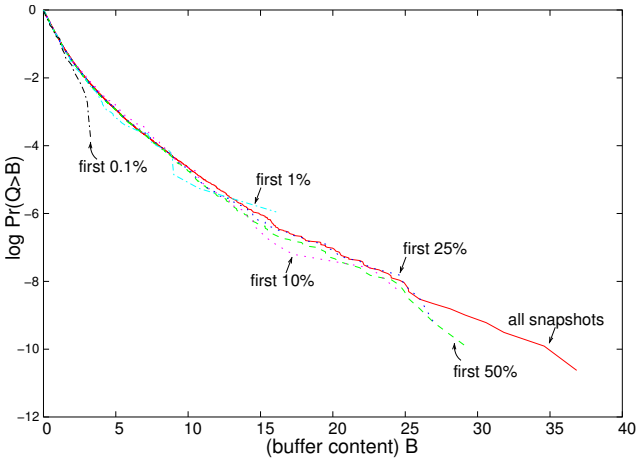


Figure 5.5: Comparing $\mathbb{P}(Q > B)$ for various trace lengths

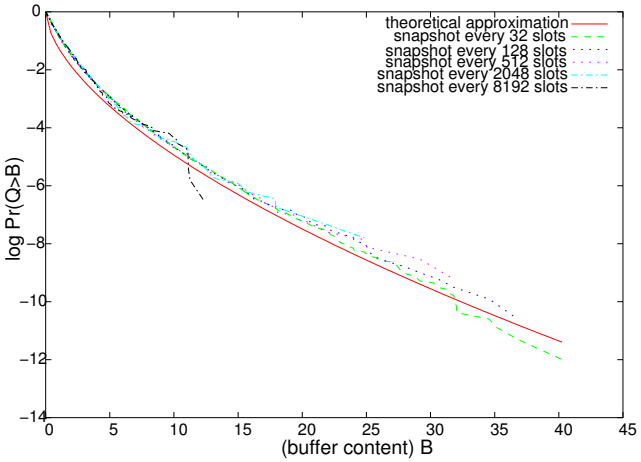


Figure 5.6: Comparing $\mathbb{P}(Q > B)$ for various polling intervals

be seen that, particularly for small B the fit is quite good, even when the buffer content is polled only relatively rarely.

5.4.3 The impact of the Gaussianity assumption

Approximation (5.4) explicitly assumes that the traffic process involved is Gaussian. Various measurement studies find that real network traffic on the Internet is (accurately described by) Gaussian, see, e.g., [LTWW94] and Chapter 3 of thesis, but the fit is of course not perfect. In this subsection we investigate the impact of the Gaussianity assumption on the inversion, i.e., we test how the procedure works for traffic that is not (perfectly) Gaussian (which is better 'in line with reality', see again Chapter 3).

We study the impact of non-Gaussianity as follows. Mix the trace $A_{i\_[\mathsf{fBm}]}$ generated by the fBm simulator with traffic $A_{i\_[\mathsf{alt}]}$ from an alternative (non-Gaussian) input stream:

$$A_i := \alpha \cdot A_{i\_[\mathsf{fBm}]} + (1-\alpha) \cdot A_{i\_[\mathsf{alt}]},$$

where $\alpha \in [0,1]$ corresponds to the fraction of fBm traffic in the mixture. The resulting traffic stream is fed into the queue, cf. Algorithm 5.5. Clearly, by definition, the variance of the traffic mixture is (as we consider the [fBm] and [alt] components of the mixtures to be independent of each other, and hence, having zero covariance):

$$v(t) = \alpha^2 v_{[\mathsf{fBm}]}(t) + (1-\alpha)^2 v_{[\mathsf{alt}]}(t). \tag{5.11}$$

For $\alpha = 1$ we are in the pure-Gaussian case, of which we have seen that the inversion procedure performed well. We now vary α from 1 to 0, to see the impact of the non-Gaussianity.

The alternative input model that we choose here is an M/G/∞ input model, inspired by, e.g., [AMN02]. In the M/G/∞ input model, jobs arrive according to a Poisson process. The job durations are i.i.d., and during their duration each job generates traffic at a constant rate r. In line with measurements studies, we choose Pareto(β) jobs, obeying the distribution function

$$F_D(x) = 1 - 1/(x+1)^{\beta}.$$

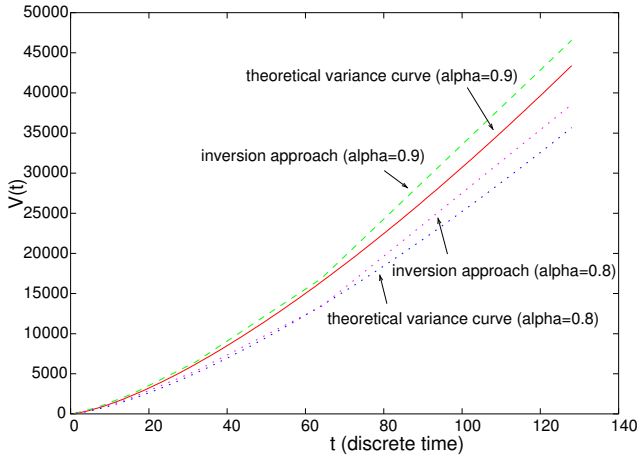


Figure 5.7: Variance curves for Gaussian/non-Gaussian traffic mixtures, for $\alpha = 0.8$ and $\alpha = 0.9$.

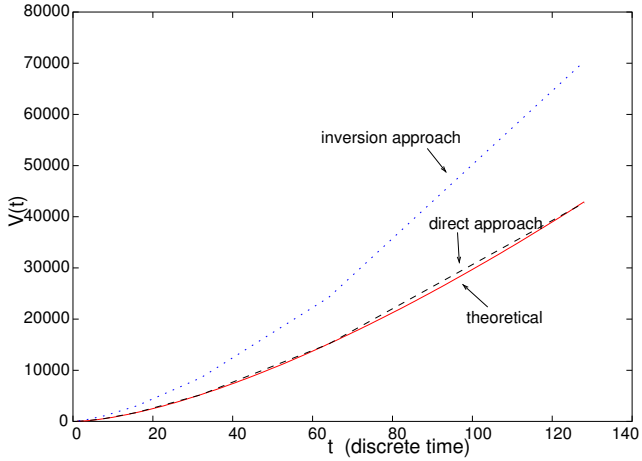


Figure 5.8: Variance curves for Gaussian/non-Gaussian traffic mixture, for $\alpha = 0$

As the objective is to assess the impact of varying the parameter α, we have chosen to select the parameters of the M/G/∞ input model such that the processes $A_{i\_[\mathsf{fBm}]}$ and $A_{i\_[\mathsf{alt}]}$ are 'compatible', in that their mean μ and variance $v(\cdot)$ are 'similar'. This has been done as follows:

- The mean of the described M/G/∞ input model is given by $\mu_{[\mathsf{alt}]} = \lambda r/(\beta-1)$. We choose $\lambda = 10$ and $r = 1$ for ease; the other involved parameters will be used to achieve the desired 'compatibility'. The mean of the original fBm traffic model is $\mu_{[\mathsf{fBm}]} = 0$; however, as argued earlier, we may add a drift of $\mu_{[\mathsf{alt}]}$ to the $A_{i\_[\mathsf{fBm}]}$-values to ensure that $\mu_{[\mathsf{alt}]} = \mu_{[\mathsf{fBm}]}$.
- In earlier work, see e.g. [MSS05], an exact formula for the variance curve $v_{[\mathsf{alt}]}(t)$ has been derived. It is not possible to achieve the desired 'compatibility' of the variance at all timescales. As long-range dependence is mainly a property of long timescales, we choose to focus on these. For larger timescales, the variance $v_{[\mathsf{alt}]}(t)$ from [MSS05] roughly looks like, assuming $\beta \in (1,2)$,

$$v_{[\mathsf{alt}]}(t) \approx r^2 \lambda \frac{2}{(3-\beta)(2-\beta)(\beta-1)} t^{3-\beta}.$$

  The variance of the original fBm traffic model is also known: $v_{[\mathsf{fBm}]}(t) = \sigma^2 t^{2H}$. Assuming $H = 0.7$ and sufficiently large t, we can now determine the remaining parameters: $\beta \approx 1.6$, $\sigma \approx 7.72$, and $\mu \approx 16.67$. The variance function of the traffic mixture may now be computed using (5.11).

The next step is to run, for different values of α, the simulation, and to determine the theoretical variance curve of (5.11), and to perform the inversion procedure of Algorithm 5.4.

In Figure 5.7 we focus on the 'nearly-Gaussian' cases $\alpha = 0.8$ and $\alpha = 0.9$, which are plotted together with their theoretical counterparts. The figure shows that the presence of non-Gaussian traffic has some, but no crucial impact on our inversion procedure.

We also consider the (extreme) case of $\alpha = 0$, i.e., no Gaussian traffic at all, to see if our inversion procedure still works. In Figure 5.8 the various variance

curves are shown: the theoretical curve, the curve based on the 'direct approach', as well as the curve based on the inversion approach. Although not a perfect fit, the curves look similar and still relatively close to each other (but, of course, the fit is worse than for $\alpha = 0.8$ and 0.9). Note that the non-Gaussian traffic may 'have some Gaussian characteristics' as long as there is a large degree of aggregation, by virtue of central-limit type of arguments, which may explain that the fit is still reasonable.

This concludes our analysis of the possible errors in the indirect method to estimate burstiness. Extensive validation of the indirect method to estimate burstiness when used in the context of link dimensioning, is presented in Chapter 6.

5.5 Hints on implementation

The experiments with artificial traffic, as described in Sections 5.3 and 5.4, have been performed off-line, in that we have used scripts that 'parse' the synthetic traffic trace, mimicking the buffer content dynamics — more details on the mimicking are given in Section 6.3.1.

An interesting question is whether the approach that we proposed to indirectly determine the burstiness is feasible in run-time, in operational environments. From the proposed procedure, see Algorithm 5.4, we can derive the following functional requirements for an implementation of the inversion procedure:

1. a notion of the amount of data in a buffer;
2. a way to regularly poll this information;
3. software/hardware to determine the BCD, and then determine the resulting estimate of $\nu(\cdot)$.

To our best knowledge, these requirements do not lead to any fundamental or conceptual problems. The first requirement is already addressed: Random Early Detection (RED) queuing algorithms, which are widely implemented in modern routers, also keep track of the amount of queued data. In RED the buffer content (or, more precisely, a proxy of the buffer contents in the near past) is used to randomly discard packets (to which the TCP-users

react by reducing their window size), see [FJ93]. It is evident that information on the buffer occupancy can also be used for other purposes, such as the estimation of the BCD.

The second requirement may be fulfilled by, for instance, the use of SNMP (in case the entire procedure is not run on the router itself). We have, however, not found an (IETF-standardized) MIB that gives access to buffer occupancy data. The most close comes the `ifOutQLen` object, described as 'the length of the output packet queue (in packets)' in the Interfaces Group MIB. At time of writing, however, use of this object has been deprecated by the IETF. Also, the specification of the `ifOutQLen` object states that it returns the length in packets, whereas our methodology uses the queue length in bits (or bytes). One could possibly circumvent this 'incompatibility' by using deterministic or probabilistic computations to come from amount of packets to (expected) packet sizes. We recommend that implementers of our inversion approach seek standardization through the IETF or other appropriate forums, of a process to poll the buffer occupancy.

The last requirement has already been addressed — see our results in this and the next chapter. Hence, we conclude that there are no fundamental or conceptual problems preventing the actual application of our approach in practice.

> **Remark:** *Virtual queues* — Interestingly, there is the possibility of decoupling the inversion procedure from the actual queue in the router. More precisely: in software, one could keep track of a 'virtual queue' that is drained at a link rate C_q that might be different from the actual link rate C of the router. The idea of such decoupling is illustrated in Figure 5.9. Particularly when the 'real queue' is empty during a substantial fraction of the time, which inevitably results in poor estimates of the BCD, one could better use a virtual queue that is drained at a lower rate C_q.

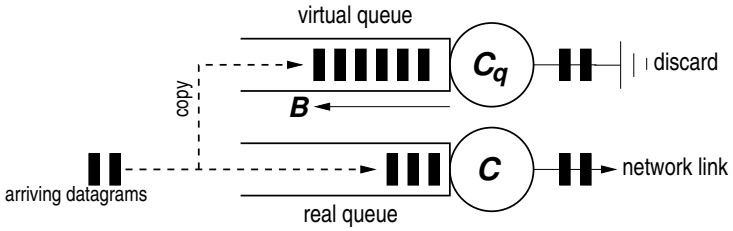


Figure 5.9: Decoupling the real queue from a virtual queue used in the inversion procedure

5.6 Concluding remarks

There are two fundamentally different methods to estimate burstiness in network traffic:

1. directly inferring the variance at small timescales by measuring the traffic rates at small timescales. This may not be attractive, as measuring at such small timescales is not feasible using SNMP in practice; and
2. indirectly estimating the variance through coarse-grained polling of the buffer occupancy, and then 'inverting' to the variance curve.

For the second approach, we have found that we can accurately estimate the variance of the traffic at any timescale, through the following 'inversion formula':

$$v(t) \approx \inf_{B>0} \frac{(B + (C_q - \mu)\, t)^2}{-2 \log \mathbb{P}(Q > B)} \ ,$$

in which the distribution of the queue length (or buffer occupancy), which is known to determine $\mathbb{P}(Q > B)$ for any $B > 0$, is found through coarse-grained polling of the buffer occupancy.

We have performed simulation experiments illustrating the inversion procedure, and assessing possible (numerical) errors in the indirect estimation of the variance curve. We conclude that our simulation experiments show the 'robustness' of the inversion procedure. Despite the approximations involved, with a relatively low measurement effort, the variance curve is estimated accurately, even for traffic that is 'not perfectly Gaussian'. Given the evident advantages of the inversion approach over the 'direct approach' (minimal measurement effort required, retrieval of the entire variance curve

$v(\cdot)$, etc.), the indirect method is to be preferred. In the next chapter we verify whether this conclusion also holds for real (i.e., not artificially generated) network traffic — we find that it does.

We have also given some hints on implementation of our methodology to indirectly estimate the burstiness of network traffic. An analysis of the requirements has shown that there are no fundamental impediments to implementation.

We finish this chapter with some notes on related methodologies:

Comparison with alternative measurement methodologies

The purpose of our inversion method is to retrieve the essential traffic characteristics at a low measurement cost. We remark that several other 'cheap' (i.e.,, with low measurement effort) methods have been proposed. We now briefly discuss some of these, and compare them with our approach.

The method described in Duffield *et al.* [DLO+95] aims to estimate the 'asymptotic cumulant function', i.e.,

$$\Lambda(\theta) := \lim_{t\to\infty} \frac{1}{t} \log \mathbb{E} e^{\theta A(t)},$$

from traffic measurements. This function is useful, because, under some assumptions on the traffic arrival process, it holds that $\log \mathbb{P}(Q > B) \approx -\theta^\star B$, for B large, where $\theta^\star$ solves $\Lambda(\theta) = C_q\theta$. The crucial assumption on the input traffic is that it should be short-range dependent — otherwise $\Lambda(\cdot)$ does not even exist (think of fBm, for which $\mathbb{E}\exp(\theta A(t))$ is of the form $\frac{1}{2}\theta^2 t^{2H}$). Of course, this requirement is quite restrictive. It is noted that the estimation of $\Lambda(\cdot)$ turns out to be far from straightforward (block-sizes need to be chosen, etc.). A crucial difference with our approach is that [DLO+95] measures *traffic*, whereas we propose to measure (or, better: to poll) the *buffer content*.

Another related study is by Kesidis *et al.* [CKR+95]. Like in our method, their approach relies on the estimation of the buffer content distribution $\log \mathbb{P}(Q > B)$. Under the assumption of short-range dependent input, $\log \mathbb{P}(Q > B)$ is linear for large B (with some slope $-\theta^\star$). Having estimated $\theta^\star$, the probability of overflow over higher buffer levels can in their view

be estimated, by extrapolating the function $\log \mathbb{P}(Q > B)$ linearly. Also this method does not deal with long-range dependent input. Comparing this 'extrapolation of a linear function' idea with Figure 5.1 on page 114, it becomes clear that such extrapolation may not be realistic: Figure 5.1 empirically shows that $\mathbb{P}(Q > B)$ does not decay exponentially in B (it rather decays at a slower rate: $\log(\mathbb{P}(Q > B) \sim \theta^\star B^{2-2H}$, with $2-2H < 1$ (because $H > 0.5$)).

6 Large-scale validation

In the previous chapters we developed rules and procedures to find the required bandwidth capacity level for a traffic stream and a prespecified performance criterion. In the present chapter we verify whether the estimated required bandwidth is indeed accurate. Our approach in this verification is to perform an extensive number of case studies that use the traces that we collected at five locations.

This chapter is organized as follows:

- *Section 6.1 introduces the framework in which we perform our validation study, and describes some implementation aspects.*
- *Section 6.2 presents large-scale validation results of our bandwidth provisioning formula assuming Gaussian traffic, i.e.,* $C = \mu + \frac{1}{T}\sqrt{\left(-2\log\varepsilon\right) v(T)}$*, using hundreds of packet traces.*
- *Section 6.3 provides an empirical validation of our methodology to indirectly estimate the burstiness* $v(T)$*, as required in the bandwidth provisioning formula, of real traffic through coarse-grained polling of the buffer occupancy, again using many traces of real traffic.*
- *Section 6.4 concludes.*

6.1 Introduction

The purpose of this chapter is to validate the theoretical results from the previous chapters, using hundreds of packet traces that we collected through network traffic measurements at various locations.

In Chapter 4, we developed rules to estimate the required bandwidth capacity, based on the stochastic characteristics of the offered traffic and a performance criterion. For Gaussian traffic, this required bandwidth formula is $C = \mu + \frac{1}{T}\sqrt{\left(-2\log\varepsilon\right)v(T)}$ — hence it involves some mean μ and burstiness $v(T)$. In Chapter 5, we then proposed an indirect way to estimate the burstiness of network traffic. In the present validation study, we evaluate both elements of network link dimensioning:

- We consider the accuracy of the estimation of the required bandwidth, i.e., formula (4.7) in Chapter 4, for real network traffic. This formula has as argument the burstiness, and this burstiness is estimated through the direct approach.
- We investigate the accuracy of our methodology to *indirectly* estimate burstiness as described in Chapter 5, for real network traffic. As noted earlier, this indirect approach to estimate the burstiness has as main advantage over the direct approach that it does not require measurements at small timescales.

The 'decoupling' in our assessment allows us to gain precise insight into, — and thus a good validation of — both elements of network link dimensioning. Hence, it allows to precisely determine where possible errors in the end-result (i.e., network link dimensioning) stem from. A description of the approach followed during both validation studies is given at the start of Section 6.2 and 6.3, respectively.

In the validation of the required bandwidth estimation, we choose to focus exclusively on the required bandwidth formula for Gaussian input traffic. This choice is motivated by the observation in Chapter 3 that it can be difficult (if at all possible) to estimate the modeling parameters for, say, an $M/G/\infty$ input model to accurately describe real traffic. As we will see in Section 6.2, for the validation of a required bandwidth formula, it is necessary to be able to estimate the modeling parameters for the traffic model that is assumed in that required bandwidth formula. Hence, we choose to focus on validating the required bandwidth formula for Gaussian traffic — we saw, in Chapter 3, that it is feasible to estimate the modeling parameters for a Gaussian traffic model from our traces.

6.2 Validation of the required bandwidth estimation

6.2.1 Approach

In Chapter 3, we found that real traffic can (in most cases) appropriately be described through a Gaussian traffic model. In Chapter 4, we then derived a formula that estimates the required bandwidth capacity C, for a given mean traffic load μ and burstiness $v(\cdot)$, and a prespecified performance criterion. We focus on achieving 'link transparency': in no more than a fraction ε of time intervals of length T, the offered traffic may exceed the link's capacity. Recall that this formula to estimate the required bandwidth capacity is as follows, cf. (4.7) on page 88:

$$C = \mu + \frac{1}{T}\sqrt{\left(-2\log\varepsilon\right) v(T)}.$$

Our approach in the present validation study of the above required bandwidth formula is as follows. Each packet trace is processed, using the `Net::Pcap` Perl module and rather straightforward computations in the Perl programming language. The `Net::Pcap` module gives access to each packet in a trace (and as such, the timestamp associated with the packet, and the packet size). Hence we can determine the average traffic rate μ, and compute the variance $v(\cdot)$ of the offered traffic at timescale T, as follows (where A_i denotes the amount of traffic offered over the ith interval of length T):

$$\mu = \frac{1}{nT}\sum_{i=1}^{n} A_i,$$
$$v(T) = \frac{1}{n-1}\sum_{i=1}^{n} (A_i - \mu)^2.$$

Then, it is a matter of inserting the resulting μ and $v(T)$, as well as the specified values of T and ε, into the above formula for C to get the (estimated) required bandwidth. As we argued in Chapter 5, it may not be feasible to find $v(T)$ in such 'direct' way in practice, because of the required measurements at small timescales. At this point, however, we want to only validate the required bandwidth formula (4.7), as motivated in Section 6.1.

We choose to determine the average traffic rate μ per 15 minutes (recall that each trace contains 15 minutes of traffic), and set T to 1 second, 500 milliseconds and 100 milliseconds (and thus determine the variance at those timescales), which are timescales that we believe are important to the perception of quality by (human) users. We set ε to 1%. After the case studies, we will give a few examples of validation using other parameter settings, in Section 6.2.7. It is noted that a network operator may choose its own settings of T and ε, depending on, e.g., the application(s) being used, Service Level Agreements, etc.

In order to validate if the estimated bandwidth capacity C indeed corresponds to the required bandwidth, we introduce the notion of 'realized exceedance', denoted with $\hat{\varepsilon}$. We define the realized exceedance as the fraction of cases in which the amount of offered traffic A_i exceeds the estimated required capacity C — we stress the fact that 'exceedance' in this context should not be confused with 'packet loss'. In other words:

$$\hat{\varepsilon} := \frac{\#\{A_i \mid A_i > CT\}}{n} \quad (i \in 1 \ldots n).$$

By definition, exceedance (as in $A(T) > CT$) may be expected in a fraction ε of all intervals. There are, however, (at least) two reasons why $\hat{\varepsilon}$ and ε may not be equal in practice. Firstly, the bandwidth provisioning formula assumed 'perfectly Gaussian' traffic, but real traffic may not be 'perfectly Gaussian', as we already observed in Chapter 3. Deviation of 'perfectly Gaussian' traffic (in other words: violation of the modeling assumption) may have an impact on the estimated C. Secondly, in the derivation of the generic bandwidth provisioning formula upon which the required bandwidth formula for Gaussian traffic is based, an approximation (Chernoff bound) is used (see (4.1)).

To assess to what extent the provisioning formula for Gaussian traffic is accurate for real traffic, it is clearly interesting to compare ε with $\hat{\varepsilon}$. Thus, we study the difference between ε and $\hat{\varepsilon}$ to get insight into the deviation between the outcome of the model-based formula and the real traffic.

The validation study as described above is performed using the hundreds of traces that we collected at five different locations. For more information on the measurement locations, see the descriptions from page 35 onward.

6.2.2 Case-study #1: location *U*

The first case-study deals with two example traces taken from the measurements at location *U*. In this case-study as well as in the other case-studies in the subsequent sections, we have chosen to select traces that are interesting in that they either show positive or even negative results, or provide us with useful insights. We have tried to select representative traces from the hundreds of traces in total. After these graphical examples, validation results are tabulated, comprising results for *all* traces of this location.

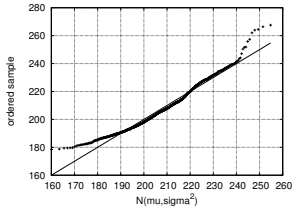
Figure 6.1 on the next page shows a first example trace from location *U*. The top row of graphs in Figure 6.1 shows the Q-Q plots (see Chapter 3 for more information on Q-Q plots) at timescales $T = 1$ second, 500 and 100 milliseconds, comparing the empirical distribution of the offered traffic per interval, to the Gaussian distribution with mean and variance equal to the corresponding empirical values. Given the 'goodness-of-fit' γ values that are close to the 'perfectly Gaussian' value of 1, one can say that the traffic in this trace is 'fairly Gaussian'.

One can also see from Figure 6.1 that the highest peaks in the traffic rates are 'unexpectedly high', if one would assume Gaussian traffic: these highest peaks correspond to the few points in the top-right corners of the Q-Q plots. The fact that these are above the 'diagonal' indicate that these peaks are higher than could be expected from a Gaussian distribution with such mean and variance.

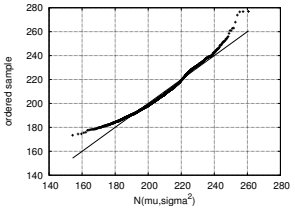
It can clearly be seen, in the second row of graphs, that although the average rate over the entire 900 seconds interval is some 207 Mbit/s, at smaller timescales the peak rates are significantly higher, i.e., up to 300 Mbit/s. The fluctuations also show from the variances: at a timescale of $T = 1$ second, $\sqrt{v(T)}$ is some 15.5 Mbit.

Next, we determine the estimated bandwidth capacity using the required bandwidth formula (4.7). As can be seen in the captions under the bottom row graphs in Figure 6.1, the required bandwidth capacities C are estimated at some 254.3, 257.0 and 266.4 Mbit/s, for $T = 1$ second, 500 and 100 milliseconds, respectively.

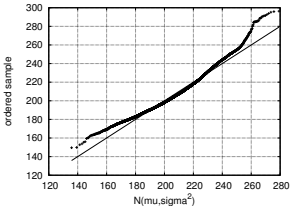
Subsequently, the realized exceedance figures are determined, and these are also shown in the captions. The realized exceedance is clearly within the



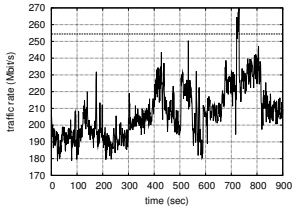
(a) $T = 1$ sec, $\gamma = 0.983$



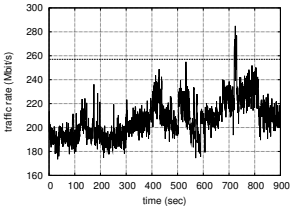
(b) $T = 500$ msec, $\gamma = 0.985$



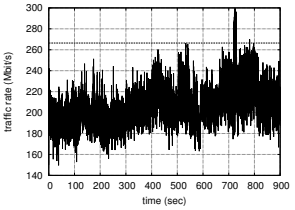
(c) $T = 100$ msec, $\gamma = 0.987$



(d) $T = 1$ sec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 15.5$ Mbit, $C = 254.3$ Mbit/sec, $\hat{\varepsilon}$: 0.0089

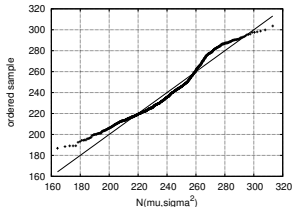


(e) $T = 500$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 8.2$ Mbit, $C = 257.0$ Mbit/sec, $\hat{\varepsilon}$: 0.0061

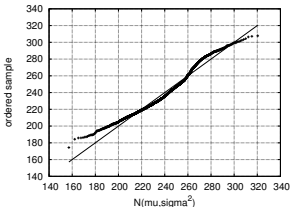


(f) $T = 100$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 1.9$ Mbit, $C = 266.4$ Mbit/sec, $\hat{\varepsilon}$: 0.0059

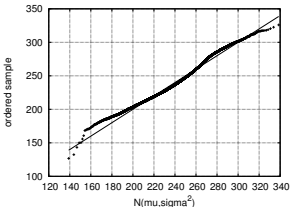
Figure 6.1: Case-study #1 (location U), example trace #1 ($\mu = 207$ Mbit/sec)



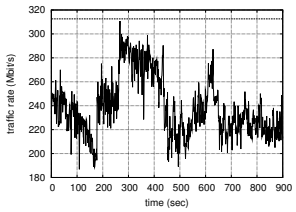
(a) $T = 1$ sec, $\gamma = 0.980$



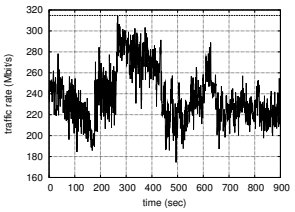
(b) $T = 500$ msec, $\gamma = 0.984$



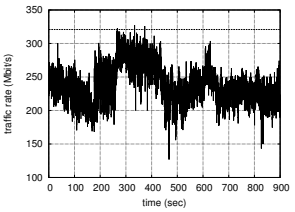
(c) $T = 100$ msec, $\gamma = 0.992$



(d) $T = 1$ sec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 24.3$ Mbit, $C = 312.6$ Mbit/sec, $\hat{\varepsilon}$: 0.0000



(e) $T = 500$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 12.6$ Mbit, $C = 314.8$ Mbit/sec, $\hat{\varepsilon}$: 0.0000



(f) $T = 100$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 2.7$ Mbit, $C = 320.8$ Mbit/sec, $\hat{\varepsilon}$: 0.0003

Figure 6.2: Case-study #1 (location U), example trace #2 ($\mu = 239$ Mbit/sec)

limit of $\varepsilon = 0.01$. Hence, the required bandwidth capacity is correctly estimated: the desired performance target is met.

It is noted that in this example there is only a relatively small difference between the estimated required bandwidth capacity, and the peak traffic rates at the timescales we assessed. Thus, one could argue, one could also measure the peak traffic rates at such timescales and dimension for the resulting averages. As we will see in the case-studies for some other locations, however, this claim does not always apply: the differences between the estimated required bandwidth capacity, allowing for e.g., 1% exceedance, and the maximum traffic rates can be large, especially at small timescales (up to hundreds of percents). It is also noted that the (relative) differences between the estimated bandwidth capacities at different timescales (which are rather small in this example), are not typical for all traces nor for all locations. The other examples in these case-studies clearly show that larger differences occur as well.

Figure 6.2 shows another interesting example trace from measurement location *U*. This is one of the few traces we have found that have actually lower-than-expected peak traffic rates, which is indicated by right-top data points in the Q-Q plots that are this time below the diagonal. Looking at the traffic rates graph in the bottom row of Figure 6.2, these lower-than-expected peak rates might be explained[1] by the observation that there is a significant interval, from the approximately 280th to 420th second in the trace, over which the average rate is higher than in the other parts of the trace. Such a long interval obviously increases the average rate over the entire measurement interval of 900 seconds, and actually leads to over-estimations of the (expected) rates outside the 280-420 second interval. This can also be seen from the Q-Q plot, where quite a few data points in the middle are below the diagonal. The mentioned over-estimations ultimately lead to an estimated required bandwidth level, which is somewhat higher than needed, as is reflected by the near-zero realized exceedance fractions.

After having discussed two example traces for measurement location *U*, we now turn to the error statistics of the estimation of the required bandwidth capacity for *all* traces at location *U*. Therefore we list, in Table 6.1, the aver-

[1] Another reason could be that the measured link was actually fully loaded at some time during this measurement interval. Our measurement setup does, however, not allow us to further investigate this in much detail.

T	avg $\lvert\varepsilon - \hat{\varepsilon}\rvert$	stderr $\lvert\varepsilon - \hat{\varepsilon}\rvert$
1 sec	0.0095	0.0067
500 msec	0.0089	0.0067
100 msec	0.0077	0.0047

Table 6.1: Required bandwidth estimation errors (location U) ($\varepsilon = 0.01$)

T	avg dimensioning factor	stderr dimensioning factor
1 sec	1.33	0.10
500 msec	1.35	0.09
100 msec	1.42	0.09

Table 6.2: Required bandwidth versus mean load (location U) ($\varepsilon = 0.01$)

age difference between the realized exceedance fractions $\hat{\varepsilon}$ and the specified ε. Table 6.1 shows that the required bandwidth formula results in an estimation that is on average less than 1%-point off the projected error. We also list for completeness the standard deviation (i.e., 'stderr') of the errors over all traces.

For bandwidth provisioning 'rules of thumb'-purposes, it is interesting to get an idea of the required 'dimensioning factor', i.e., the (estimated) required bandwidth capacity compared to the average load (in other words: C/μ). The dimensioning factor, averaged over all traces at location U, is shown in Table 6.2. It shows that in the case of location U, some 30-40% extra bandwidth capacity is required on top of the 'long-term' average traffic rate, to cater for peaks on smaller timescales. Later on in this section, we will see that these values can be much larger in other networking environments. There are various reasons that these overdimensioning figures change from location to location, e.g., it depends on the utilization of the network ('is there room for large traffic rate peaks?'), and the access link speeds ('what is the impact that a single user can have on the aggregated network traffic?').

6.2.3 Case-study #2: location *R*

Figure 6.3 on page 136 shows a first example trace taken from location R. The trace is 'almost perfectly Gaussian' at the all timescales we assessed, according to the goodness-of-fit γ of over 0.99 at all three investigated timescales.

T	avg $\lvert\varepsilon - \hat{\varepsilon}\rvert$	stderr $\lvert\varepsilon - \hat{\varepsilon}\rvert$
1 sec	0.0062	0.0060
500 msec	0.0063	0.0064
100 msec	0.0050	0.0053

Table 6.3: Required bandwidth estimation errors (location R) ($\varepsilon = 0.01$)

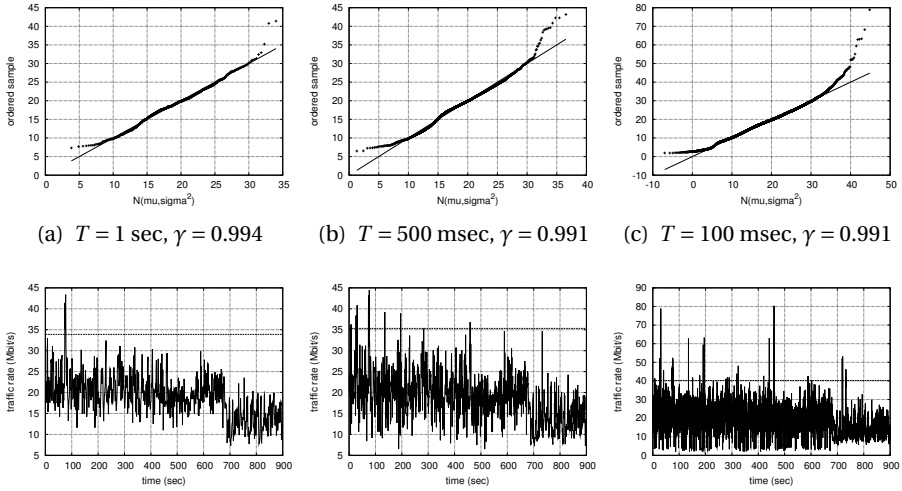
T	avg dimensioning factor	stderr dimensioning factor
1 sec	2.91	1.51
500 msec	3.12	1.57
100 msec	3.82	1.84

Table 6.4: Required bandwidth versus mean load (location R) ($\varepsilon = 0.01$)

The bottom row graphs in Figure 6.3 show the actual traffic rates at these timescales, as well as the estimated required bandwidth capacity to meet the performance criterion with $\varepsilon = 0.01$ at the different Ts. Given the realized exceedance values that are well below the 'limit' 0.01, the performance criteria are easily met.

Figure 6.4 shows a second example trace from location R. In contrast with Figure 6.3, this second trace is not at all Gaussian: $\gamma < 0.9$. Still, however, the performance criteria are met at all three timescales, as is indicated by the realized exceedance values in the bottom row captions.

As for the case-study for location U, we list the required bandwidth estimation errors for all traces at location R in Table 6.3. The errors are, again, small, which means that the required bandwidth is accurately estimated given the performance criterion that should be met. The dimensioning factors are also listed, in Table 6.4. Clearly, there is (relatively) more extra bandwidth capacity required to cater for fluctuations on small timescales, than was the case at location U. This is due to the low utilization of the network, and the relative high access link speeds: a single user with a 100 Mbit/s access link can have significant impact on the 1 Gbit/s 'uplink'. This causes the peak traffic rates in the network to be relatively high (i.e., high variance or burstiness); to cater for these peaks, the network should be overdimensioned by, in this case, some factor 3 to 4.



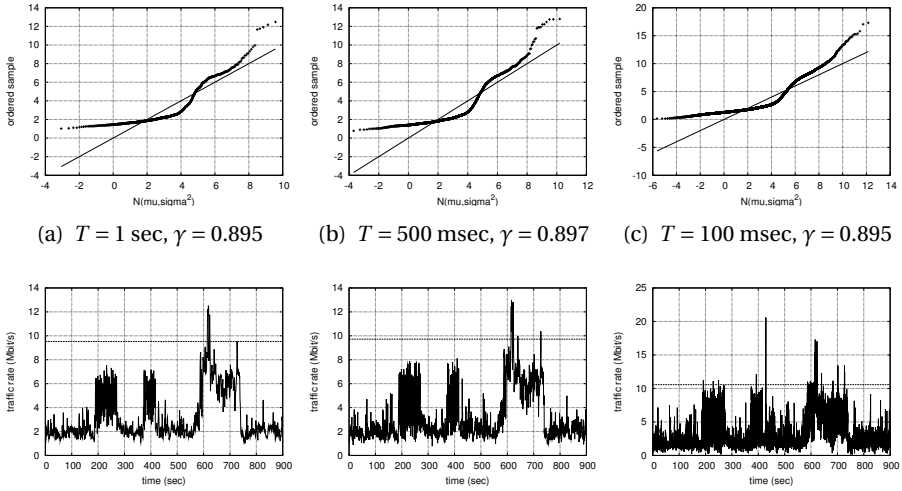
(a) $T = 1$ sec, $\gamma = 0.994$ (b) $T = 500$ msec, $\gamma = 0.991$ (c) $T = 100$ msec, $\gamma = 0.991$

(d) $T = 1$ sec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 4.9$ Mbit, $C = 33.9$ Mbit/sec, $\hat{\varepsilon}$: 0.0044

(e) $T = 500$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 2.7$ Mbit, $C = 35.3$ Mbit/sec, $\hat{\varepsilon}$: 0.0078

(f) $T = 100$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 0.70$ Mbit, $C = 40.2$ Mbit/sec, $\hat{\varepsilon}$: 0.0067

Figure 6.3: Case-study #2 (location R), example trace #1 ($\mu = 18.9$ Mbit/sec)



(a) $T = 1$ sec, $\gamma = 0.895$ (b) $T = 500$ msec, $\gamma = 0.897$ (c) $T = 100$ msec, $\gamma = 0.895$

(d) $T = 1$ sec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 2.1$ Mbit, $C = 9.5$ Mbit/sec, $\hat{\varepsilon}$: 0.0100

(e) $T = 500$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 1.0$ Mbit, $C = 9.7$ Mbit/sec, $\hat{\varepsilon}$: 0.0094

(f) $T = 100$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 0.24$ Mbit, $C = 10.1$ Mbit/sec, $\hat{\varepsilon}$: 0.0094

Figure 6.4: Case-study #2 (location R), example trace #2 ($\mu = 3.3$ Mbit/sec)

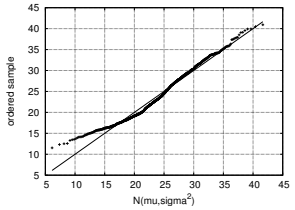
6.2.4 Case-study #3: location *C*

The third case-study deals with traces from measurement location *C*. The first example trace, shown in Figure 6.5 on the next page, nicely fits the Gaussian model ($\gamma \approx 0.99$). As could be expected by now, the required bandwidth estimation is rather accurate as indicated by the realized exceedance fractions.

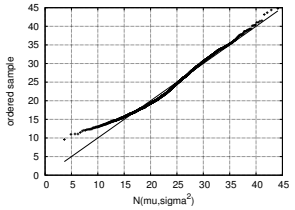
The second example trace at location *C*, shown in Figure 6.6 is particularly interesting. There is clearly more network usage in the last part of the measurement interval compared with the first part. This fuels the idea of this trace being an example of observing non-stationary traffic: the characteristics of the offered traffic change too much within the trace, and therefore they cannot be accurately modeled. The shape of the Q-Q plot curve does reflect this: it has a rather irregular shape with both significant under- as well as over-estimations of the expected traffic rates. It can also be seen from the realized exceedance fractions, which are too large: in more than a fraction ε of intervals, more traffic is offered than can be handled, should the link be dimensioned at the proposed estimated required bandwidth capacity. Consequently, the performance criterion would not be met in this example.

A bit of speculation on the reason of the presumed non-stationarity of this particular trace (which is *not* representative for the traces at this location, but is included as an example of a case in which the theory fails): measurement location *C* is the uplink of a college's network. This particular measurement interval started just after noon, at 12:05. Speculatively, it could be the case that lessons stop around 12:10 or 12:15, and that students then rush to computers to log on to the network, browse the web, read mail, download files, etc. This would explain the rather drastic increase in traffic rates in the course of the measurement interval. The described example also shows the 'limitations' of an approach like ours in link dimensioning: no formula would 'predict' such significant traffic growth.

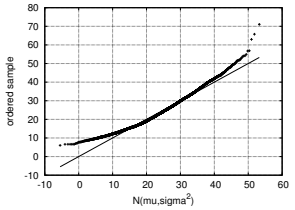
As for the other locations we have estimated the required bandwidth for all traces at location *C*, and compared the realized exceedance fraction numbers with the target $\varepsilon = 0.01$. The results of this comparison are listed in Table 6.5 on page 139. The table shows that the estimations are rather accurate, again.



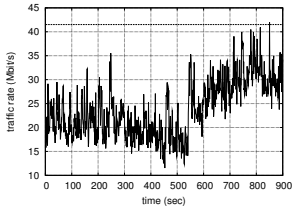
(a) $T = 1$ sec, $\gamma = 0.987$



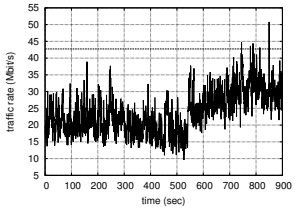
(b) $T = 500$ msec, $\gamma = 0.991$



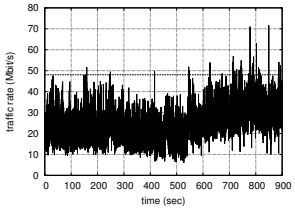
(c) $T = 100$ msec, $\gamma = 0.988$



(d) $T = 1$ sec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 5.8$ Mbit,
$C = 41.5$ Mbit/sec,
$\hat{\varepsilon}$: 0.0011

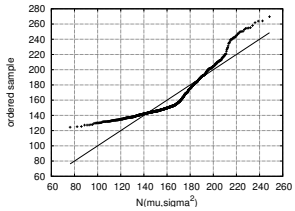


(e) $T = 500$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 3.1$ Mbit,
$C = 42.7$ Mbit/sec,
$\hat{\varepsilon}$: 0.0028

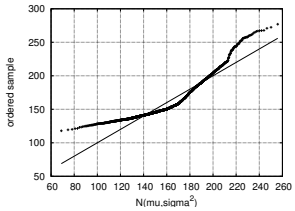


(f) $T = 100$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 0.79$ Mbit,
$C = 48.0$ Mbit/sec,
$\hat{\varepsilon}$: 0.0044

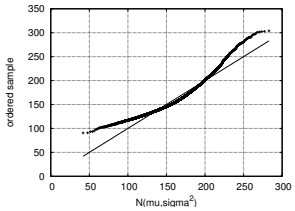
Figure 6.5: Case-study #3 (location *C*), example trace #1 ($\mu = 23.4$ Mbit/sec)



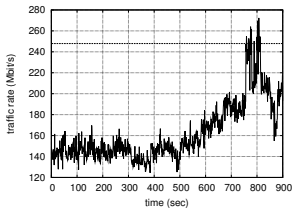
(a) $T = 1$ sec, $\gamma = 0.924$



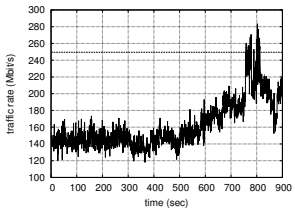
(b) $T = 500$ msec, $\gamma = 0.934$



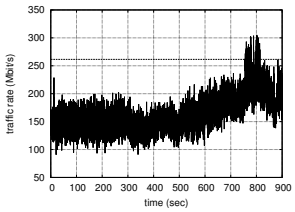
(c) $T = 100$ msec, $\gamma = 0.967$



(d) $T = 1$ sec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 28.1$ Mbit,
$C = 247.9$ Mbit/sec,
$\hat{\varepsilon}$: 0.0178



(e) $T = 500$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 14.5$ Mbit,
$C = 249.5$ Mbit/sec,
$\hat{\varepsilon}$: 0.0178



(f) $T = 100$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 3.3$ Mbit,
$C = 261.4$ Mbit/sec,
$\hat{\varepsilon}$: 0.0126

Figure 6.6: Case-study #3 (location *C*), example trace #2 ($\mu = 162$ Mbit/sec)

T	avg $\lvert\varepsilon - \hat{\varepsilon}\rvert$	stderr $\lvert\varepsilon - \hat{\varepsilon}\rvert$
1 sec	0.0069	0.0047
500 msec	0.0066	0.0043
100 msec	0.0055	0.0041

Table 6.5: Required bandwidth estimation errors (location *C*) ($\varepsilon = 0.01$)

T	avg dimensioning factor	stderr dimensioning factor
1 sec	1.71	0.44
500 msec	1.83	0.49
100 msec	2.13	0.67

Table 6.6: Required bandwidth versus mean load (location *C*) ($\varepsilon = 0.01$)

In Table 6.6 the required overdimensioning figures are shown for the traces from location *C*. To cater for peaks on small timescales, some 2 times the amount of the 15 minute average traffic rate is required.

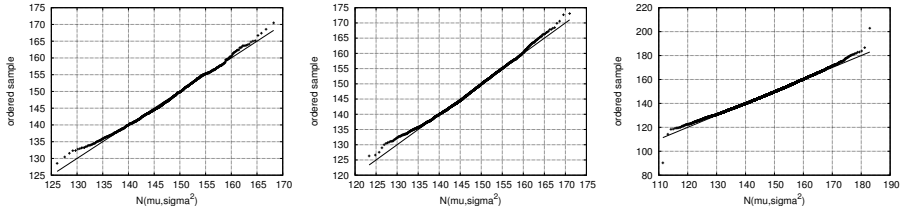
6.2.5 Case-study #4: location *A*

Case-study #4 deals with the ADSL access network, i.e., measurement location *A*.

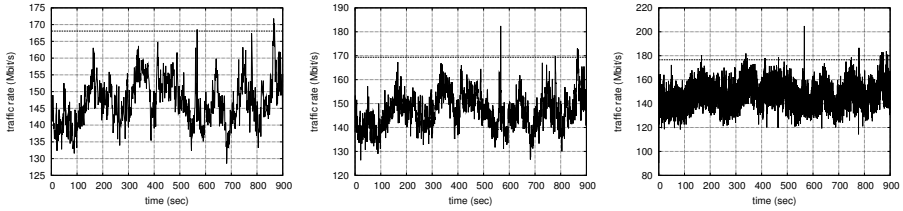
First, two example traces are presented, see Figures 6.7 and 6.8 on the next page. Both traces are 'almost perfectly Gaussian', which is reflected by the goodness-of-fit γ values that are very close to 1. A reason for this near-perfect Gaussian traffic likely is, e.g., that the access links are relatively small (up to 1 Mbit/s) compared to the 'backbone' (up)link of 1 Gbit/s. There is large aggregation in terms of users: the traffic of hundreds (if not more) users is aggregated (at our measurement point). Central-limit-theorem type of arguments then dictate that the resulting traffic stream tends to Gaussian.

Another observation, related to the relatively small access links, is that the fluctuations in the aggregated traffic rate are small. For instance, as can be seen from the caption in Figure 6.7 for example trace #1, the square-root of the variance at timescale 1 second is just 6.9 Mbit, which is low compared to the average traffic rate of 147 Mbit/s.

Interestingly, while the traffic in the second example trace is 'almost perfectly Gaussian', the exceedance is significantly below the projected level of



(a) $T = 1$ sec, $\gamma = 0.997$ (b) $T = 500$ msec, $\gamma = 0.998$ (c) $T = 100$ msec, $\gamma = 0.998$

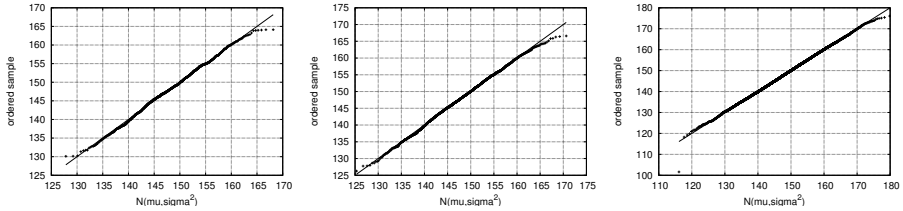


(d) $T = 1$ sec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 6.9$ Mbit,
$C = 168.1$ Mbit/sec,
$\hat{\varepsilon}$: 0.0033

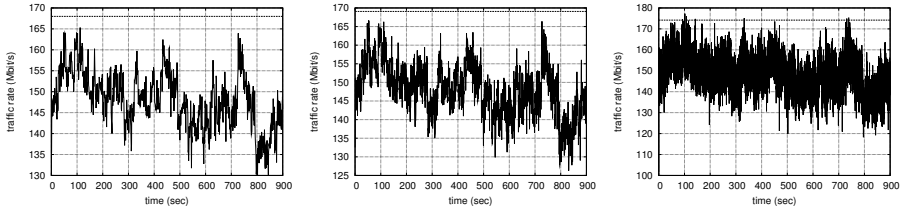
(e) $T = 500$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 3.6$ Mbit,
$C = 169.4$ Mbit/sec,
$\hat{\varepsilon}$: 0.0028

(f) $T = 100$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 0.97$ Mbit,
$C = 176.6$ Mbit/sec,
$\hat{\varepsilon}$: 0.0027

Figure 6.7: Case-study #4 (location A), example trace #1 ($\mu = 147$ Mbit/sec)



(a) $T = 1$ sec, $\gamma = 0.999$ (b) $T = 500$ msec, $\gamma = 0.999$ (c) $T = 100$ msec, $\gamma = 1.000$



(d) $T = 1$ sec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 6.6$ Mbit,
$C = 168.0$ Mbit/sec,
$\hat{\varepsilon}$: 0.0000

(e) $T = 500$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 3.5$ Mbit,
$C = 169.1$ Mbit/sec,
$\hat{\varepsilon}$: 0.0000

(f) $T = 100$ msec, $\varepsilon = 0.01$,
$\sqrt{v(T)} = 0.86$ Mbit,
$C = 174.2$ Mbit/sec,
$\hat{\varepsilon}$: 0.0008

Figure 6.8: Case-study #4 (location A), example trace #2 ($\mu = 148$ Mbit/sec)

T	avg $\lvert\varepsilon - \hat{\varepsilon}\rvert$	stderr $\lvert\varepsilon - \hat{\varepsilon}\rvert$
1 sec	0.0083	0.0027
500 msec	0.0083	0.0024
100 msec	0.0079	0.0020

Table 6.7: Required bandwidth estimation errors (location A) ($\varepsilon = 0.01$)

T	avg dimensioning factor	stderr dimensioning factor
1 sec	1.13	0.03
500 msec	1.14	0.03
100 msec	1.19	0.03

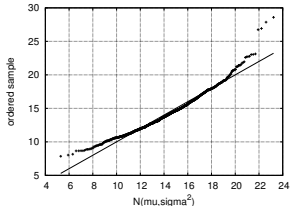
Table 6.8: Required bandwidth versus mean load (location A) ($\varepsilon = 0.01$)

0.01. The reason for this can be seen from the top-right corners of the Q-Q plots in Figure 6.8: the highest peak rates are quite a bit below the 'expected peak rates', given a Gaussian distribution of the offered traffic. It can be seen that the actual peaks are lower than the traffic model expects, and hence, the required bandwidth formula (which is based on that traffic model) indeed over-estimates the required capacity to cater the offered traffic.

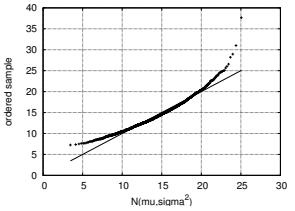
A traffic model that would put more emphasis on the (upper)tail of the actual traffic, with a suitable required bandwidth formula, could help here to more accurately predict the bandwidth needed to handle the offered traffic. A similar argument goes for cases where the (upper)tail of the distribution of the actual traffic is unexpectedly high according to the presently used Gaussian model — this could now lead to under-estimation of the required bandwidth, whereas a model that puts more emphasis on the tail possibly would not.

As for the other locations we have estimated the required bandwidth for all traces at location A, and compared the resulting realized exceedance fractions with the target $\varepsilon = 0.01$. The results of this comparison are listed in Table 6.7. The table shows that the estimations are rather accurate, again.

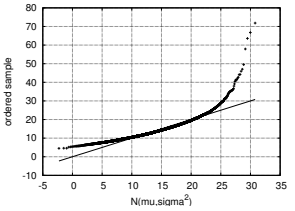
The dimensioning factors for location A in Table 6.8 are low: close to 1, and very stable over all traces as indicated by the (rounded) zero standard deviation. This all is most likely caused by the relatively small access links in this ADSL network.



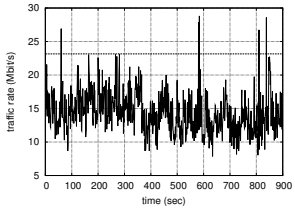
(a) $T = 1$ sec, $\gamma = 0.982$



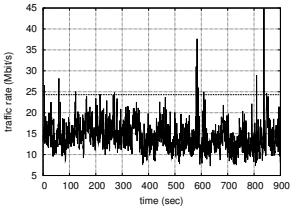
(b) $T = 500$ msec, $\gamma = 0.980$



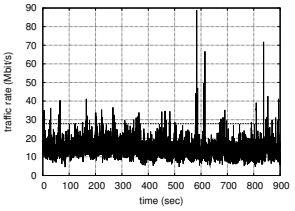
(c) $T = 100$ msec, $\gamma = 0.949$



(d) $T = 1$ sec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 2.9$ Mbit, $C = 23.2$ Mbit/sec, $\hat{\varepsilon}$: 0.0056

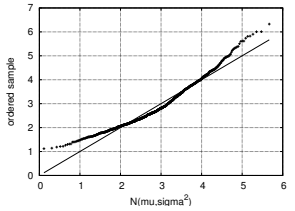


(e) $T = 500$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 1.6$ Mbit, $C = 24.3$ Mbit/sec, $\hat{\varepsilon}$: 0.0083

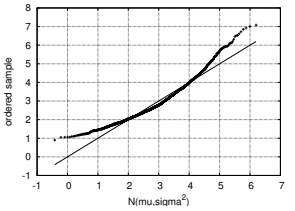


(f) $T = 100$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 0.45$ Mbit, $C = 27.8$ Mbit/sec, $\hat{\varepsilon}$: 0.0100

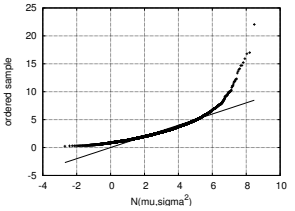
Figure 6.9: Case-study #5 (location *S*), example trace #1 ($\mu = 14.3$ Mbit/sec)



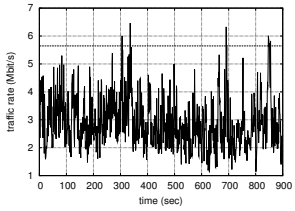
(a) $T = 1$ sec, $\gamma = 0.980$



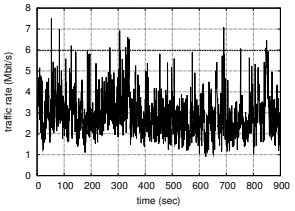
(b) $T = 500$ msec, $\gamma = 0.972$



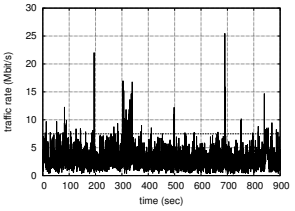
(c) $T = 100$ msec, $\gamma = 0.943$



(d) $T = 1$ sec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 0.91$ Mbit, $C = 5.6$ Mbit/sec, $\hat{\varepsilon}$: 0.0089



(e) $T = 500$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 0.51$ Mbit, $C = 6.0$ Mbit/sec, $\hat{\varepsilon}$: 0.0094



(f) $T = 100$ msec, $\varepsilon = 0.01$, $\sqrt{v(T)} = 0.15$ Mbit, $C = 7.5$ Mbit/sec, $\hat{\varepsilon}$: 0.0112

Figure 6.10: Case-study #5 (location *S*), example trace #2 ($\mu = 2.9$ Mbit/sec)

6.2.6 Case-study #5: location *S*

Figure 6.9 on the preceding page shows a first example trace from our last measurement location. The traffic is 'fairly Gaussian', as can again be seen from the γ values, and the required bandwidth capacity to meet the performance criteria are well estimated.

Interestingly, there is quite a difference between the estimated required capacity and the maximum achieved traffic rate, especially at timescale $T = 100$ milliseconds. We saw in other traces, e.g., case study #1 (location *U*), that the estimated required bandwidth capacity was only just below the maximum achieved traffic rate (at the assessed timescales) for that trace. In the present case, however, looking at the $T = 100$ milliseconds timescale (bottom-right graph in Figure 6.9), there is a factor 3 difference between the estimated required bandwidth (which still meets the desired performance criterion, as can be seen from the realized exceedance fraction of 0.01, equal to ε), and the highest traffic rate in the trace. A network manager who is interested in providing 'good performance' with as little bandwidth as possible, would thus clearly benefit from using our approach.

Figure 6.10 shows a second example trace from measurement location *S*. The difference with the trace presented in Figure 6.9 is that the average traffic rate is rather low (this second example trace was taken during the night, which likely explains the low utilization). Still it can be seen that the traffic is, again, 'fairly Gaussian'. Only at our smallest timescale of $T = 100$ milliseconds, the required bandwidth capacity is slightly underestimated. As the Q-Q plot shows, at the 100 milliseconds timescale there is a quite significant upper tail. Hence, the peak traffic rates are higher than 'expected' under a normal distribution, which in turn leads to the slight underestimation of the required bandwidth.

Finally, Table 6.9 and Table 6.10 show the average required bandwidth estimation errors and overdimensioning figures for all traces from measurement location *S*. It is clear that the bandwidth estimation formula gives accurate results: it yields on average only a small deviation from the projected ε. The average required overdimensioning to cater for peaks on small timescales, compared with the 15 minute average traffic rates, is some factor 2-2.5 in the case of location *S*.

T	avg $\lvert\varepsilon - \hat{\varepsilon}\rvert$	stderr $\lvert\varepsilon - \hat{\varepsilon}\rvert$
1 sec	0.0052	0.0050
500 msec	0.0049	0.0055
100 msec	0.0040	0.0059

Table 6.9: Required bandwidth estimation errors (location S) ($\varepsilon = 0.01$)

T	avg dimensioning factor	stderr dimensioning factor
1 sec	1.98	0.78
500 msec	2.10	0.87
100 msec	2.44	1.01

Table 6.10: Required bandwidth versus mean load (location S) ($\varepsilon = 0.01$)

This almost concludes our validation of the required bandwidth formula for Gaussian traffic. We finish with a few examples using different parameter settings for T and ε, to study the impact of the settings of these parameters on the required bandwidth.

6.2.7 Case-studies with other performance criterion settings

The case-studies presented in Sections 6.2.2 to 6.2.6 focused on validating the required bandwidth formula (4.7) for timescales $T = 1$ second, 500 milliseconds and 100 milliseconds, and $\varepsilon = 0.01$. Network operators, however, may want to use other settings of the parameters T and ε, depending on the applications, business aspects, etc.

In this section, we give a few examples of using other settings for the performance criterion parameters T and ε, namely $T = 10, 50, 100$ and 500 msec, and ε ranging from 0.00001 to 0.1. We use the first example trace of each location as discussed in the previous sections. For each of these examples, we compute the required bandwidth capacity C according to (4.7), as a function of T and ε; the burstiness levels $v(T)$ at each timescale T are computed (directly) from the traffic trace, and so is the average throughput μ (per 15 minutes, i.e., the length of the trace). The results are presented in Figure 6.11 on page 146, where each curve corresponds to a specific setting of T.

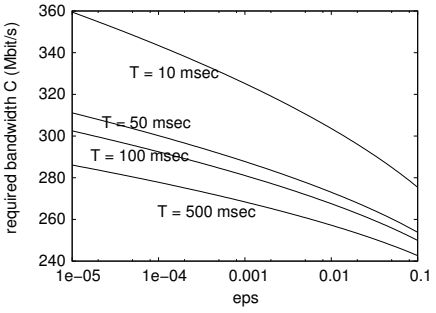
It can clearly be seen from Figure 6.11 that the required bandwidth C decreases in both T and ε, which is intuitively clear. The figures show that C

is more sensitive to T than to ε — take for instance Figure 6.11.(a), i.e., location U, example trace #1; at $\varepsilon = 10^{-5}$, the difference in required bandwidth between $T = 10$ msec and $T = 100$ msec, is some 20%. At $T = 100$ msec, the difference in required bandwidth between $\varepsilon = 10^{-5}$ and $\varepsilon = 10^{-4}$ is just 3% approximately. For other examples, the precise differences may change but the impression stays the same: a tenfold increase in stringency with respect to T requires (relatively) more extra bandwidth, than a tenfold increase in stringency with respect to ε (of course, this could already be expected given the required bandwidth formula).

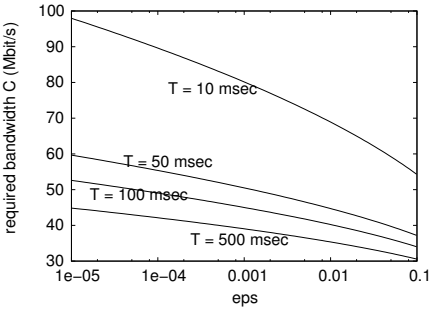
We have verified whether the required bandwidth is accurately estimated for these case-studies with different settings of T and ε. The estimation errors in these new situations are similar to the earlier obtained results (cf., e.g., Table 6.9 for location S). It should be noted however, that we have not been able to verify this for all possible combinations of T and ε: for $\varepsilon = 10^{-5}$ and $T = 500$ msec for instance, there are only 1800 samples in our traffic trace (which has a length of 15 minutes) and hence, we cannot compute the accuracy of our estimation. Another remark that should be made here, is that for locations with only limited aggregation in terms of users (say some tens concurrent users), combined with a small timescale of $T = 10$ msec, the traffic is no longer Gaussian (i.e., $\gamma \ll 1$). As a consequence, the accuracy of our required bandwidth estimation decreases.

The dimensioning factors (cf., e.g., Table 6.10 for location S) for the present case-studies can be obtained through division of C at certain T and ε, and μ. As indicated above, the dimensioning factor increases when the performance criterion (through ε and T) becomes more stringent (as can be expected, given the required bandwidth formula (4.7)).

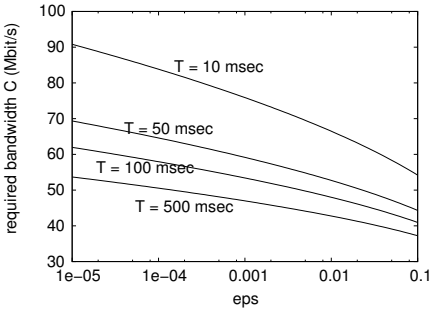
This concludes our validation of the required bandwidth formula. In the next section, the focus is on validating our indirect approach to estimate the burstiness of network traffic.



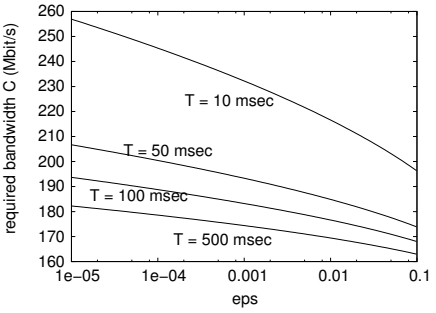
(a) Loc. U ex. #1, $\mu \approx 207$ Mbit/s



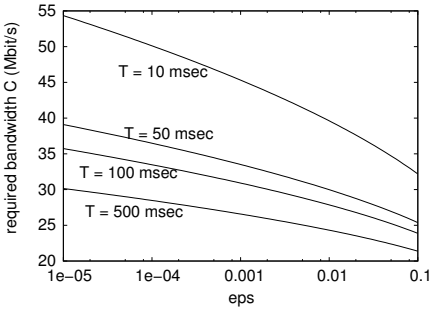
(b) Loc. R ex. #1, $\mu \approx 19$ Mbit/s



(c) Loc. C ex. #1, $\mu \approx 24$ Mbit/s



(d) Loc. A ex. #1, $\mu \approx 147$ Mbit/s



(e) Loc. S ex. #1, $\mu \approx 14$ Mbit/s

Figure 6.11: Required bandwidth for other settings of T and ε

6.3 Validation of the indirect burstiness estimation

In the previous section, we have validated the required bandwidth estimation formula for Gaussian traffic (4.7). In that validation, we directly estimated the burstiness (variance) of the real traffic. The validation using hundreds of traces of real traffic showed that the required bandwidth is accurately estimated by this formula.

The required bandwidth formula (4.7) requires the variance $v(T)$ to be known on timescale T. As we argued in Chapter 5, given that T may be rather small, it may be difficult to directly determine $v(T)$ in real scenarios. In Chapter 5, we proposed a methodology in which $v(T)$ was indirectly estimated through infrequent polling of the occupancy of a buffer in front of the network link. This methodology does not require measurements on small timescale T to actually estimate $v(T)$.

The purpose of this section is to validate our methodology using our traces of real traffic: is $v(T)$, indeed, properly estimated through (5.9)? We first outline the validation approach (Section 6.3.1), and then perform an extensive number of case-studies using our traces (Section 6.3.2).

6.3.1 Approach

In Chapter 5 we described a methodology to estimate the variance $v(\cdot)$ at timescale T using the empirical distribution function of the buffer contents. The approximation for $v(\cdot)$ is as follows, cf. (5.9):

$$v(t) \approx \inf_{B>0} \Phi(B),$$

with

$$\Phi(B) := \frac{(B + (C_q - \mu)\,t)^2}{-2\log \mathbb{P}(Q > B)}.$$

In order to validate whether the above approximation is accurate, when using real traffic[2], we have implemented a simulation environment: the traces of real traffic are 'replayed' by inserting them into a virtual queue with ca-

[2] In Chapter 5 we used synthetic traffic, fBm, to validate whether the methodology is correct.

pacity C_q which is emulated in a `Perl` script. The approach to validate our methodology is described in the following pages.

First, like in the case-studies of Section 6.2, determine the average traffic rate μ from the traffic offered to the virtual queue. Similarly, compute the (sample) variance directly from the offered traffic stream, at some selected timescale T. In the sequel of this section, we denote this directly estimated variance with $v_{\mathsf{direct}}(\cdot)$.

We recall that such direct estimation of $v_{\mathsf{direct}}(T)$ may be difficult in real world environments, as argued in Chapters 1, 4 and 5 of this thesis, as T is likely chosen to be rather small.

Second, the variance is also indirectly estimated through the formula for the approximation of the variance as indicated above, which is in the sequel denoted with $v_{\mathsf{indirect}}(\cdot)$.

The approximation of $v_{\mathsf{indirect}}(\cdot)$ consists of two steps:

1. estimation of the empirical distribution function of the buffer occupancy $\mathbb{P}(Q > B)$, denoted with BCD, i.e., the probability that there are more than B bytes in the buffer (at a random point in time);
2. an optimization of $\Phi(B)$ over B, i.e., the infimum as indicated in the approximation.

Step 2 is straightforward. Step 1, however, involves setting various 'under the hood' parameters, which we detail next.

Recall from Chapter 5 that the BCD is estimated as follows (in line with Algorithm 5.5 on page 112, and the illustration of this procedure in Figure 5.9 — although there is no decoupling of real and virtual queue (like the decoupling shown in Figure 5.9) in our simulation, as we simulate only a virtual queue):

1. The trace with real traffic is processed packet by packet, say $p_1, \ldots, p_n$ for n the total number of packets in the trace. In the trace, each packet has an associated *len* and *ts* attribute, which are the original packet's size (including payload) in bytes and the timestamp at which the packet was measured, respectively. As for notation, $p_{j,\mathsf{len}}$ is the size of the jth packet, and $p_{j,\mathsf{ts}}$ its timestamp. In the same order as the packets in the trace, they are fed one-by-one into the virtual queue.

2. After each packet, the occupancy of the virtual queue is computed. With Q_j we denote the occupancy of the virtual queue (in bytes) after the jth packet has been fed to the virtual queue. Hence,

$$Q_j := p_{j,\mathsf{len}} + \max\left\{Q_{j-1} - C_q \cdot \left(p_{j,\mathsf{ts}} - p_{j-1,\mathsf{ts}}\right), 0\right\},$$

   which says that the occupancy after the ith packet is equal to that packet's size plus any data that was still left in the virtual queue, if any. Note that the virtual queue is drained through time at rate C_q, reflecting a virtual link with bandwidth capacity C_q that serves this virtual queue.

3. An observation q_i of the buffer occupancy is scheduled every τ seconds. For ease, we choose to perform this observation after the first packet that arrives at the virtual queue when τ seconds are passed since the last scheduled observation. As there are, depending on the trace, hundreds to thousands of packets fed into the virtual queue per second (real time, not simulation time), this only leads to a tiny deviation of the actual observation times from the scheduled observation times. Moreover, we have already argued in Chapter 5 that it is not important to perform this observation at very specific points in time. As we use traces with 900 seconds of real traffic, we get $900/\tau$ snapshots q_i of the occupancy. These are used to determine the empirical distribution function of the virtual queue's occupancy (BCD), as in Algorithm 5.4 on page 111.

In the case-studies have chosen to set τ to 1 second, which ensures that we have a sufficient number of snapshots to reliably estimate the BCD (see Section 5.4). Furthermore, we set T, the timescale that we aim to determine the variance for, to 100 milliseconds. As we argued before, such a timescale is in line with timescales at which the (perceived) level of performance is determined. Of course, a similar procedure can be followed for different settings of τ and T — experiments have shown that other settings do not influence our conclusions.

The remaining 'under-the-hood' parameter to be set is C_q, the virtual queue's (virtual) service rate. Clearly, when C_q is too small, say $C_q < \mu$, the system is not stable in that it will never be able to (timely) service all the traffic offered. Hence, we choose $C_q \geq \mu$. On the other hand, if C_q is chosen

much larger than μ, the virtual queue's occupancy will, obviously, be zero at most observation times. As this would lead to an unreliable estimation of the BCD (if any), setting C_q too large is to be avoided.

We have performed numerous experiments to see if there is a generally usable C_q, for instance $C_q = c \cdot \mu$ where c is some constant, that ultimately leads to an accurate approximation of $v_{\mathsf{indirect}}(T)$. We report on these experiments in Appendix B of this thesis. As it turns out, there is no single value of c or C_q that is generally usable. Choosing C_q anywhere between $1 \cdot \mu$ and $2 \cdot \mu$, however, gives reasonable results: $v_{\mathsf{indirect}}(T)$ is always in the same order of magnitude as $v_{\mathsf{direct}}(T)$, and in most cases within 5-20% of each other.

We stress that the impact of an estimation error in $v_{\mathsf{indirect}}(T)$ on our ultimate goal of bandwidth provisioning is somewhat mitigated: in the required bandwidth formula, first the square-root of the variance v is taken, and secondly it then is added to some average rate μ.

The third and last step in our validation is to compare the determined $v_{\mathsf{direct}}(T)$ and the estimated $v_{\mathsf{indirect}}(T)$ with each other. Clearly, if they are close to each other, it means that the estimation is good. We therefore introduce ν as an indicator of the accuracy of the estimation:

$$\nu := \frac{v_{\mathsf{indirect}}(T)}{v_{\mathsf{direct}}(T)}.$$

The approach described above is applied using real traffic, as reported next.

6.3.2 Case-studies

In this section we apply the validation approach outlined in the previous section using the measurement traces. We use the same example traces from each measurement location as in Section 6.2.

The case-study starts with determining the BCD. We use the same two example traces from location R that were used in Section 6.2. Figure 6.12 on page 152 shows the (log)plot of the respective BCDs, where in the case of example trace #1, $C_q = 1.2 \cdot \mu$, and for example trace #2, $C_q = 1.3 \cdot \mu$.

Next, $v(\cdot)$ is estimated at some timescale T; at this point we choose $T =$ 100 milliseconds. Note that this is 10 times as small as the 'measurement

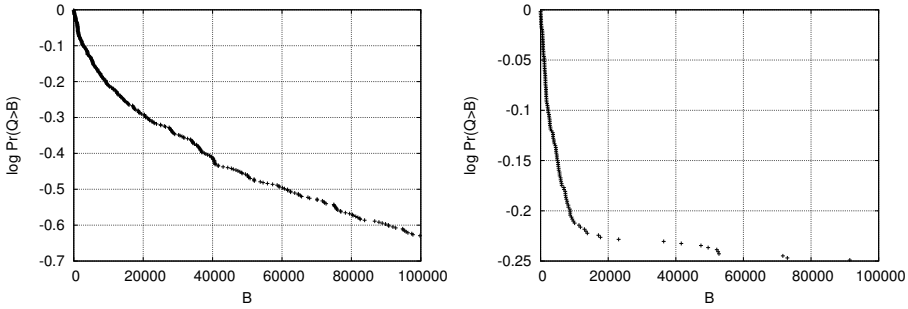
timescale' τ of 1 second: one snapshot of the buffer occupancy is taken every second, and still stochastic characteristics are inferred at the 100 milliseconds timescale. In Figure 6.13, for the entire BCD the corresponding values of $\Phi(B)$ are plotted. Minimization over B then yields the approximation of $v(T)$ at this timescale. Note that, as can been seen especially in the first example trace in Figure 6.13, it may not be exactly clear at which buffer occupancy level B the minimal value of $v(T)$ may be achieved (in that the graph is rather flat). Importantly, as we only need the value of the minimum $v(T)$, however, and not the B at which the minimum of $\Phi(B)$ is achieved, this does not cause (stability) problems for the minimization. In other words, despite the 'flat' behavior of $\Phi(B)$ around the minimum, the procedure is still highly robust.

One of the attractive features of our methodology to indirectly estimate the burstiness through polling of the buffer occupancy, is that it gives the entire variance function $v(t)$ for $t \geq 0$, once the BCD is known. Figure 6.14 shows this feature by plotting $v(t)$ against t, after the BCD is only estimated once.

We have done the same validation as described above for traces from location R, using traces from the other measurement locations. Motivated by our ultimate goal of network link dimensioning, we would like to compare the required bandwidth C_{direct} as estimated using the directly computed variance, with the required bandwidth C_{indirect} as estimated using the variance estimation approach. We have already seen in Section 6.2 that C_{direct} is an accurate estimation of the required bandwidth. Next to that, we would also like to compare the required bandwidth as estimated through the indirect approach, with the 'empirical' minimally required bandwidth, which gives the absolute minimum bandwidth that, according to the trace, suffices to meet the performance criterion:

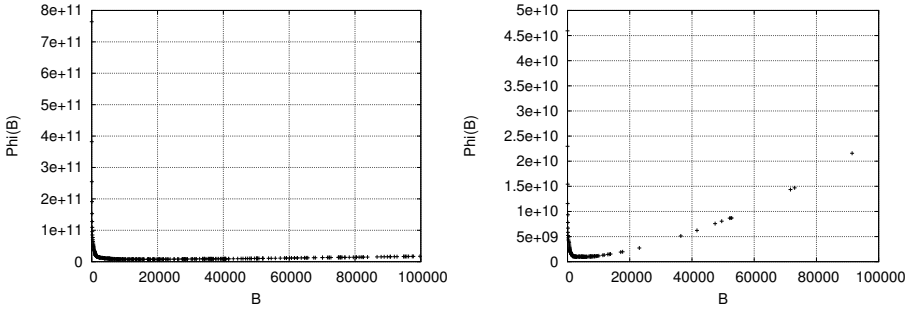
$$C_{\text{empirical}} := \min\left\{C \ : \ \frac{\#\{A_i | A_i > CT\}}{n} \leq \varepsilon\right\}.$$

We introduce Δ as an indicator of the 'goodness' of the estimation of the required bandwidth through the indirect approach. We compare this estima-



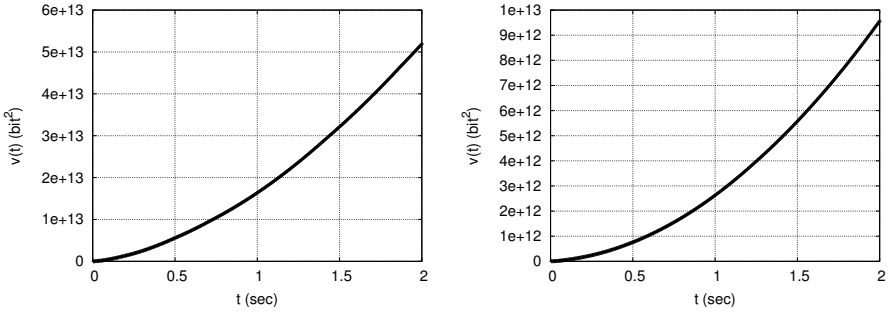
(a) BCD location R, example trace #1 (b) BCD location R, example trace #2

Figure 6.12: Statistical distribution of the buffer occupancy (BCD)



(a) location R, example trace #1 (b) location R, example trace #2

Figure 6.13: $\Phi(B)$ (at timescale $T = 100$ msec)



(a) location R, example trace #1 (b) location R, example trace #2

Figure 6.14: Entire variance function $v(t)$

tion with both the result from the direct approach to determine the variance, i.e., C_{direct}, as well as $C_{\text{empirical}}$. Hence,

$$\Delta_{\text{var}} := \frac{C_{\text{indirect}}}{C_{\text{direct}}},$$

and

$$\Delta_{\text{cap}} := \frac{C_{\text{indirect}}}{C_{\text{empirical}}}.$$

Thus, if Δ_{var} is close to 1, our methodology to (indirectly) estimate the burstiness of network traffic, as proposed in Chapter 5, leads to similar required bandwidth capacity levels as the 'direct' approach, which we validated to be accurate in Section 6.2. If Δ_{cap} is close to 1, our 'indirect' burstiness estimation approach ultimately yields a similar required bandwidth capacity level as the absolute minimum bandwidth that, according to the traces of real traffic, suffices to meet the performance criterion.

Table 6.11 on the following page lists the results of the case-studies, in which we used the same example traces as in Section 6.2. The timescale at which the variances $v(T)$ are determined and estimated, is $T = 100$ milliseconds. It can clearly be seen from Table 6.11 that the variances are rather accurately estimated. Table 6.12 continues the validation results, by comparing the estimated required capacity (with $\varepsilon = 0.01$, $T = 100$ milliseconds) computed via both the direct and indirect approaches, cf. (4.7). Also the empirically found minimum required bandwidth is tabulated.

As can immediately be seen from the values of Δ_{var} in Table 6.12, the required bandwidth capacity as estimated through the indirect approach to estimate the variance, is remarkably close to the direct approach: on average, the differences are less than 1%. Also, comparison with the empirical minimum required bandwidth, through Δ_{cap}, shows that the our coarse-grained measurement (i.e., indirect) procedure leads to estimations for the required bandwidth that are, remarkably, not more than 7% and on average less than 4% off.

Comparing the respective values for v and Δ_{var} in Table 6.11 and Table 6.12, it becomes clear that an estimation error in $v(\cdot)$, indeed, has only limited impact on the error in C.

trace	μ	$\sqrt{v_{\text{direct}}(T)}$	$\sqrt{v_{\text{indirect}}(T)}$	ν
loc. U ex. #1	207.494	1.942	2.006	1.067
loc. U ex. #2	238.773	2.704	2.773	1.052
loc. R ex. #1	18.927	0.701	0.695	0.981
loc. R ex. #2	3.253	0.241	0.249	1.062
loc. C ex. #1	23.894	0.796	0.802	1.018
loc. C ex. #2	162.404	3.263	3.518	1.162
loc. A ex. #1	147.180	0.969	1.032	1.133
loc. A ex. #2	147.984	0.863	0.864	1.003
loc. S ex. #1	14.254	0.447	0.448	1.004
loc. S ex. #2	2.890	0.152	0.152	1.022

Table 6.11: Validation results (rounded) for the burstiness estimation methodology (μ is in Mbit/sec, $\sqrt{v}$s are in Mbit).

trace	$C_{\text{empirical}}$	C_{direct}	C_{indirect}	Δ_{var}	Δ_{cap}
loc. U ex. #1	258.398	266.440	268.385	1.007	1.039
loc. U ex. #2	302.663	320.842	322.934	1.007	1.067
loc. R ex. #1	37.653	40.221	40.020	0.995	1.063
loc. R ex. #2	10.452	10.568	10.793	1.021	1.033
loc. C ex. #1	44.784	48.033	48.250	1.005	1.077
loc. C ex. #2	265.087	261.444	269.182	1.030	1.015
loc. A ex. #1	171.191	176.588	178.480	1.011	1.043
loc. A ex. #2	168.005	174.178	174.218	1.000	1.037
loc. S ex. #1	27.894	27.843	27.873	1.001	0.999
loc. S ex. #2	7.674	7.482	7.532	1.007	0.981

Table 6.12: Validation results for the burstiness estimation methodology (continued) (Cs are in Mbit/sec).

Finally, to gain further insight in the accuracy of our burstiness estimation methodology in the context of link dimensioning, we determine Δ_{var} and Δ_{cap} taking *all* traces into account.

Table 6.13 on the next page shows the averages and standard deviations of both metrics at all five locations. For all locations but R, Δ_{var} is very close to 1, indicating that the required bandwidth as estimated through the indirect approach is equal to the required bandwidth as estimated through the direct approach. Thus, our methodology to infer the burstiness of the traffic without measuring at correspondingly small timescales, gives a reliable estimate of the burstiness for use in network link dimensioning. The deviation

location	avg Δ_{var}	stderr Δ_{var}	avg Δ_{cap}	stderr Δ_{cap}
U	1.00	0.01	1.01	0.07
R	0.96	0.10	0.90	0.19
C	1.00	0.03	1.04	0.11
A	1.00	0.01	1.04	0.02
S	1.00	0.03	0.99	0.10

Table 6.13: Validation results for the burstiness estimation methodology (overall results) — *all traces*

location	avg Δ_{var}	stderr Δ_{var}	avg Δ_{cap}	stderr Δ_{cap}
U	1.00	0.01	1.03	0.06
R	1.00	0.02	1.00	0.10
C	1.00	0.02	1.05	0.08
A	1.00	0.01	1.04	0.01
S	1.00	0.01	1.01	0.05

Table 6.14: Validation results for the burstiness estimation methodology (overall results) — *traces with* $\gamma > 0.9$

in Δ_{var} from 1 may be caused by the observation (see Chapter 3) that traffic at location R is on average 'less Gaussian' compared to the other measurement locations — as our methodology to indirectly estimate the burstiness assumes Gaussian traffic, some error in the resulting estimate can be expected when the traffic is 'not so Gaussian'. To further investigate this, we computed the same statistics as in Table 6.13, but limiting ourselves to all traces that have a (Gaussian) goodness-of-fit $\gamma > 0.9$, i.e., traffic that is 'almost Gaussian'. The results are presented in Table 6.14; the Δ_{var} are all 1.00 now, indicating that the deviation mentioned above in Table 6.13 for location R, is indeed caused by traces that are 'not so Gaussian'.

The values of Δ_{cap} in Tables 6.13 and 6.14 further confirm our earlier observation that the estimated required bandwidth using the indirect approach, is on average only a few percents off the empirically found minimally required bandwidth levels: especially for traces with 'almost Gaussian' traffic (Table 6.14), the differences are less than 5% on average.

6.4 Concluding remarks

In this chapter we have extensively validated our theoretical results on network link dimensioning from Chapters 4 and 5.

The first conclusion is that, after validating the required bandwidth formula (4.7) in Section 6.2, is that the bandwidth estimation is very accurate.

Secondly, we validated our methodology to indirectly estimate the burstiness (variance) of network traffic using course-grained measurements of the buffer occupancy. From this validation study, we may conclude that the burstiness is accurately estimated through our methodology. In our validation scenario, we looked at the variance at a timescales 10 times smaller than the timescale of the buffer occupancy measurements. The estimation of the bandwidth is, generally, off by less than 5%.

7 Conclusions

This chapter presents the contributions and conclusions for the research presented in this thesis, and suggests some directions for further research.

The organization of this chapter is as follows:

- *Section 7.1 gives a brief overview of the research presented in this thesis;*
- *Section 7.2 lists the contributions of this thesis;*
- *Section 7.3 provides answers to the research questions that were formulated in Section 1.4;*
- *Section 7.4 identifies directions for further research.*

7.1 Overview

For network link dimensioning, network operators generally rely on information from coarse measurements: typically, they determine the 5 minute average traffic rate. Network users, however, experience performance at timescales that are orders of magnitudes smaller, in the order of seconds to 100 milliseconds or even less. Hence, traffic rate fluctuations at those small timescales — which are nonnegligible as the peak traffic rates at such timescales can be up to hundreds of percents higher than the 5 minute average rate — should be taken into account for network link dimensioning.

Therefore, adequate network link dimensioning requires thorough insight into the interrelationship between:

- the traffic that is offered to the network link, in terms of the average load but also its fluctuations;
- the desired performance level; and

– the required bandwidth capacity level.

It is clear that the required bandwidth capacity level increases when the average offered traffic load becomes higher, the fluctuations become fiercer, or the desired performance level increases. Insight into the aforementioned interrelationship between these elements of link dimensioning is a necessary condition for making precise predictions about the amount of capacity that should be added. Such predictions are of importance, as too scarce a dimensioning inevitably leads to performance degradation, whereas 'generous' dimensioning policies essentially result in a waste of resources.

The goal of the research presented in this thesis, is to develop network link dimensioning formulas that determine the minimum bandwidth needed to achieve a certain performance level. These formulas should cater for traffic peaks at small timescales, but not require traffic measurements at such timescales. In the course of this thesis, we have achieved this goal by following an approach that stands midway between the areas of traffic measurement procedures, traffic modeling, and queuing theory.

We have performed hundreds of detailed measurements of Internet traffic, at five different locations. These locations have different characteristics in terms of number of users, type of users, etc., increasing the representativeness of our measurements, which ensures that our findings that are based on these measurements can be used in a broad variety of Internet environments. A typical application scenario that we envisage, is that of a small to medium sized organization that wants its Internet-connection to be adequately dimensioned. Other possible application scenarios include that of a Virtual Private Network that interconnects various office locations of a single organization, etc.

After statistical analysis of the measurement data, we have concluded that a Gaussian traffic model can be used to accurately represent real Internet traffic, as long as sufficient aggregation in terms of time and number of users is maintained. A Gaussian model describes the real traffic through the average traffic load, and fluctuations around this load at some timescale T. This timescale T should be chosen such that it corresponds to the timescale that determines the performance that network users experience — the exact choice depends on aspects such as the applications that are used, business level decisions, etc.

We used the Gaussian traffic model as the basis for deriving a network link dimensioning formula that predicts the (minimum) required bandwidth capacity level to achieve a certain performance level, for a traffic stream (described through a Gaussian model).

Although the average traffic rate is easy to determine in practice, determining fluctuations around this mean rate at small timescales is harder, as measuring traffic at small timescales is involved (in practice). We have developed a novel methodology that estimates the fluctuations at small timescales, by coarse-grained polling of the buffer occupancy, thereby removing the need for traffic measurements at small timescales.

Finally, we have verified the link dimensioning formula and our methodology to estimate the traffic rate fluctuations at small timescales, again using large-scale statistical analysis of our measurement data.

Concluding, all the ingredients for network link dimensioning, relying on just coarse measurements, are present. For instance, a procedure to use this research in practice could be summarized as follows:

- determine the average traffic rate through, e.g., standard SNMP measurements;
- estimate the fluctuations of the traffic rate using our new methodology; and
- estimate the required bandwidth by inserting these two traffic characteristics in the required bandwidth formula.

7.2 Contributions

This thesis describes *how to determine the required bandwidth capacity level to meet a prespecified performance target, using network traffic measurements.*

The major contributions of this thesis are:

- A systematic assessment of real Internet traffic: hundreds of detailed measurements of network traffic are performed at 5 different locations. each with different characteristics in terms of number of users, type of users, access link speeds, etc. Analysis of these measurements

shows that real network traffic can often be described using a Gaussian traffic model, as long as there is sufficient aggregation in time (in the order of milliseconds and up) and number of users (some tens or more) (Chapters 2, 3).

- Link dimensioning formulas that give the required bandwidth capacity to achieve link transparency for a given offered traffic load. Formulas are derived for the case of general traffic (i.e., no modeling assumptions are made), for the case of using an $M/G/\infty$ input model, and the case of using a Gaussian model to describe the real network traffic (Chapter 4).
- A novel methodology to estimate the burstiness (fluctuations) of network traffic at small timescales is proposed. This methodology relies on coarse-grained polling of the occupancy of the buffer in front of a network link to, indirectly, infer the burstiness of the offered traffic. Hence, it removes the need for detailed measurements of the network traffic in order to estimate the fluctuations of the traffic at timescales that (we belief) are important to the user's perception of the performance of a network link (Chapter 5).

Next we detail the contributions per chapter:

Chapter 2: Internet traffic measurements

- Provides a state-of-the-art overview of Internet traffic measurement technologies that are relevant to network link dimensioning.
- Describes the measurements that we have performed as part of the research presented in this thesis, and provides general results about these measurements.

Chapter 3: Traffic modeling

- Provides a state-of-the-art overview of traffic models to describe Internet traffic.
- Discusses the advantages and disadvantages of black-box (e.g., Gaussian) and flow-level (e.g., $M/G/\infty$) modeling techniques to describe real Internet traffic in the context of link dimensioning.

– Argues that, based on a comparison of both modeling approaches to describe real traffic, using our measurements, Gaussian traffic modeling is attractive for wide-scale use to describe real network traffic, especially in the context of link dimensioning.

– Shows that real network traffic can accurately be described using a Gaussian traffic model, as long as there is sufficient aggregation in terms of time (i.e., milliseconds or more) and users (i.e., some tens of users or more).

Chapter 4: Bandwidth provisioning rules

– Proposes a generic mathematical formula that estimates the required bandwidth capacity to meet a prespecified performance criterion, without any assumptions on the input traffic process (besides stationarity), as long as this traffic process is known.

– Specializes the generic formula for the case that the input traffic process is described using an $M/G/\infty$ input model.

– Specializes the generic formula for the case that the input traffic process is described using a Gaussian model.

– Presents and evaluates some alternative, empirically derived formulas to estimate the required bandwidth capacity.

Chapter 5: Burstiness estimation

– Reviews basic principles of Gaussian traffic and queues.

– Inspired by earlier work by Addie *et. al* [AMN02], the distribution of the occupancy of a buffer in front of a network link is written as a function of, among other things, the fluctuations of the traffic rate.

– Proposes a new methodology that 'inverts' the above function. The inverted function thus estimates the traffic burstiness (i.e., fluctuations of the traffic rate), with the buffer occupancy distribution as an argument. As the buffer occupancy distribution can be estimated through just coarse measurements, the burstiness of network traffic at small timescales can be estimated without requiring measurements at such timescales, using this methodology.

- Discusses and evaluates the impact of possible numerical errors in this new methodology.
- Outlines an approach to implement the methodology in practice.

Chapter 6: Large-scale validation

- Validates the Gaussian link dimensioning formula from Chapter 4, by comparing the estimated required bandwidth capacity with the information obtained through the measurements.
- Provides a sensitivity analysis of the required bandwidth, with respect to the performance criterion parameters.
- Validates the novel methodology to estimate traffic burstiness through coarse measurements from Chapter 5, by comparing the burstiness as obtained from the detailed measurements, with the burstiness estimated through our proposed methodology.

7.3 Conclusions per research question

This section provides answers to the research questions posed in Section 1.4 of this thesis:

Research question (i): how to perform measurements on a (high-speed) network link with the required (detailed) granularity?

In Chapter 2, we reviewed existing network traffic measurement technologies. SNMP is the commonly used protocol to fetch information about the amount of data sent over a network link from the IF-MIB that is present in most 'manageable' network nodes. SNMP can certainly be used to retrieve information about the long-term average load, for instance based on measurement intervals of 5 minutes.

Fluctuations of the traffic rate at timescales that are relevant for the user's perception of the performance of a network link (which we argue is in the order of milliseconds to seconds), are hard or impossible to measure using SNMP. Alternatively, one could capture the traffic using a tool such as

`libpcap/tcpdump`, which does work at the aforementioned detailed granularity.

Research question (ii): is it possible to infer detailed information (in case needed) about the traffic characteristics, without relying on detailed measurements?

In Chapter 5, we proposed a novel methodology to infer information about the fluctuations of the traffic rate (which we refer to as burstiness in this thesis), by coarse-grained measurements of the buffer occupancy. In Chapter 6 we then (positively) validated this methodology by comparing the estimated burstiness with the burstiness as obtained through `libpcap/tcpdump`, for hundreds of traces of real network traffic.

Research question (iii): which statistical traffic model(s) describe the traffic we have measured 'good enough' to rely on for use in network link dimensioning?

We have found, in Chapter 3, that a Gaussian traffic model can, in most cases, accurately describe real network traffic. From our validation study in Chapter 6, it (implicitly) follows that a Gaussian traffic model (at least) suffices for our goal (i.e., network link dimensioning).

Research question (iv): what is an accurate bandwidth provisioning formula for a given traffic model?

In Chapter 4, we derived bandwidth provisioning (i.e., link dimensioning) formulas for general, $M/G/\infty$ and Gaussian traffic. The formula for Gaussian traffic is: $C = \mu + \frac{1}{T}\sqrt{\left(-2\log\varepsilon\right)v(T)}$.

We validated this required bandwidth formula for Gaussian traffic in Chapter 6, and concluded that it accurately predicts the required bandwidth capacity level. We also validated if this formula gives an accurate prediction of the required bandwidth capacity when our methodology to estimate the fluctuations of the traffic rate $v(T)$ is used, and showed that it does.

7.4 Future research

We suggest some possibilities to continue the research presented in this thesis:

- Although we have found (in Chapter 3 of this thesis) that most traffic streams can be accurately described using a Gaussian traffic model, we have also seen that it is quite common that the 'upper tail' of the Gaussian distribution underestimates the upper tail of real Internet traffic, e.g., in a scenario where the users' access link is high-speed, compared to the average rate of the (aggregated) network traffic. In order to further improve the accuracy of the traffic model, it is interesting to research traffic models that better approximate the upper tail of real network traffic, yet are simple enough for practical use.

- In this thesis we focused on dimensioning of high-speed, wired Internet links. Further research is required to determine whether the models, formulas and methodologies presented in this thesis are also applicable to other types of networks, for instance GPRS or UMTS access networks: wireless, lower speed, and possible less aggregation in terms of (concurrent) users — in such situations, the Central Limit Theorem may no longer apply, and hence, the applicability of a Gaussian traffic model is questionable.

- In this thesis, we have mainly used $\mathbb{P}(A(T) \geq CT) \leq \varepsilon$ as the performance target, for instance in our derivation of the required bandwidth formulas. It would be interesting to study the relation between this performance target and measures such as delay, throughput, etc. It would also be interesting to develop required bandwidth formulas for other performance criteria, e.g., an upper bound on the delay or jitter that is incurred on a network link, as these may better correspond to certain application requirements (for instance, VoIP benefits from low delay and jitter).

- We focused on dimensioning of a single network link in this thesis. Clearly, dimensioning may get more complicated when (optimizing the) dimensioning (of) an entire network: How to guarantee a performance target network-wide? Also, other factors may come into play, e.g., routing can also help in improving network-wide performance.

– Another dimension could be added to this research by incorporating business level (as in economics) elements. For instance, the mapping of resource usage to pricing, or letting the performance target depend on pricing, etc.

A Mathematical background information

The purpose of this appendix is to provide some relevant mathematical background information.

Distribution function

A *cumulative distribution function* (cdf) completely describes the probability distribution of a random variable X. For all x, the cdf $F(x)$ is given by

$$F(x) = \mathbb{P}\big(X \leq x\big).$$

The probability that x lies between a and b is:

$$\mathbb{P}\big(a \leq X \leq b\big) = F(b) - F(a) = \int_a^b f(x)\, dx,$$

where $f(x)$ is called the *probability density function* (pdf). For instance, the standard normal distribution has pdf $f(x) = e^{-x^2/2}/\sqrt{2\pi}$.

The mean (or expected value) of a distribution is given by

$$\mathbb{E}X = \int_{-\infty}^{\infty} x \cdot f(x) dx.$$

When dealing with observations, the *empirical (cumulative) distribution function* (ecdf) is commonly used. The ecdf is a cdf that concentrations probability mass $1/n$ at each of the n observations. Hence,

$$F_n(x) = \frac{\text{number of obserbations } \leq x}{n}.$$

Heavy-tailed distribution

A distribution is said to be *heavy-tailed*, when its probability distribution function assigns relatively high probabilities to regions far from its mean (or median). A more formal condition is given below:

$$\mathbb{P}(X > x) \sim x^{-\alpha} \quad \text{when } x \to \infty, 1 < \alpha < 2 \ .$$

An interpretation of this is as follows. Regardless of the distribution for small values of a random variable X, the distribution of X is heavy-tailed if the asymptotic shape of the distribution is hyperbolic (with $1 < \alpha < 2$). For $1 < \alpha < 2$, the mean $\mathbb{E}X$ of the probability distribution exists, but the distribution has infinite variance.

A relevant example: if one considers the sizes of files transferred over a network link, then, the distribution of these sizes is heavy-tailed if there are a large number of small files transferred, but (crucially) the number of very large files transferred is still significant. This is common on Internet links.

A well-known distribution that is heavy-tailed, is the Pareto distribution (hyperbolic over its entire range).

Moment-generating function

A moment-generating function allows for computation of the moments of a distribution. By definition, the first moment is the mean, the second moment the variance. The moment-generating function $M_X(\theta)$ of a random variable X is given by:

$$M_X(\theta) := \mathbb{E}\left(e^{\theta X}\right), \quad \theta \in \mathbb{R},$$

wherever this expectation exists.

The moment-generating function for a random variable X which has a continuous probability density function $f(x)$, is given by:

$$\begin{aligned} M_X(\theta) &:= \int_{-\infty}^{\infty} e^{\theta X} f(x)\, dx \\ &= \int_{-\infty}^{\infty} \left(1 + \theta x + \frac{\theta^2 x^2}{2!} + \cdots\right) f(x)\, dx \\ &= 1 + \theta M^{(1)} + \frac{\theta^2 M^{(2)}}{2} + \cdots, \end{aligned}$$

where $M^{(i)}$ is the ith moment.

Using the above, the ith moment for the random variable $A(T)$ (as used throughout this thesis) can be found as follows:

$$\mathbb{E}\left[\left(A(T)\right)^{(i)}\right] = \frac{d^i}{d\theta^i}\left[\mathbb{E}e^{\theta A(T)}\right]\Big|_{\theta=0}$$

B Addendum burstiness estimation validation

The purpose of this Appendix is to investigate the parameter C_q in the inversion formula, which we introduced in Chapters 5 and 6 of this thesis.

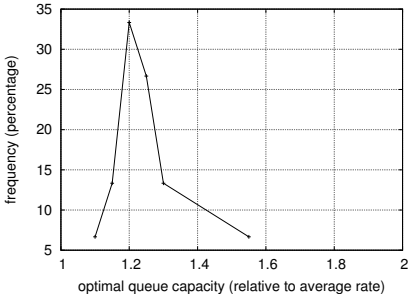
The inversion formula is as follows, cf. (5.9) on page 111:

$$v(t) \approx \inf_{B>0} \frac{(B + (C_q - \mu)\, t)^2}{-2 \log \mathbb{P}(Q > B)},$$

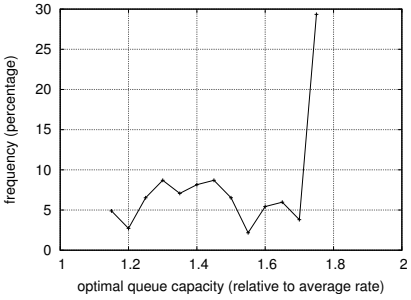
where $v(t)$ denotes the traffic's variance, B denotes the buffer occupancy, C_q denotes the drain capacity of a queue, and μ denotes the average traffic rate.

In operational environments, C_q could be the link rate of the link that may need to be upgraded. By polling the occupancy of the (output) buffer in front of that link, one estimate the buffer occupancy distribution and subsequently estimate the variance of the traffic — after inserting the variance into the required bandwidth formula (e.g., (4.7) on page 88), one could determine whether the link needs to be upgraded.

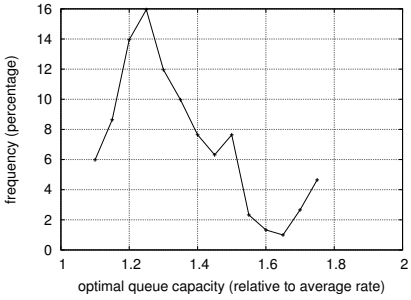
In the traffic measurement environments that we reported on in this thesis, however, the (measured) link is (sometimes excessively) overprovisioned. In such cases, it is unlikely that an (output) buffer would be filling up, and hence, one would likely not be able to get an accurate estimate of the buffer occupancy distribution. Therefore, in the validation study of the inversion formula in Section 6.3, we used a virtual queue — see Figure 5.9 on page 124 — to simulate a buffer that fills with input traffic and drains at some rate C_q. The bottom line is that the buffer (in our case the virtual queue) should be



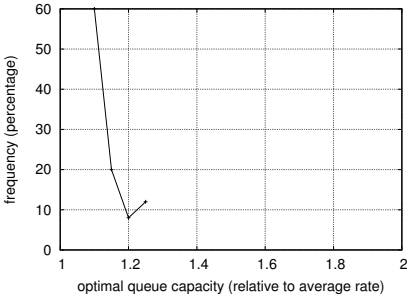
(a) Location U



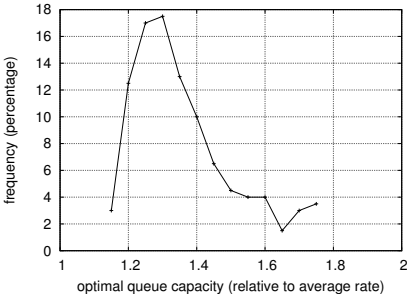
(b) Location R



(c) Location C



(d) Location A



(e) Location S

Figure B.1: Frequency distribution of the 'optimal c' in $C_q = c \cdot \mu$

filled 'sufficiently often' such that a 'realistic buffer occupancy distribution' can be estimated.

An obvious choice is to take $C_q = c \cdot \mu$, where $c > 1$ is some constant. In order to investigate if there is some c that (always) ultimately yields an accurate estimate of the variance $v(t)$, we 'replay' each traffic trace through a virtual queue with drain capacity $c \cdot \mu$, for $c = 1, 1.05, 1.10, \ldots, 1.95, 2$, and μ the 15 minute average throughput of that trace. For each trace, we then determine the 'optimal c', i.e., the value of c which yields in the best estimate of the variance (at timescale $T = 100$ msec), compared to the variance estimated through the direct approach.

Figure B.1 reports on these experiments. The horizontal axes represent the optimal c, whereas the vertical axes represent the relative frequency that a certain c is considered optimal (across all traces at each location).

A first observation is that it becomes clear from Figure B.1 that there is no 'generally optimal c', as the optimal c that is found most often, differs from location to location: whereas at location U the optimal C_q is often $1.2 \cdot \mu$, at location R the best estimate of the variance is most likely obtained using $C_q = 1.75 \cdot \mu$.

A second observation from Figure B.1 is that, at all locations, there is a single (or a few similar) c that most often gives the best estimates for $v(T)$. The advantage of this, is that the 'optimal c' for a specific location apparently is stable. The precise value of the 'optimal c' at a location likely depends on the burstiness of the traffic itself (and possibly other aspects as well): it intuitively is clear that when traffic is very bursty, it is more likely to fill a buffer, compared to traffic that is not so bursty. For instance, when the access link speeds are relatively high (compared to the 'uplink' capacity), a single user can cause bursts with relatively high traffic rates on the uplink — e.g., such is the case at location R — which leads to relatively high variability of the traffic. Compared to location A, where the traffic is relatively smooth (i.e., not so bursty), at location R the 'optimal c' indeed is higher (1.75 and 1.25 for locations R and A, respectively, according to Figure B.1). It is noted that when a 'non-optimal c' is used, this ultimately leads to some error in the estimate of the burstiness $v(\cdot)$. Experiments have shown, however, that these errors are small in that the estimate of $v(\cdot)$ while using a 'non-optimal c', is generally within 25% of the best estimate.

Finally, as a last remark, recall from Chapter 6 that estimation errors in $\nu(\cdot)$ only have limited impact on the ultimate required bandwidth estimation. Therefore we argue that the above conclusion that there is no 'generally optimal c' does not lead to fundamental problems with our burstiness estimation approach. Should the need for using a virtual queue arise (because of an overprovisioning situation, for instance), a reasonable choice (possibly to determine through 'trial and error') for c suffices to get accurate estimates of the burstiness, i.e., $\nu(\cdot)$.

References

[ACD+97] N. Anerousis, R. Caceres, N. Duffield, A. Feldmann, A. Greenberg, C. Kalmanek, P. Mishra, K.K. Ramakrishnan, and J. Rexford. Using the AT&T Labs PacketScope for Internet Measurement, Design, and Performance Analysis. AT&T technical report, October 1997.

[ACFG04] N. Ben Azzouna, F. Clerot, C. Fricker, and F. Guillemin. Modeling ADSL traffic on an IP backbone link. *Annals of Telecommunications*, 59(11-12), November-December 2004.

[ACTW96] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder. OC3MON: Flexible, Affordable, High Performance Statistics Collection. In *Proceedings of the 10th USENIX conference on System Administration*, pages 97–112, Chicago, IL, USA, 1996.

[AMN02] R. Addie, P. Mannersalo, and I. Norros. Most probable paths and performance formulae for buffers with Gaussian input traffic. *European Transactions on Telecommunications*, 13(3):183–196, 2002.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. IETF RFC 2475, December 1998.

[BCS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: An Overview. IETF RFC 1633, July 1994.

[BD98] O.J. Boxma and V. Dumas. Fluid queues with long-tailed activity period distributions. *Computer Communications*, 17(21):1509–1529, November 1998.

[Ber92] J. Beran. Statistical methods for data with long-range dependence. *Statistical Science*, 7(4):404–416, November 1992.

[Ber94] J. Beran. *Statistics for Long-Memory Processes*. Chapman & Hall/CRC, 1994.

[BL01] N. Brownlee and C. Loosley. Fundamentals of Internet Measurement: A Tutorial. *CMG Journal of Computer Resource Management*, -(102), Spring 2001.

[Bla02] E. Blanton. Tcpurify, 2002. `http://masaka.cs.ohiou.edu/~eblanton/tcpurify/`.

[BMR99] N. Brownlee, C. Mills, and G. Ruth. Traffic Flow Measurement: Architecture, October 1999. IETF RFC 2722.

[BOR03] T. Bonald, P. Olivier, and J. Roberts. Dimensioning high speed IP access networks. In *Proceedings of 18th International Teletraffic Congress (ITC 2003)*, pages 241–251, Berlin, Germany, 2003.

[BP01] P. Barford and D. Plonka. Characteristics of Network Traffic Flow Anomalies. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 69–73, San Francisco, CA, USA, 2001.

[Bro02] N. Brownlee. NeTraMet - a Network Traffic Flow Measurement Tool, 2002. `http://www.caida.org/tools/measurement/netramet/`.

[BTI+03] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski. Modeling Internet backbone traffic at the flow level. *IEEE Transactions on Signal Processing*, 51(8), August 2003.

[CDF+00] R. Cáceres, N. Duffield, A. Feldmann, J. Friedmann, et al. Measurement and analysis of IP network usage and behavior. *IEEE Communications Magazine*, pages 144–151, May 2000.

[CKR+95] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. Weber. Admission control and routing in ATM networks using inferences from measured buffered occupancy. *IEEE Transactions on Communications*, 43:1778–1784, 1995.

[Die] A.B. Dieker. Fractional Brownian motion simulator. `http://homepages.cwi.nl/~ton/fbm/index.html`.

[DLO+95] N.G. Duffield, J.T. Lewis, N. O'Connell, R. Russell, and F. Toomey. Entropy of ATM traffic streams: a tool for estimating quality of service parameters. *IEEE Journal on Selected Areas in Communications*, 13:981–990, 1995.

[DM03a] K. Dębicki and M.R.H. Mandjes. Exact overflow asymptotics for queues with many Gaussian inputs. *Journal of Applied Probability*, 40:704–720, 2003.

[DM03b] A.B. Dieker and M.R.H. Mandjes. On Spectral Simulation Of Fractional Brownian Motion. *Probability in the Engineering and Informational Sciences*, 17(3):417–434, 2003.

[DS86] R. B. D'Agostino and M. A. Stephens, editors. *Goodness-of-fit techniques*. Marcel Dekker, Inc, 1986.

[EM93] A.I. Elwalid and D. Mitra. Effective bandwidth of general Markovian traffic sources andadmission control of high speed networks. *IEEE/ACM Transactions on Networking*, 1(3):329–343, June 1993.

[eth04] Ethereal: A Network Protocol Analyzer, 2004. `http://www.ethereal.com/`.

[FJ93] S. Floyd and V. Jacobson. Random Early Detection (RED) gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[FML+03] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot. Packet-Level Traffic Measurements from the Sprint IP Backbone. *IEEE Network*, 17(6), November 2003.

[Fra02] C. Fraleigh. *Provisioning Internet Backbone Networks to Support Latency Sensitive Applications*. PhD thesis, Stanford University, May 2002.

[FTD03] C. Fraleigh, F. Tobagi, and C. Diot. Provisioning IP Backbone Networks to Support Latency Sensitive Traffic. In *Proceedings of IEEE Infocom*, San Francisco, USA, April 2003.

[GAN91] R. Guérin, H. Ahmadi, and M. Naghsineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, September 1991.

[GM01] L. Guo and I. Matta. The war between mice and elephants. In *Proceedings of the 9th International Conference on Network Protocols 2001 (ICNP 2001)*, pages 180–188, Riverside, CA, USA, November 2001.

[Hui88] J. Y. Hui. Resource allocation for broadband networks. *IEEE Journal on Selected Areas in Communications*, 6:1598–1608, 1988.

[HvdM05] G.K. Haan and R. van de Meent. Towards detection of portscans using IP header data. In C.D. Kloos et al., editor, *Proceedings of the 11th Open European Summer School and IFIP WG 6.6, WG 6.4 and*

WG 6.9 Workshop (EUNICE 2005), pages 275–282, Madrid, Spain, July 2005.

[HvdMP02] H.E. Holland, R. van de Meent, and A. Pras. Evaluating MIB II (RFC1213) Implementations. *The Simple Times*, 10(1), December 2002.

[iet04a] IETF IP Flow Information Export Working Group Charter, 2004. `http://www.ietf.org/html.charters/psamp-charter.html`.

[iet04b] IETF Packet Sampling Working Group Charter, 2004. `http://www.ietf.org/html.charters/psamp-charter.html`.

[Int89] International Organization for Standardization. Information processing systems - Open Systems Interconnection, Basic Reference Model, Part 4: Management Framework, 1989. ISO/IEC 7894-4.

[Ips97] Ipsilon Networks. tcpdpriv, 1997. `http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html`.

[Jen48] A. Jensen. *The life and works of A.K. Erlang*, volume 2, chapter An elucidation of A.K. Erlang's statistical works through the theory of stochastic processes, pages 23–100. Transactions of the Danish Academy of Technical Sciences, 1948.

[JSPA05] I. Juva, R. Susitaival, M. Peuhkuri, and S. Aalto. Traffic Characterization for Traffic Engineering Purposes: Analysis of Funet Data. In *Proceedings of the 1st EuroNGI Conference on Next Generation Internet Networks Traffic Engineering*, Rome, Italy, April 2005. IEEE.

[Kau81] J. Kaufman. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 29(10):1474–1481, 1981.

[Kel91] F.P. Kelly. Loss networks. *The Annals of Applied Probability*, 1(3):319–378, 1991.

[Kel96] F. Kelly. *Stochastic networks: theory and applications*, chapter Notes on effective bandwidth, pages 141–168. Oxford University Press, September 1996.

[KHF03] E. Kohler, M. Handley, and S. Floyd. Designing DCCP: Congestion control without reliability. Available at `http://www.icir.org/kohler/dccp/`, May 2003.

[KN02] J. Kilpi and I. Norros. Testing the Gaussian approximation of aggregate traffic. In *Proceedings of the 2nd ACM SIGCOMM Internet Measurement Workshop*, pages 49–61, Marseille, France, 2002.

[Law05a] Lawrence Berkeley National Laboratory Network Research. libpcap: Packet Capture library, 2005. http://www.tcpdump.org/.

[Law05b] Lawrence Berkeley National Laboratory Network Research. TCPDump: the Protocol Packet Capture and Dumper Program, 2005. http://www.tcpdump.org/.

[LTWW94] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 1(2):1–15, February 1994.

[Man04] M.R.H. Mandjes. A note on the benefits of buffering. *Stochastic Models*, 20:43–53, 2004.

[MK00] K. McCloghrie and F. KastenHolz. The Interfaces Group MIB, June 2000. IETF RFC 2863.

[MK01] M.R.H. Mandjes and J.H. Kim. An analysis of the phase transition phenomenon in packet networks. *Advances in Applied Probability*, 33:260–280, 2001.

[MSS05] M.R.H. Mandjes, I. Saniee, and A. Stolyar. Load characterization, overload prediction, and load anomaly detection for voice over IP traffic. *IEEE Transactions on Neural Networks*, 16:1019–1028, 2005.

[MTK02] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessment of VoIP Quality over Internet Backbones. In *Proceedings of IEEE INFOCOM*, pages 150–159, New York, NY, USA, 2002.

[MvdM05] M.R.H. Mandjes and R. van de Meent. Inferring traffic characteristics by observing the buffer content distribution. In R. Boutaba et al., editor, *Proceedings of the 4th International IFIP-TC6 Networking Conference (NETWORKING 2005)*, number 3462 in Lecture Notes in Computer Science (LNCS), pages 303–315, Waterloo, Canada, May 2005.

[net04] Cisco IOS Software NetFlow, 2004. http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml.

[Nie91] L.J.M. Nieuwenhuis. *Fault Tolerance through Program Transformation*. PhD thesis, University of Twente, 1991.

[Nor94] I. Norros. A Storage Model with Self-Similar Input. *Queuing Systems*, 16:387–396, 1994.

[Nor95] I. Norros. On the Use of fractional Brownian motion in the Theory of Connectionless Networks. *IEEE Journal of Selected Areas in Communications*, 13(6):953–962, 1995.

[Odl03] A. M. Odlyzko. Data networks are lightly utilized, and will stay that way. *Review of Network Economics*, 2(3):210–237, Sep 2003.

[Oet03] Tobias Oetiker. MRTG: Multi Router Traffic Grapher, 2003. `http://people.ee.ethz.ch/~oetiker/webtools/mrtg/`.

[Onv94] R.O. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues*. Artech House, Norwood, 1994.

[Pap03] K. Papagiannaki. *Provisioning IP Backbone Networks Based on Measurements*. PhD thesis, University of London, 2003.

[Pax94] V. Paxson. Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, August 1994.

[PE95] P. Pruthi and A. Erramilli. Heavy-tailed on/off source behavior and self-similar traffic. In *Proceedings of the IEEE International Conference on Communications (ICC95)*, pages 445–450, Seattle, USA, 1995.

[PF95] V. Paxson and S. Floyd. Wide area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[PM97] M. Parulekar and A. M. Makowski. M/G/∞ input processes: A versatile class of models for network traffic. In *Proceedings of IEEE INFOCOM*, pages 419–426, Kobe, Japan, April 1997.

[Pop71] Karl Raimund Popper. *Open Society and Its Enemies*, volume 1 & 2. Princeton University Press, 5th Revise edition, 1971.

[Pra95] A. Pras. *Network Management Architectures*. PhD thesis, University of Twente, 1995.

[PvdMM05] A. Pras, R. van de Meent, and M.R.H. Mandjes. QoS in Hybrid Networks - An Operator's Perspective. In Hermann de Meer and Nina T. Bhatti, editors, *Proceedings of the 13th International Workshop on Quality of Service (IWQoS 2005)*, volume 3552 of *Lecture Notes in Computer Science (LNCS)*, pages 388–391, Passau, Germany, June 2005.

[PW02] L. Pantel and L.C. Wolf. On the impact of delay on real-time multiplayer games. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video (NOSSDAV 2002)*, pages 23–29, Miami Beach, FL, USA, 2002.

[QZCZ02] J. Quittek, T. Zseby, G. Carle, and S. Zander. Traffic Flow Measurements within IP Networks: Requirements, Technology, and Standardization. In *Proceedings of the SAINT Symposium on Applications and the Internet Workshops*, Nara, Japan, January 2002.

[RMV96] J. Roberts, U. Mocci, and J. Virtamo. *Broadband network traffic: performance evaluation and design of broadband multiservice networks*. Springer, 1996. Final report of action COST 242.

[Rob81] J.W. Roberts. A service system with heterogeneous service requirements – applications to multi-service telecommunications systems. In G. Pujolle, editor, *Proceedings of Performance of Data-communications Systems and their Applications*, pages 423–431, Amsterdam, Netherlands, 1981.

[SF01] K. Salamatian and S. Fdida. Measurement Based Modeling of Quality of Service in the Internet: A Methodological Approach. In S. Palazzo, editor, *Proceedings of the International Workshop on Digital Communications (IWDC 2001)*, number 2170 in Lecture Notes in Computer Science (LNCS), pages 158–174, Taormina, Italy, September 2001.

[Soh93] K. Sohraby. On the theory of general on-off sources with applications in high-speed networks. In *Proceedings of IEEE INFOCOM*, pages 401–410, San Francisco, CA, USA, 1993.

[Tan02] A.S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, 4th edition, 2002.

[TGV00] P. Tran-Gia and N. Vicari, editors. *Impacts of New Services on the Architecture and Performance of Broadband Networks*. , 2000. Final report of action COST 257.

[Tha01] O.J. Thatcher, editor. *The Library of Original Sources*, volume V: The Early Medieval World. pages 369-376, 1901.

[Tij94] H.C. Tijms. *Stochastic models : an algorithmic approach*. Wiley, Chichester, New York, 1994.

[Uni96] International Telecommunications Union. Methods for subjective determination of transmission quality, 1996. Recommendation ITU-T/P.800.

[Uni01] International Telecommunications Union. Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs, 2001. Recommendation ITU-T/P.862.

[Uni03] International Telecommunications Union. The e-model, a computational model for use in transmission planning, 2003. Recommendation ITU-T/G.107.

[vdBMvdM+06] J.L. van den Berg, M.R.H. Mandjes, R. van de Meent, A. Pras, F. Roijers, and P. Venemans. QoS-aware bandwidth provisioning of IP links. *Computer Networks*, 50(5):631–647, April 2006.

[vdMM05] R. van de Meent and M.R.H. Mandjes. Evaluation of 'user-oriented' and 'black-box' traffic models for link provisioning. In *Proceedings of the 1st EuroNGI Conference on Next Generation Internet Networks Traffic Engineering*, Rome, Italy, April 2005. IEEE.

[vdMMP06] R. van de Meent, M.R.H. Mandjes, and A. Pras. Gaussian traffic everywhere? In *Proceedings of the 2006 IEEE International Conference on Communications (ICC 2006)*, Istanbul, Turkey, June 2006.

[vdMPM+03] R. van de Meent, A. Pras, M.R.H. Mandjes, J.L. van den Berg, and L.J.M. Nieuwenhuis. Traffic Measurements for Link Dimensioning: A Case Study. In M. Brunner and A. Keller, editors, *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM2003)*, number 2867 in Lecture Notes in Computer Science (LNCS), pages 106–117, Heidelberg, Germany, October 2003.

[vdMPM+04] R. van de Meent, A. Pras, M.R.H. Mandjes, J.L. van den Berg, F. Roijers, P. Venemans, and L.J.M. Nieuwenhuis. Burstiness predictions based on rough network traffic measurements. In *Proceedings of the 19th World Telecommunications Congress (WTC/ISS 2004)*, Seoul, South-Korea, September 2004.

[Wal00] S. Waldbusser. Remote Network Monitoring Management Information Base, May 2000.

[WAL04] W. Willinger, D. Alderson, and L. Li. A pragmatic approach to dealing with high-variability in network measurements. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 88–100, Taormina, Sicily, Italy, 2004.

[WP98] W. Willinger and V. Paxson. Where Mathemetics Meets the Internet. *Notices of the AMS*, 45(8):961–970, September 1998.

[ZOS00] W. Zhao, D. Olshefski, and H. Schulzrinne. Internet Quality of Service: an Overview. Technical report, Columbia University, December 2000. CUCS-003-00.

[ZZX01] L. Zheng, L. Zhang, and D. Xu. Characteristics of network delay and delay jitter and its effect on voice over IP (VoIP). In *Proceedings of the IEEE International Conference on Communications (ICC 2001)*, pages 122–126, Helsinki, Finland, June 2001.

List of symbols

$A(T)$	amount of traffic offered over an interval of length T
$A(s,t)$	amount of traffic offered over the interval $[s,t]$
$A(t)$	amount of traffic offered over the interval $[0,t]$
A_i	amount of traffic offered in ith slot of length T
B	amount of data in a buffer or queue
C	link bandwidth capacity level
C_q	capacity (i.e., drain rate) of a virtual queue
D	flow duration distribution, with mean $\delta = \mathbb{E}D$
ε	'blocking' or 'overflow' probability
γ	linear correlation coefficient (i.e., goodness-of-fit)
λ	flow arrival rate
μ	(long-term) average or mean traffic rate
r	traffic rate (i.e., amount of traffic offered per time interval)
t	time
$v(T)$	variance of the offered traffic at timescale T, i.e., $\mathrm{Var}(A(T))$
$v(\cdot)$	entire variance curve of the offered traffic

Index

Samenvatting

Het adequaat dimensioneren van netwerkverbindingen vereist een goed inzicht in het onderlinge verband tussen (i) het netwerkverkeer dat aangeboden wordt (in termen van het gemiddelde aanbod maar ook de fluctuaties rondom dat gemiddelde), (ii) het gewenste kwaliteitsniveau en (iii) de benodigde bandbreedte. Uiteraard is meer bandbreedte benodigd als het gemiddelde verkeersaanbod groeit, de fluctuaties toenemen, of het gewenste kwaliteitsniveau stijgt.

Vaak worden netwerkverbindingen gedimensioneerd aan de hand van vuistregels, bijvoorbeeld: 'bepaal het gemiddelde verkeersaanbod gedurende de drukke uren, en tel hier 30% bij op om de fluctuaties aan te kunnen'. Dergelijke vuistregels houden niet expliciet rekening met de sterkte van de fluctuaties van het verkeersaanbod, noch met het gewenste kwaliteitsniveau.

Een gebruikelijke methode om het gemiddelde verkeersaanbod te bepalen, is de volgende. Een netwerkbeheerder leest regelmatig de zogeheten 'Interfaces Group MIB' uit, via het 'Simple Network Management Protocol' (SNMP), meestal door middel van een programma als de 'Multi-Router Traffic Grapher' (MRTG). Hieruit is het gemiddelde verkeersaanbod van een verbinding af te leiden. Het is gebruikelijk dit ongeveer eens per 5 minuten te doen. Deze methode stelt de beheerder niet in staat om inzicht te verkrijgen in de sterkte van de fluctuaties van het verkeersaanbod binnen deze 5 minuten. Het is echter bekend dat deze fluctuaties behoorlijk groot kunnen zijn, en merkbaar voor netwerkgebruikers. Stel bijvoorbeeld dat op een tijdsschaal van 5 seconden meer verkeer aangeboden wordt dan de netwerkverbinding kan afhandelen. Dan kan een deel van het aangeboden verkeer verloren gaan. Het is algemeen bekend dat een dergelijk verlies kan leiden tot een voor netwerkgebruikers merkbare kwaliteitsverslechtering. Zo kun-

nen bijvoorbeeld hele woorden in een telefoongesprek over Internet verloren gaan. Het is dus in het belang van netwerkgebruikers, en uiteindelijk ook van leveranciers van netwerkverbindingen, dat voldoende bandbreedte beschikbaar is om het verkeersaanbod goed af te handelen, ook op tijdsschalen die veel kleiner zijn dan 5 minuten.

Dit proefschrift beschrijft een alternatieve methode om netwerkverbindingen te dimensioneren. Deze methode houdt expliciet rekening met het gemiddelde verkeersaanbod, de sterkte van de fluctuaties van dit verkeersaanbod op kleine tijdsschalen en het gewenste kwaliteitsniveau. Dit wordt uitgedrukt door middel van wiskundige formules waarmee, gegeven de karakteristieken van het aangeboden verkeer en het gewenste kwaliteitsniveau, de benodigde bandbreedte berekend kan worden.

Verkeersmodellen beschrijven de karakteristieken van het aangeboden verkeer. Om goede verkeersmodellen te vinden, zijn honderden gedetailleerde metingen aan netwerkverkeer uitgevoerd, op vijf verschillende locaties in Nederland. Omdat deze locaties verschillen in types en aantallen gebruikers, applicaties, netwerktechnologieën, etc., is een brede inzetbaarheid van de gekozen verkeersmodellen gewaarborgd.

We hebben geconcludeerd dat een zogeheten Gaussisch verkeersmodel, in het algemeen, goed aansluit bij echt netwerkverkeer. Een Gaussisch verkeersmodel beschrijft netwerkverkeer als volgt: de hoeveelheid aangeboden verkeer $A(T)$, aangeboden over een interval van duur T, is verdeeld volgens de Gaussische (normale) verdeling, en wel met gemiddelde $\mathbb{E}A(T)$ en variantie $\mathrm{Var}A(T)$. Vaak geziene karakteristieken van Internet verkeer, zoals 'long-range dependency' en 'self-similarity', passen in het kader van Gaussische verkeersmodellering.

De kwaliteitsmaat waarmee we in dit proefschrift werken, richt zich op het transparant maken van de netwerkverbinding voor de gebruiker; in niet meer dan een fractie ε van de intervallen van duur T mag de hoeveelheid aangeboden verkeer $A(T)$ groter zijn dan de beschikbare bandbreedte C. Anders gezegd: $\mathbb{P}\left(A(T) \geq CT\right) \leq \varepsilon$.

We tonen in dit proefschrift aan dat voor Gaussisch verkeer geldt dat de volgende formule de benodigde bandbreedte schat om aan bovenstaande kwaliteitseis te voldoen: $C = \mu + 1/T \cdot \sqrt{(-2\log\varepsilon) \cdot \mathrm{Var}A(T)}$, waarbij μ het gemiddelde verkeersaanbod voorstelt.

Het gemiddelde verkeersaanbod μ kan bepaald worden met behulp van de gebruikelijke methode, via relatief ruwe SNMP metingen. Het bepalen van de fluctuaties van het verkeersaanbod, $\mathrm{Var}A(T)$, vereist metingen op tijdsschaal T. Aangezien T waarschijnlijk klein zal zijn (in de orde van seconden of kleiner, in lijn met een tijdsschaal die bepalend is voor de kwaliteit), lijkt het nodig te zijn om relatief gedetailleerde metingen te doen om $\mathrm{Var}A(T)$ te bepalen. Het is (in het algemeen) niet haalbaar om dit soort gedetailleerde metingen te doen met SNMP. In dit proefschrift ontwikkelen we een alternatieve methode om $\mathrm{Var}A(T)$ te schatten.

Onze methode om $\mathrm{Var}A(T)$ te schatten maakt gebruik van relatief ruwe metingen aan de bezetting van een buffer voor de netwerkverbinding die we willen dimensioneren. Deze ruwe metingen zijn vergelijkbaar met de metingen die nodig zijn om μ te schatten. Door regelmatig de bufferbezetting te bepalen, kan de empirische verdelingsfunctie worden bepaald van deze bufferbezetting. We leiden in dit proefschrift een formule af die deze verdelingsfunctie 'inverteert' tot $\mathrm{Var}A(T)$. Merk op dat we hierdoor in staat zijn om $\mathrm{Var}A(T)$ te schatten zonder metingen op (kleine) tijdsschaal T nodig te hebben.

We hebben onze alternatieve methode om netwerkverbindingen te dimensioneren, in het bijzonder de formule voor de benodigde bandbreedte en de 'inversie' om $\mathrm{Var}A(T)$ te schatten, uitgebreid gevalideerd. Bij deze validatie is gebruik gemaakt van de honderden metingen van echt netwerkverkeer.

Beheerders van netwerken kunnen de resultaten van dit onderzoek gebruiken voor netwerk-dimensionering. Een voorbeeld is het bepalen van de benodigde capaciteit om in een aanbodsgroei te voorzien zonder dat kwaliteitsverslechtering optreedt, of om aan 'Service Level Agreements' te kunnen voldoen. Te denken valt hierbij bijvoorbeeld aan kleine tot middelgrote organisaties die een adequaat gedimensioneerde verbinding met Internet willen hebben, of een 'Virtual Private Network' tussen diverse locaties van die organisatie willen opzetten.

Dankwoord

Hoewel ook mijn promotietraject zeker zijn eenzame kanten gekend heeft, mag ik me gelukkig prijzen dat een heleboel mensen in meer of minder directe zin hebben willen bijdragen aan een succesvolle afronding.

Op de eerste plaats wil ik bedanken mijn promotoren Michel Mandjes en Bart Nieuwenhuis, en mijn directe begeleiders Aiko Pras en Hans van den Berg. Aiko heeft me, nu bijna 5 jaar geleden, overtuigd van de meerwaarde van een promotietraject na een academische studie en me in contact gebracht met mijn latere promotoren. Aiko, gedurende de afgelopen jaren heb ik je mogen meemaken als begeleider en als mentor. Bedankt voor al je steun. Michel, jij was m'n wetenschappelijke tutor de afgelopen jaren; een rol die je met glans vervuld hebt. Bart had oog voor detail en de grote lijn tegelijk. Hans speelde zeker in het begin een belangrijke rol in het opzetten van het onderzoek. Gezamenlijk hebben we de brug geslagen tussen wat we noemden de praktijk (de metingen van Internet verkeer) en de theorie (de wiskundige modellering van dat verkeer en het daarop loslaten van formules), en hebben we ons verbaasd over hoe goed de theorie bleek aan te sluiten bij de praktijk. Naast jullie inhoudelijke inbreng, was het ook buitengewoon prettig om met jullie samen te werken.

De metingen die we verricht hebben voor ons onderzoek hadden niet gedaan kunnen worden zonder de medewerking van de diensten ITBE en B&O van de Universiteit Twente, SURFnet en Virtu Secure Webservices. Bedankt dat jullie ons hebben toegestaan meet-apparatuur te verbinden met jullie netwerken, om zodoende inzicht te verkrijgen in wat er 'in het echt' gebeurt, in plaats van ons te moeten baseren op lab-experimenten. Vooral het Telematica Instituut wil ik bedanken voor de financiële middelen die het onderzoek mogelijk gemaakt hebben.

De sfeer binnen onze groep Architecture and Services of Network Applications is, in goede en minder goede tijden, altijd fantastisch geweest. Met name wil ik noemen Aart, Annelies, Bert, Dick, Giancarlo, Ing, João Paulo, Luís, Maarten, Marten, Patrícia, Remco, Róbert en Tom. Ook niet onvermeld mogen blijven Helen, Marlous, Pieter-Tjerk en Wilma. Bedankt jullie allemaal voor de directe en minder directe bijdragen, en vooral ook de gezelligheid zowel binnen als buiten werktijd.

Familie en vrienden zijn altijd belangrijk in het leven. Tijdens zowel de moeilijkere als minder moeilijke tijden gedurende een promotietraject is hun steun onmisbaar. Oma, Ans en Jos, Els, Jan Willem, Annemarie, Marjolein, Stijn, Casper Joost en Sjoerd: jullie hebben elk op jullie eigen manier bijgedragen aan de voltooiing van m'n promotie. Bedankt pap, mam en Liesbeth, voor jullie voortdurende steun, begrip en wijsheid. Weet dat ik het óók voor jullie gedaan heb. Lieve Olga, je hebt een nieuwe betekenis gegeven aan mijn leven, me geholpen met alle laatste loodjes, en nu gaan we samen de toekomst tegemoet.

Remco van de Meent
Enschede, februari 2006.

About the author

Remco van de Meent was born in Venray (the Netherlands) on March 18, 1977. In 1995, he completed grammar school (VWO/Gymnasium) at the Scholengemeenschap Jerusalem in Venray. From 1995 till 2001, he studied Computer Science at the University of Twente. As part of this program, he spent 3 months at British Telecom Laboratories in Ipswich (United Kingdom). His M.Sc. project at the Telematics Systems and Services research group was about 'Prototyping the DiffServ MIB'. He then started doing research at the Architecture of Distributed Systems group (later: Architecture and Services of Network Applications) at the University of Twente. Currently, he is working part-time as a post-doc with the Design and Analysis of Communication Systems group at the University of Twente, and in research & development at Virtu Secure Webservices.

A list of his publications in reverse chronological order:

- Remco van de Meent, Michel Mandjes and Aiko Pras. Gaussian traffic everywhere? In *Proceedings of the 2006 IEEE International Conference on Communications (ICC 2006)*, Istanbul, Turkey, June 11-15, 2006.
- Hans van den Berg, Michel Mandjes, Remco van de Meent, Aiko Pras, Frank Roijers, and Pieter Venemans. QoS-aware bandwidth provisioning of IP links. *Computer Networks*, 50(5):631-647, April 2006.
- Aiko Pras, Remco van de Meent, and Michel Mandjes. QoS in hybrid networks - an operator's perspective. In *Proceedings of IWQoS 2005: 13th International Workshop on Quality of Service* (Hermann de Meer, Nina Bhatti, eds.), no 3552 in Lecture Notes in Computer Science (LNCS), pp. 388-391, Passau, Germany, June 21-23, 2005.
- Georg-Hendrik K. Haan and Remco van de Meent. Towards detection of portscans using IP header data. In *Proceedings of the 11th Open European*

Summer School and IFIP WG 6.6, WG 6.4 and WG 6.9 Workshop (EUNICE 2005). (C.D. Kloos, et al., eds.), pp. 275-282, Madrid, Spain, July 6-8 2005.

– Matthijs Bomhoff, Casper Joost Eyckelhof, Remco van de Meent, and Aiko Pras. Quarantine Net: design and application. *In Application session proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM2005)*, 16-18 May 2005, Nice, France.

– Michel Mandjes and Remco van de Meent. Inferring traffic burstiness by sampling the buffer occupancy. In *Proceedings of NETWORKING 2005: 4th International IFIP-TC6 Networking Conference* (Raouf Boutaba, Kevin Almeroth, Ramon Puigjaner, et al., eds.), no 3462 in Lecture Notes in Computer Science (LNCS), pp. 303-315, Waterloo, Canada, May 2-6, 2005.

– Remco van de Meent and Michel Mandjes. Evaluation of 'user-oriented' and 'black-box' traffic models for link provisioning. In *Proceedings of the 1st EuroNGI Conference on Next Generation Internet Networks Traffic Engineering*, IEEE, Rome, Italy, April 2005.

– Aiko Pras and Remco van de Meent. Meten van internetverkeer voorkomt veel virusleed. In *Automatiseringsgids #37*, September 2004.

– Aiko Pras, Thomas Drevers, Remco van de Meent and Dick Quartel. Comparing the performance of SNMP and Web services based management. *IEEE eTNSM (Transactions on Network and Service Management)*, Vol.1 No.2, December 2004, 11 pages.

– Remco van de Meent and Aiko Pras. Assessing Unknown Network Traffic. *CTIT Technical Report 04-11*, University of Twente, the Netherlands, February 2004.

– Thomas Drevers, Remco van de Meent, and Aiko Pras. Prototyping web services based network monitoring. In *Proceedings of 10th Open European Summer School and IFIP WG 6.3 Workshop (EUNICE 2004)*. (J. Harju, D. Moltchanov and B. Silverejan, eds.), pp. 135-142, June 2004.

– Remco van de Meent, Aiko Pras, Michel Mandjes, Hans van den Berg, Frank Roijers, Pieter Venemans, and Lambert Nieuwenhuis. Burstiness predictions based on rough network traffic measurements. In *Proceedings of the 19th World Telecommunications Congress (WTC/ISS 2004)*, September 2004.

– Remco van de Meent, Aiko Pras, Michel Mandjes, Hans van den Berg, and Lambert Nieuwenhuis, Traffic Measurements for Link Dimensioning: A Case Study. In *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM2003)* (M. Brunner and A. Keller, eds.), no. 2867 in Lecture Notes in Computer Science (LNCS), pp. 106-117, October 2003. Also appeared as CTIT Technical Report 03-36, University of Twente, the Netherlands.

– Hendrik E. Holland, Remco van de Meent, and Aiko Pras. Evaluating MIB II (RFC1213) Implementations. In *The Simple Times*, Vol. 10, No. 1, December 2002.

– Remco Poortinga, Remco van de Meent, and Aiko Pras. Analysing campus traffic using the meter-MIB. In *Proceedings of Passive and Active Measurement Workshop 2002*, March 2002.