Light-Weight Traffic Parameter Estimation for On-Line Bandwidth Provisioning

Wolfram Lautenschlaeger
Bell Labs
Alcatel-Lucent Deutschland AG
Stuttgart, Germany
wolfram.lautenschlaeger@alcatel-lucent.com

Frank Feller
Institute of Communication Networks and Computer Engineering
University of Stuttgart, Germany
frank.feller@ikr.uni-stuttgart.de

***Abstract*—Future network components will power down unused resources to save energy. Thereto, they need to determine the required capacity by observing the traffic. In this paper, we propose a light-weight estimator for the relevant parameters of aggregated packet traffic. The estimator assumes an M/G/∞ traffic model on flow level, which has been proposed for aggregated Internet traffic. We find that the variation of the aggregate traffic rate is defined by the bit rate of contributing *application streams*, i.e. traffic bursts triggered by end-user applications. We identify the effect of these streams on the variance-time behavior of the aggregated traffic rate. From this, we derive an estimator for the application stream bit rate based on second-order statistics of the aggregate rate. Simulation results for inelastic and TCP traffic show a good stream rate estimation accuracy, provided that the measuring period is sufficient to capture the variance of the aggregate rate.**

I. Introduction

Bandwidth provisioning is the process of dimensioning transmission resources for (aggregated) packet traffic to assure a certain quality of service (QoS). Essentially, it means finding an appropriate overdimensioning factor $\alpha > 1$ to obtain the required resource capacity $C = \alpha \cdot m$, where m is the mean traffic load. Obviously, this factor should depend on traffic variations. Due to a lack of knowledge of traffic properties, network operators often choose α according to rules of thumb.

To date, bandwidth provisioning is part of the network planning process. The chosen capacity has to accommodate traffic for months or years to come. Changes in the traffic characteristics (beyond the growth of the traffic volume) are hardly predictable. Coarse dimensioning rules may thus suffice. In the future, bandwidth provisioning will additionally be required on much shorter time scales. To save energy, network devices will activate resources on demand. Due to technological limitations, this cannot always happen instantaneously, but e.g. in the order of 15 minutes to one hour for optical resources. Hence, means to derive the required capacity from on-line traffic measurements are required.

A vast body of literature on bandwidth provisioning exists. Initial work on effective bandwidth defined bandwidth requirements for sources of variable-rate traffic [1][2]. These bandwidth values are added up upon traffic aggregation. Other approaches account for improved statistical multiplexing with increasing aggregation. For the aggregation of traffic from rate-limited sources, M/G/∞ queuing models on flow level have been proposed and studied, e.g. in [3][4][5]. Alternatively, authors of e.g. [6][7] used a Gaussian approximation for the bit rate distribution of highly aggregated traffic. Both of these approaches allow modeling the fractal behavior [8], i.e. long range dependence (LRD) and self similarity, observed in measurements of local area network and Internet traffic. For this purpose, the Gaussian model of fractal Brownian motion and M/G/∞ queues with heavy-tailed service time distributions are used. Addie *et al.* [3] discuss the versatility of M/G/∞ models for a wide range of traffic properties.

These models have given way to two classes of studies: (i) detailed analyses of the buffering performance, and (ii) coarser *buffer-less* flow-level considerations quantifying congestion, i.e. events of more traffic arriving than a resource can serve. Ben Fredj *et al.* [4] discuss both levels. On-line bandwidth provisioning requires a light-weight approach to traffic characterization by measurement without extensive parameter fitting. Since it generally requires less information, the buffer-less view is better suited. Van den Berg, Pras, *et al.* [7][9] propose an estimator for the parameters of a Gaussian flow-level model, which are mean and variance of the aggregate traffic rate.

One limitation of the statistical traffic models (both Gaussian and M/G/∞) is that they assume stochastically stationary conditions. If these are not given, e.g. when deterministic trends like day profiles dominate traffic behavior, we may still apply the models to approximately stationary sections. Such conditions however complicate the estimation of higher-order moments like the variance, which requires an observation period of a certain length. It remains feasible in the busy hour [4], but may be impractical during the transients before and after. A metric expressing traffic variation which is invariant to the traffic load is therefore highly desirable. The M/G/∞ model allows defining such a metric: the bit rate of the contributing traffic streams.

In this paper, we propose an estimator for flow-level parameters of the M/G/∞ model: the mean load (which is meas-

The work presented in this paper was partly funded within the Integrated Project STRONGEST by the European Commission through the 7th ICT-Framework Programme under grant agreement No. 247674 and within the 100GET project by the German Bundesministerium für Bildung und Forschung under contract No. 01BP0768 and 01BP0720.

urable in relatively short time intervals) and the stream bit rate, which is derived form the measured variance. From the M/G/∞ model parameters, one can directly derive a dimensioning rule limiting the packet loss probability for inelastic traffic [10]. We suggest such a rule rather than an approach limiting the duration of congestion periods as proposed in [7], which may better capture user-perceived QoS impairments but requires a much more detailed traffic model.

Our contribution is threefold: We firstly extend the current understanding of the variance-time behavior of M/G/∞ traffic and identify a region solely depending on the rate of contributing streams. Secondly, we propose an M/G/∞ parameter estimator making use of this finding. We thirdly evaluate the performance of this estimator by simulation with access-limited sources of inelastic and TCP traffic.

The remainder of this paper is structured as follows: In section II, we present our model for aggregated Internet traffic in general terms. Section III introduces some restrictions which allow mapping it to a flow-level M/G/∞ model and addresses congestion on flow level. In section IV, we extend the M/G/∞ model to the packet level, discuss the variance-time behavior of aggregated Internet traffic, and identify the impact of stream bit rates. Section V presents the parameter estimator, which is evaluated in section VI. We conclude in section VII.

II. TRAFFIC MODEL

A. Application Stream Concept

The basic item of our traffic model is a single transfer of a finite amount of data. The demand for such a transfer arises typically outside the network with no regard to network conditions. The data transfer is accomplished by an application stream, i.e. a series of symbols that is formed by an end-user application, with dedicated starting time, a certain bit rate, and an according duration. The bit rate of an application stream is determined by the minimum of link capacity, application or protocol limitations, end-system or server performance, or by network congestion. Hence, in any case, the bit rate of an application stream is limited from above.

An application stream is what results from one event in the end-system or one action by the user. In the simplest case, it is a single file transfer or a voice call, but it may also comprise several TCP connections used by a web browser to retrieve the parts of one web page. The download of a further web page at a later instant of time reusing the same TCP connection constitutes a different application stream.

In the network, the application streams of many independent users superimpose to larger aggregated traffic flows. Since arrival and termination of application streams is random, the actual number of contributing streams, and hence the cumulative bit rate are random, too. Congestion occurs if at a given transmission link the randomly fluctuating aggregated traffic exceeds the link capacity. It is a primary goal of network dimensioning to make congestion a rare event. Since the mixture of contributing applications and protocols is unknown, this is the only way to limit traffic impairments (admission blocking, packet loss, delay, throughput degradation). This design objective is also called *link transparency* in the literature [4][7].

B. Packet Streams and Multiplexing

In packet networks, application streams are subdivided into packets of limited size (which is today, due to the predominating Ethernet infrastructure, generally ≈1500 Bytes at the maximum). Correspondingly, the application stream bit rates translate into packet distances. If e.g. an end-user's access link is the bottleneck, packets fly back-to-back there. On faster aggregation links, however, packet durations are reduced, while packet distances, and thus stream durations, remain basically unchanged. Then, the multiplexing of several application streams into one larger aggregated traffic flow is essentially the mutual interlacing of the contained packets as illustrated in Fig. 1. The acceleration from a comparably small application stream bit rate b to the larger transport link capacity C creates the necessary inter-packet spaces for interlacing. Randomly occurring collisions between simultaneously arriving packets from different application streams are resolved by comparably small buffers.

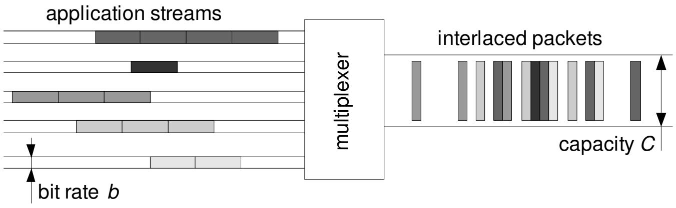

Fig. 1: Packet multiplexing, application streams are chopped into packets and interlaced at higher bit rates

If, however, at a given moment the cumulative bit rate of the concurrent application streams is too high, we have congestion that cannot reasonably be resolved by buffering. It would require very large buffers with holding times beyond any reasonable size.

III. STREAM LEVEL ANALYSIS

To estimate the congestion probability, we use a flow-level model, i.e. we consider the aggregated traffic as an overlay of application streams and disregard the packet structure.

We assume a high number of independent sources generating application streams, which we approximate by an infinite source model with negative-exponentially distributed interarrival times. (Here any other distribution would collide with the independence assumption [11].)

The application streams are assumed of uniform bit rate b. We derive this from the worst case assumption (in terms of traffic volatility) that each application stream loads its access link at the capacity limit. In addition, we assume that all end users are connected to the network by access links of identical capacities. These restrictions will partly be relaxed later on.

We make only mild assumptions on the size of the application streams. For analytical considerations, we allow any (i.e.

a general) distribution, but we require the majority of the carried traffic volume to be constituted of streams comprising a significant number of packets. (In practice we are talking here about at least 10 to 100 packets.)

Since we are interested in the offered traffic without network feedback (i.e. the superposition of all streams without interference), we can describe the number of active streams by an infinite server queuing model. With the above assumptions, it is an M/G/∞ model, and its probabilities of state follow a Poisson distribution [11]. With the average number λ of simultaneously active application streams, the probability P_k to see k active application streams at any given moment in time is

$$P_k = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (1)$$

The dimensionless stream counts can be translated into bit rates as follows:

$$m = \lambda \cdot b \quad (2)$$

$$\xi = X \cdot b \quad (3)$$

where m is the mean bit rate of aggregated traffic, b the bit rate of a single application stream, X the random stream number, and ξ the random bit rate at any given moment in time.

The probability distribution P_k is infinite, but the capability to carry traffic is always finite, so the distribution is truncated by different loss mechanisms. In case of admission blocking, which rejects whole streams if the capacity is exhausted, the loss is given by Erlang's B-formula [11]. In case of packet streams without admission control, the overshooting fraction of packets is dropped, irrespective of the particular stream they belong to. Nevertheless, the drop probability is only slightly less than the blocking in case of admission control, as has been shown in [10][12] both numerically and in experiment. In case of really large buffers, parts of the overshooting application streams could be held in buffer instead of dropping them. However, the volume of the application streams queued during a congestion period would create an undesired spike of packet latency. Finally, in case of TCP traffic, the congestion control algorithm in the end-systems would lower the sending rates during a congestion period, which manifests itself to end-users as throughput degradation.

Hence, the requirement of low congestion translates into keeping the truncated quantile of (1) small. This minimizes traffic impairments (loss, latency, throughput) irrespective of the actual stream size distribution, (large) buffer sizes, or correct TCP usage in the end systems.

IV. Packet Level Analysis

The analysis at stream level (section III) assumes knowledge of stream shaping factors, like the limitation by access links. For aggregated traffic, such knowledge may not be available. While flow accounting mechanisms (NetFlow [13], IPFIX [14]) can provide some insight into the composition of a traffic aggregate (in terms of TCP or UDP flows), they fall short of identifying the statistically relevant parameters such as rates of application streams according to our definition in section II. In addition, we are interested in a light-weight provisioning mechanism which scales to very high bit rates and numbers of streams. We therefore investigate in the following how to obtain the parameters of equations (1) to (3) from solely observing packet arrivals and sizes in aggregated traffic.

For simplicity, we assume a uniform packet size s (in number of bits) and uniform application stream bit rates b for all participating sources. Furthermore, we assume that application streams have a mean total size of S (in number of bits). We denote the forwarding capacity of the link by C (bit per second), and we assume that this link is loaded by a cumulative traffic of m (bit per second) in average.

In the following, we determine whether stream shaping factors influence the mean and variance of the aggregate traffic rate at different time scales. We identify four ranges of different behavior separated by: T_1, the packet forwarding duration on the aggregation link, T_2, the packet distance in a particular application stream, and T_3, the application stream duration.

$$T_1 = \frac{s}{C}, \qquad T_2 = \frac{s}{b}, \qquad T_3 = \frac{S}{b} \quad (4)$$

All calculations are rough estimates in that they disregard the transitions between the ranges.

A. Time Scale $T < T_1$

We start with intervals below packet duration, i.e. $T < T_1$. At this time scale we see a real on/off source since consecutive packets cannot overlap. The random load variable ξ is either C (packet in flight), or zero. The probability P_{pkt} of packet in flight depends on the mean load.

$$\xi \in \{0, C\} \quad (5)$$

$$P_{pkt} = \Pr(\xi = C) = \frac{m}{C} = r \quad (6)$$

For mean and variance of this distribution, we obtain

$$\mu_I = C \cdot P_{pkt} = m$$
$$v_I = C^2 \cdot P_{pkt} - \mu_I^2 = C^2 \cdot r(1 - r) = m(C - m) \quad (7)$$

Obviously, the variance of the instantaneous traffic rate does not reflect the presence of application streams.

B. Time Scale $T_1 < T < T_2$

The next time scale is larger than the packet duration but smaller than the packet distance in an application stream, i.e. $T_1 < T < T_2$. In this case the random load variable ξ is not defined by just one packet but by the cumulative load of all packets that fall into the interval T. With X_P, the random number of such packets, we obtain

$$\xi = \frac{s \cdot X_P}{T} \quad (8)$$

Since the interval T is smaller than the packet distance in an application stream, all packets we see within T belong to different application streams. Their arrivals are mutually independent since we assume independence of the streams. The number of independent random events in an interval is known to be Poisson distributed. X_P thus follows a Poisson distribution with λ_P, the mean number of packets in interval T:

$$\lambda_P = \frac{m}{s} T \qquad (9)$$

With the mean and variance of the Poisson distributed X_P (both equal λ_P) and (9), we get for the random traffic ξ:

$$\mu_P = \frac{s}{T} \lambda_P = m$$
$$\nu_P = \left(\frac{s}{T}\right)^2 \cdot \lambda_P = \frac{s}{T} m \qquad (10)$$

The variance declines reciprocally with increasing interval T, as expected for Poisson traffic. The presence of application streams is still not reflected in the variance of the traffic.

C. *Time Scale $T_2 < T < T_3$*

This does change on the third time scale, which is larger than the packet distance in application streams, but smaller than the mean application stream duration, i.e. $T_2 < T < T_3$. As explained in section III, the number X of simultaneously active application streams is a Poisson distributed random number of mean λ (the mean number of concurrent application streams, recall (2), $\lambda = m / b$).

If we disregard the case that an application stream starts or ends in interval T, one particular stream contributes n packets to the traffic in this interval:

$$n = \left\lfloor \frac{b}{s} T \right\rfloor \qquad (11)$$

Correspondingly the random load variable ξ in interval T is:

$$\xi = \frac{1}{T} n \cdot s \cdot X = \frac{1}{T} \left\lfloor \frac{b}{s} T \right\rfloor s \cdot X \qquad (12)$$

For a rough estimation we can ignore the quantization by the integer operator and get

$$\xi \cong b \cdot X \qquad (13)$$

which conforms to the rather high level consideration of (3).

With the knowledge of mean and variance of the Poisson distributed X we get:

$$\mu = b\lambda = m$$
$$\nu = b^2 \lambda = b \cdot m \qquad (14)$$

Obviously, at given load the variance is proportional to the application stream bit rate b. Even more important, the variance does not decline anymore with increasing interval T.

The variance-to-mean ratio (VMR) at this timescale equals the application stream bit rate, and we can use the VMR to measure this bit rate in unknown aggregated traffic:

$$\frac{\nu}{\mu} = b \qquad (15)$$

We propose this value as a new parameter for the quantification of traffic fluctuations. It is easily measurable, and it can alternatively be derived from the installed base of access links. To a certain degree, it is orthogonal to the traffic load. This property is in line with the load-independent *peakedness* of Poisson traffic well known in teletraffic theory [11].

D. *Time Scale $T > T_3$*

Finally, the time scale larger than application stream durations, $T > T_3$: Temporal effects in this range are irrelevant for the buffering performance in practice, since according buffer sizes would imply delays of minutes and more. We only give an estimation for the marginal case of constant stream size S. In this case, whole application streams fall into the interval T. The application streams are independent of each other, hence the load variable X_S is again Poisson distributed, but this time for whole application streams instead of single packets. Neglecting that flows may only partially fall into T, we get in analogy to (8) – (11):

$$\xi = \frac{S \cdot X_S}{T} \qquad (16)$$

$$\lambda_S = \frac{m}{S} T \qquad (17)$$

The resulting mean and variance are:

$$\mu_S = \frac{S}{T} \lambda_S = m$$
$$\nu_S = \left(\frac{S}{T}\right)^2 \cdot \lambda_S = \frac{S}{T} m \qquad (18)$$

For constant-size streams, the variance returns to the reciprocal decline with increasing duration at large intervals T.

The calculations of this section tend to under-estimate the variance observed in practice. The stream size distribution is often open ended, in contrast to the packet size bounded above (e.g. at 1500 byte). At any interval setting T, there remain some large application streams that still spread over several intervals. In consequence, the transition into this section ($T > T_3$) is never complete and the variance decline remains slower than Poisson. Simulation results in Fig. 2 illustrate the effect, in particular for heavy-tailed, Pareto distributed stream sizes. Asymptotic LRD and self-similarity manifest themselves in this range. Nonetheless, the marginal distribution of the number of concurrently active streams remains Poisson.

E. *Interpretation*

Table 1 summarizes the results for the four ranges of different behavior of aggregated packet traffic. The variance-to-mean ratio (VMR) in the last row deserves special attention. Except for the instantaneous traffic, it is independent of the actual traffic load m, which makes it preferable for the characterization of aggregated traffic. In particular in intervals larger than the packet distance in streams (T_2) but smaller than the application stream duration (T_3), it reflects the application stream bit rate b without any further addition.

Fig. 2 shows the asymptotic trend lines in the different sections together with exemplary VMR plots obtained by simula-

TABLE 1: Variance of aggregated packet traffic at different time scales

	$T<T_1$	$T_1<T<T_2$	$T_2<T<T_3$	$T_3<T$
variance v	$m\cdot(C-m)$	$m\cdot s/T$	$m\cdot b$	$\geq m\cdot S/T$
VMR v/μ	$C-m$	s/T	b	$\geq S/T$

tions (see section VI for the setup).

Obviously, the reciprocal variance decline with increasing interval durations stops at T_2 and resumes only beyond T_3. The existence of the VMR plateau at the level of the application stream bit rate b has direct impact on buffer performance. Buffer holding times beyond T_2 contribute to latency and jitter, but cannot significantly reduce packet loss.

We verified the effect with different stream size distributions: constant, negative-exponential, and Pareto (heavy-tailed), the latter being inspired by the observation of heavy-tailed file size distributions in [15]. The effect is present regardless of the stream size distribution, i.e. it is unrelated to asymptotic LRD. Nevertheless, at large time intervals ($T > T_3$), the Pareto distributed traffic shows the typical asymptotic LRD behavior with a much slower variance decline than under the Poisson assumption.

In practice, the transition zones between the different ranges may overlap so that the stepped trend is not as obvious as in Fig. 2. In particular, if the acceleration factor from application stream bit rate b to core link capacity C is small, then the bends at T_1 and T_2 may merge into one rather unspecific decline. Furthermore, the bends at T_2 and T_3 may merge if the number of packets per application stream is low. The bend at T_3 depends on the distribution of application stream durations and may in particular be blurred by heavy-tailed stream size distributions. Finally, the application stream bit rate may not be as uniform as assumed. Besides, the TCP transmit window mechanism is known to form chunks of packets at higher than average rate, which could cause additional bends in the stepped trend. All those effects together create the repeatedly reported multi-fractal behavior of Internet traffic [8].

Despite all uncertainties with respect to the real extent of

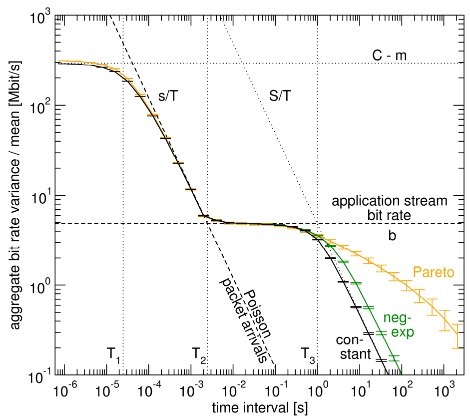

Fig. 2: Principle trend of the variance-to-mean ratio with increasing observation interval for different application stream size distributions

the particular effects, there is one reliable traffic shaping effect – the access link capacity. Whatever happens in the application or the protocol stack, a terminal cannot form (or receive) packet streams faster than the access link capacity. In this way, the large acceleration factors C/b in the today's Internet create a first dominating step as of Fig. 2 without further multi-fractal structuring below T_2. This could explain the seeming recurrence of Poisson behavior in highly aggregated traffic [6].

V. Measurement of the Application Stream Bit Rate

In order to apply dimensioning rules to unspecified traffic, we need to estimate the parameters of equations (1) to (3). The mean traffic load m can easily be assessed by reading out traffic counters. The results of Table 1 enable us to estimate the application stream bit rate b without further a-priori knowledge of the traffic origin. According to (15), it is sufficient to estimate the variance-to-mean ratio in intervals T that are larger than the packet distance in a particular application stream and smaller than typical application stream durations. In practice we choose here, as a compromise, an interval in the range of 25 ms. We then estimate the first and second raw statistical moments by temporal averages of the traffic value and its square.

$$E[\xi] \cong \frac{1}{N}\sum_i X_i = m_1 \tag{19}$$

$$E[\xi^2] \cong \frac{1}{N}\sum_i X_i^2 = m_2 \tag{20}$$

The temporal averages are good enough as soon as the averaging period is much larger than the duration of the majority of application streams. In practice, we use averaging periods in the range of 5 minutes up to 1 hour and more. We obtain the requested application stream bit rate b as:

$$b = \frac{m_2 - m_1^2}{m_1} \tag{21}$$

Fig. 3 shows a block diagram of a corresponding sliding estimator of the effective application stream rate b in an undeclared traffic aggregate. We obtain the random traffic value X_i by periodic read out of the byte counter. The traffic value and its square are fed into low pass filters that accordingly estimate the first and second raw moment of X_i. Finally, the application stream bit rate is calculated according to (21). Since the

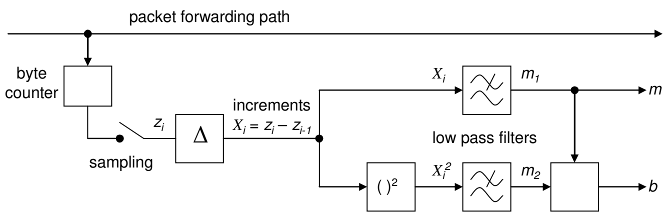

Fig. 3: Block diagram of a sliding estimator of the effective application stream bit rate in undeclared aggregated traffic

estimator is essentially a chain of low pass filters, the processing can go along with a cascade of down-sampling operations. Only the byte counter has to operate at the line rate of the corresponding packet forwarding path.

VI. EVALUATION

We evaluated the performance of the application stream bit rate estimator by means of event-driven simulation based on the IKR SimLib [16].

A. *Simulation Scenario*

Fig. 4 depicts the simulation scenario. The sources emit application streams with negative-exponentially distributed interarrival times at a rate of 40 / s. Each source is connected to the multiplexer by a dedicated access link. The access links have a bit rate of b = 5 Mbit/s and feature access buffers dimensioned following the bandwidth-delay product (BDP) rule [17] (which enables full link utilization by individual TCP connections). The capacity C of the link carrying the aggregated traffic varies from 210 Mbit/s to 500 Mbit/s, with resulting link load from 40% to 95%. The aggregation buffer preceding this link is dimensioned according to the *small buffer* rule [17] to BDP/$\sqrt{N}$, where $N = 40$ is the number of concurrently active (TCP) streams. A demultiplexer distributes the traffic streams to the sinks. The figure omits the backward channels for TCP signaling. Emulating wide area network conditions, the base round-trip time (RTT) between sources and sinks is 100 ms.

Application stream rate estimation is done after the aggregation link and optionally before the aggregation buffer. We set the sampling interval to 25 ms, which covers 10 packets of an access-limited application stream. The averaging period varies from seconds to more than one hour. As baseline, we additionally determine the actual stream bit rates on the access links.

We use two source configurations: (i) inelastic traffic sources, which send application streams as streams of equidistant packets at the rate of the access links; and (ii) TCP sources, which transfer application streams in TCP connections using the Cubic congestion control scheme. For the latter, we embed the actual Linux protocol stack implementation in the simulation by means of the Network Simulation Cradle (NSC) [18].

Application stream sizes follow either a negative-exponential or a Pareto distribution with parameter $\alpha = 1.2$ (exemplarily for heavy-tailed distributions). In both cases, the mean stream size is 595 KB, which corresponds to a stream duration of 1 s in case of inelastic traffic (accounting for Ethernet overhead).

B. *Temporal Behavior*

We illustrate the temporal performance of the application stream rate estimator for inelastic, negative-exponentially sized streams in the following scenario: During the first 1000 s, the network is dimensioned as outlined above. Then, the access bit rate is increased to the ten-fold (50 Mbit/s) for 1500 s, and finally returns to the initial value. The stream arrival pattern and sizes meanwhile remain unchanged. That is, the same user population is temporarily provided with higher access speeds. Fig. 5 plots the instantaneous aggregate bit rate on an unlimited aggregation link. It clearly shows the increase of the traffic variation between 1000 s and 2500 s. Fig. 6 gives the trace of the stream bit rate estimate, which reliably follows the changes with a delay in the order of the averaging period of 250 s. Note that the abrupt variation of access bit rates is an artificial assumption and the detection delay thus of no practical relevance. The slight under-estimation of the 50 Mbit/s stream rates is attributable to the small number of samples each application stream is present in.

C. *Mean Estimate for Uncongested Link*

Fig. 7 plots the mean of the stream bit rate estimate obtained for an uncongested aggregation link over the averaging period (along with 95% confidence intervals). All curves show the trend to converge to the actual stream bit rates for increasing averaging periods. For negative-exponentially distributed stream sizes, averaging periods of five minutes (300 s) and more provide accurate estimates. In case of a heavy-tailed stream size distribution, periods of more than one hour are necessary to achieve a comparable performance. This traces back the slower decrease of the autocorrelation function of the aggregate bit rate in the latter case, which increases the time required to obtain a sufficient number of independent samples for variance estimation. This handicap is inherent to any

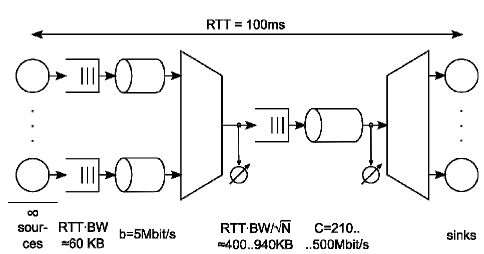

Fig. 4: Simulation scenario

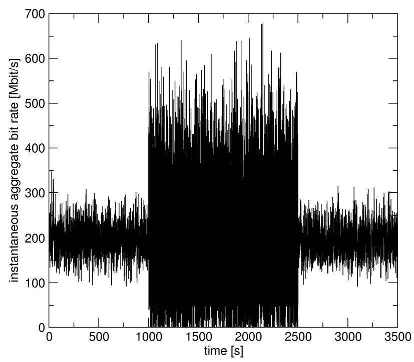

Fig. 5: Trace of instantaneous aggregate traffic rate

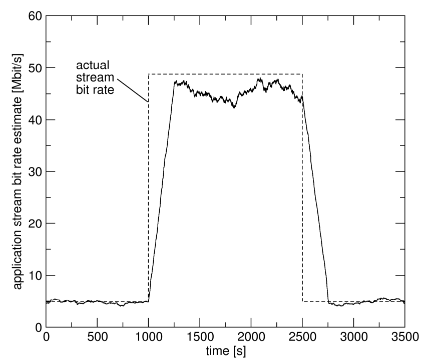

Fig.6: Trace of stream bit rate estimate

purely statistical traffic characterization approach.

The application stream bit rate estimates for inelastic traffic and for TCP traffic essentially follow the same trend, but show an offset which only depends on the stream size distribution. This offset is caused by TCP's slow start mechanism, which prevents TCP streams from fully utilizing the access link at the start of their transmission. Consequently, TCP streams achieve a lower mean bit rate, as indicated by the separate asymptote for TCP traffic in Fig. 7. This effect is more pronounced for the heavy-tailed stream size distribution, which produces a higher share of small streams (since it needs to compensate its characteristic huge streams to yield the same mean). The equilibrating effect of the aggregation buffer explains the small difference between the estimates for TCP traffic obtained before and after this buffer.

From these observations, we can conclude that the estimator accurately detects the application stream bit rate of both inelastic and TCP traffic in the considered scenario – provided that the averaging period is sufficiently large. In further investigations, which we cannot detail here for space restrictions, we found a comparable estimator performance for a wide range of application stream bit rates and aggregation levels. If confronted with an aggregate of application streams of different bit rates, the estimation yields the average stream rate weighted by the contribution of the streams to the traffic volume.

D. *Impact of Congestion*

Any mechanism trying to estimate traffic source properties based on traffic observation in a network suffers from the limitation that its input data may be distorted due to insufficient capacity elsewhere in the network. Fig. 8 illustrates such an effect on the application stream bit rate estimation for negative-exponentially distributed stream sizes and an averaging period of 250 s. We gradually reduce the aggregation link capacity to obtain higher occupancies (and more frequent congestion) for the same input traffic. All curves show significant under-estimation for link occupancies exceeding 75%. Since the capacity limit removes peaks of the aggregate rate, it reduces the variance of this signal and, in turn, lowers the stream rate estimates. The aggregation buffer amplifies this effect by further smoothing the aggregate rate.

While this explanation applies to inelastic traffic, it is insufficient for the effect on TCP traffic. In particular, the measurement upstream of the aggregation link cannot be directly affected by this link. The TCP sources rather proactively remove the peaks from the traffic aggregate as they rapidly react to congestion signals. Since this mechanism tends to reduce TCP stream rates for periods exceeding the instants of congestion, we observe lower stream rate estimates than for inelastic traffic. The difference between the estimates for TCP traffic before and after the aggregation link is due to buffering effects and occasional losses at the aggregation buffer.

E. *Bandwidth Provisioning*

For on-line bandwidth provisioning, we derive the capacity required to respect a given loss probability limit from the stream rate estimate and the measured mean load as described in [10]. It is crucial that the capacity estimate is reasonably insensitive to congestion, which may arise when the bandwidth demand grows. We therefore investigate the effect of congestion-related stream rate underestimation on the capacity estimate.

Fig. 9 plots the estimate of the bandwidth required for a packet loss probability of 10^{-3} (for inelastic traffic) over the link occupancy. We derived these curves from the stream rate estimates given in Fig. 8. As baseline, the graph additionally shows the bandwidth requirement computed from the parameters of the simulation scenario. The dotted curve finally indicates the actual aggregation link capacity provoking the respective occupancy and congestion situation.

The intersection of the curves of the required and the actual capacity at an occupancy of 67% indicates the target operation point. At lower loads, we can deactivate unused resources. At higher loads, we need to provide more capacity. For occupancies up to 75%, the capacity estimate is accurate and we can reach the target operation point with the next adaptation. Above 75%, the capacity estimate is too low, but still indicates that we need more resources. We will thus return to the target operation point after several adaptations. Since occupancies of

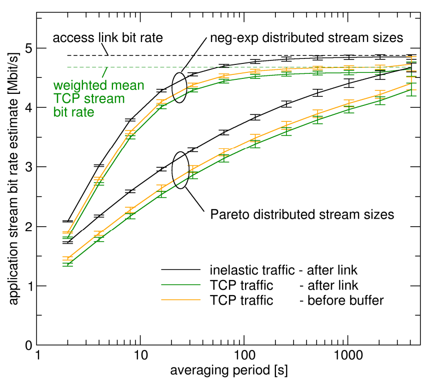

Fig. 7: Mean estimate over averaging period for uncongested aggregation link

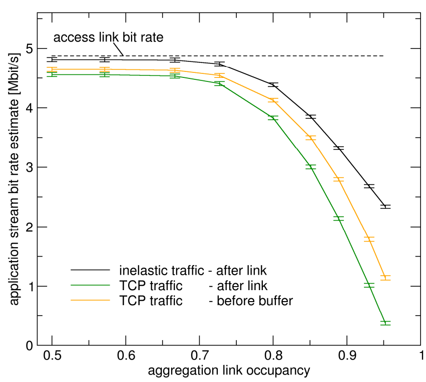

Fig. 8: Mean estimate over aggregate link occupancy for neg.-exp. stream sizes and averaging period 250s

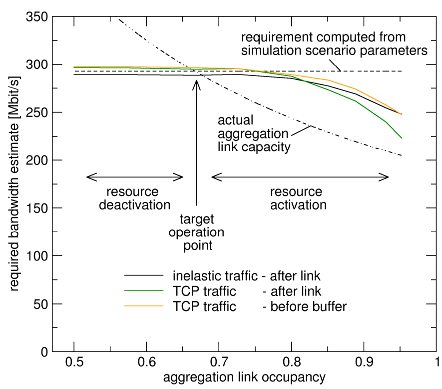

Fig. 9: Estimate of bandwidth required for packet-loss probability of 10^{-3} over link occupancy

more than 75% imply a packet loss probability of at least 10^{-2}, this condition should rarely occur in practice. We observed these effects under a variety of traffic configurations.

These observations give way to three important statements on bandwidth provisioning: Firstly, the capacity estimate remains accurate under moderate congestion and can thus keep track of a reasonably slow evolution of the bandwidth demand. Secondly, if this condition is not met, the provisioning mechanism proves safe in that it demands the activation of further resources and will thus converge to the appropriate capacity. Finally, if faced with a traffic aggregate that is limited elsewhere in the network, the mechanism will overestimate the required bandwidth. Although it reduces energy efficiency, this behavior is desirable in order to prevent further impairment to such traffic.

VII. Conclusion

In this paper, we lay the grounds for an on-line bandwidth provisioning scheme for aggregated Internet traffic. For this purpose, we first motivated the use of an M/G/∞ queue as a model for aggregated traffic made up of bandwidth-limited application streams. For this model, dimensioning rules exist which target a given level of QoS in terms of packet loss. We then identified a time scale – between the packet distance in application streams and the application stream duration – where the variance of the aggregate traffic rate solely depends on its mean and the contributing application streams' bit rate. We established that this observation explains the poor buffering performance of Internet traffic frequently attributed to asymptotic long-range dependency or self-similarity. We further stated that it is present regardless of the application stream size distribution, which defines the asymptotic properties. Making use of these findings, we proposed an estimator extracting the model parameters required for bandwidth provisioning from easily observable properties of a traffic aggregate. We finally evaluated the performance of this estimator by means of simulation.

Provided a sufficient measurement period, the estimated model parameters proved accurate for both inelastic traffic and TCP traffic regardless of the application stream size distribution. The length of the measurement period needs to allow for correct estimation of the variance of the aggregate traffic rate. For negative-exponentially distributed stream sizes, we require periods of two orders of magnitude above the average stream duration. For heavy-tailed stream size distributions, three to four orders of magnitude are necessary. Since congestion reduces traffic variations, it impairs any mechanism relying on the variance of a traffic aggregate – including our estimator. For on-line bandwidth provisioning, however, this limitation is uncritical under relevant network operation conditions.

Future work is required to validate our model assumptions and the estimator with actual Internet traffic. Most freely available traffic traces are unsuitable for this purpose, since they have been collected in local area networks. The validation requires Internet traffic that undergoes significant acceleration prior to multiplexing. Aggregation and backbone links of Internet service providers offer such conditions. Another open issue is determining the most suitable sampling interval of the aggregate rate, which needs to range between the packet distance in application streams and the application stream duration. While these bounds are unknown in the general case, one can often base the choice of the sampling interval on reasonable assumptions. However, a self-tuning mechanism would likely increase the accuracy of the estimator.

Acknowledgment

The authors would like to thank Joachim Scharf for valuable discussions and feedback.

References

[1] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 7, pp. 968–981, 1991.

[2] F. P. Kelly, "Notes on effective bandwidths," in *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary, and I. Zeidins, Eds. Oxford University Press, 1996, pp. 141–168.

[3] R. G. Addie, T. D. Neame, and M. Zukerman, "Performance evaluation of a queue fed by a Poisson Pareto burst process," *Computer Networks*, vol. 40, no. 3, pp. 377 – 397, 2002.

[4] S. B. Fredj *et al.* , "Statistical bandwidth sharing: a study of congestion at flow level," in *Proc. SIGCOMM*. New York, NY, USA, 2001.

[5] T. Bonald and J. Roberts, "Congestion at flow level and the impact of user behaviour," *Computer Networks*, vol. 42, no. 4, 2003.

[6] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning IP backbone networks to support latency sensitive traffic," in *Proc. INFOCOM*, vol. 1, 2003.

[7] H. van den Berg *et al.*, "QoS-aware bandwidth provisioning for IP network links," *Computer Networks*, vol. 50, no. 5, pp. 631 – 647, 2006.

[8] A. Feldmann *et al.*, "Dynamics of IP traffic: a study of the role of variability and the impact of control," *SIGCOMM Comput. Commun. Rev.*, vol. 29, pp. 301–313, August 1999.

[9] A. Pras *et al.*, "Dimensioning network links: a new look at equivalent bandwidth," *IEEE Netw.*, vol. 23, no. 2, pp. 5–10, 2009.

[10] W. Lautenschlaeger, W. Frohberg, "Bandwidth dimensioning in packet-based aggregation networks," in *Proc. 13th Int. Telecommunications Network Strategy and Planning Symp. (Networks)*, Budapest, Hungary, 2008.

[11] "Handbook Teletraffic Engineering", ITU-D Study Group 2 Question 16/2, 2008.

[12] W. Lautenschlaeger, "Equivalence conditions of buffered and bufferless network architectures," in *Proc 9th ITG Symp. on Photonic Networks*, Leipzig, Germany, May 2008.

[13] "Netflow services and applications," White Paper, Cisco Systems, 1999.

[14] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information," RFC 5101 (Proposed Standard), Internet Engineering Task Force, Jan. 2008.

[15] K. Park, G. Kim, and M. Crovella, „On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proc. Int. Conf. on Network Protocols*, Columbus, 1996

[16] IKR Simulation Library, IKR, University of Stuttgart. Online: http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/

[17] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *Proc. SIGCOMM*. New York, NY, USA, 2004.

[18] Network Simulation Cradle, Wand Network Research Group, University of Waikato in Hamilton. Online: http://www.wand.net.nz/~stj2/nsc/