

Dimensioning Network Links: A New Look at Equivalent Bandwidth

Aiko Pras and Lambert Nieuwenhuis, University of Twente
Remco van de Meent, Vodafone NL
Michel Mandjes, University of Amsterdam

Abstract

One of the tasks of network management is to dimension the capacity of access and backbone links. Rules of thumb can be used, but they lack rigor and precision, as they fail to reliably predict whether the quality, as agreed on in the service level agreement, is actually provided. To make better predictions, a more sophisticated mathematical setup is needed. The major contribution of this article is that it presents such a setup; in this a pivotal role is played by a simple, yet versatile, formula that gives the minimum amount of capacity needed as a function of the average traffic rate, traffic variance (to be thought of as a measure of “burstiness”), as well as the required performance level. In order to apply the dimensioning formula, accurate estimates of the average traffic rate and traffic variance are needed. As opposed to the average rate, the traffic variance is rather hard to estimate; this is because measurements on small timescales are needed. We present an easily implementable remedy for this problem, in which the traffic variance is inferred from occupancy statistics of the buffer within the switch or router. To validate the resulting dimensioning procedure, we collected hundreds of traces at multiple (representative) locations, estimated for each of the traces the average traffic rate and (using the approach described above) traffic variance, and inserted these in the dimensioning formula. It turns out that the capacity estimate obtained by the procedure, is usually just a few percent off from the (empirically determined) minimally required value.

To ensure that network links are sufficiently provisioned, network managers generally rely on straightforward empirical rules. They base their decisions on rough estimates of the load imposed on the link, relying on tools like MRTG [1], which poll management information base (MIB) variables like those of the *interfaces table* on a regular basis (for practical reasons, often in five-minute intervals). Since the peak load *within* such a measurement interval is in general substantially higher than the average load, one frequently uses rules of thumb like “take the bandwidth as measured with MRTG, and add a safety margin of 30 percent.”

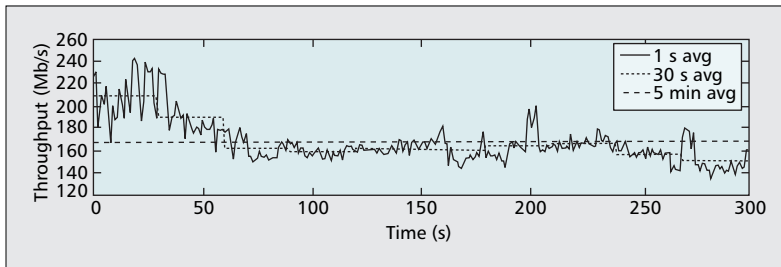
The problem with such an empirical approach is that in general it is not obvious how to choose the right safety margin. Clearly, the safety margin is strongly affected by the performance level to be delivered (i.e., that was agreed on in the service level agreement [SLA]); evidently, the stricter the SLA, the higher the capacity needed on top of the average load. Also, traffic fluctuations play an important role here: the burstier the traffic, the larger the safety margin needed. In other words, the simplistic rule mentioned above fails to incorporate the dependence of the required capacity on the SLA and traffic characteristics. Clearly, it is in the interest of the network manager to avoid inadequate dimensioning. On one hand, *underdimensioning* leads to congested links, and hence inevitably to performance degradation. On the other hand, *overdimensioning* leads to a waste of capacity (and money); for instance, in networks operating under differenti-

ated services (DiffServ), this “wasted” capacity could have been used to serve other service classes.

We further illustrate this problem by examining one of the traces we have captured. Figure 1 shows a five-minute interval of the trace. The 5 min traffic average throughput is around 170 Mb/s. The traffic average throughput of the first 30 s period equals around 210 Mb/s, 30 percent higher than the 5 min average. Some of the 1 s traffic average throughput values go up to 240 Mb/s, more than 40 percent of the 5 min average values. Although not shown in the figure, we even measured 10 ms spikes of more than 300 Mb/s, which is almost twice as much as the 5 min value. Hence, the average traffic throughput strongly depends on the time period over which the average is determined. We therefore conclude that rules of thumb lack general validity and are therefore oversimplistic in that they give inaccurate estimates of the amount of capacity needed.

There is a need for a more generic setup that encompasses the traffic characteristics (e.g., average traffic rate, and some measure for burstiness or traffic variance), the performance level to be achieved, and the required capacity. Qualitatively, it is clear that more capacity is needed if the traffic supply increases (in terms of both rate and burstiness) or the performance requirements are more stringent, but in order to successfully dimension network links, one should have *quantitative* insights into these interrelationships as well.

The goal of this article is to develop a methodology that can be used for determining the capacity needed on Internet



■ Figure 1. Traffic rates at different timescales.

links, given specific performance requirements. Our methodology is based on a dimensioning formula that describes the above-mentioned trade-offs between traffic, performance, and capacity. In our approach the traffic profile is summarized by the average traffic rate and traffic variance (to be thought of as a measure of burstiness). Given predefined performance requirements, we are then in a position to determine the required capacity of the network link by using estimates of the traffic rate and traffic variance.

We argue that particularly the traffic variance is not straightforward to estimate, especially on smaller timescales as mentioned above. We circumvent this problem by relying on an advanced estimation procedure based on occupancy statistics of the buffer within the switch or router so that, importantly, it is *not* necessary to measure traffic at these small timescales. We extensively validated our dimensioning procedure, using hundreds of traffic traces we collected at various locations that differ substantially, in terms of both size and the types of users. For each of the traces we estimated the average traffic rate and traffic variance, using the above mentioned buffer occupancy method. At the same time, we also empirically determined per trace the correct capacities, that is, the minimum capacity needed to satisfy the performance requirements. Our experiments indicate that the determined capacity of the needed Internet link is highly accurate, and usually just a few percent off from the correct value.

The material presented in this article was part of a larger project that culminated in the thesis [2]; in fact, the idea behind this article is to present the main results of that study to a broad audience. Mathematical equations are therefore kept to a minimum. Readers interested in the mathematical background or other details are therefore referred to the thesis [2] and other publications [3, 4].

The structure of this article is as follows. The next section presents the dimensioning formula that yields the capacity needed to provision an Internet link, as a function of the traffic characteristics and the performance level to be achieved. We then discuss how this formula can be used in practice; particular attention is paid to the estimation of the traffic characteristics. To assess the performance of our procedure, we then compare the capacity estimates with the “correct” values, using hundreds of traces.

Dimensioning Formula

An obvious prerequisite for a dimensioning procedure is a precisely defined performance criterion. It is clear that a variety of possible criteria can be chosen, with their specific advantages and disadvantages. We have chosen to use a rather generic performance criterion, to which we refer as *link transparency*. Link transparency is parameterized by two parameters, a time interval T and a fraction ϵ , and is defined as the fraction of (time) intervals of length T in which the offered traffic exceeds the link capacity C should be below ϵ .

The link capacity required under link transparency, say $C(T, \epsilon)$, depends on the parameters T , ϵ , but clearly also on the characteristics of the offered traffic. If we take, for exam-

ple, $\epsilon = 1$ percent and $T = 100$ ms, our criterion says that in no more than 1 percent of time intervals of length 100 ms is the offered load supposed to exceed the link capacity C . T represents the time interval over which the offered load is measured; for interactive applications like Web browsing this interval should be short, say in the range of tens or hundreds of milliseconds up to 1 s. It is intuitively clear that a shorter time interval T and/or a smaller fraction ϵ will

lead to higher required capacity C . We note that the choice of suitable values for T and ϵ is primarily the task of the network operator; he/she should choose a value that suits his/her (business) needs best. It is clear that the specific values evidently depend on the underlying applications, and should reflect the SLAs agreed on with end users.

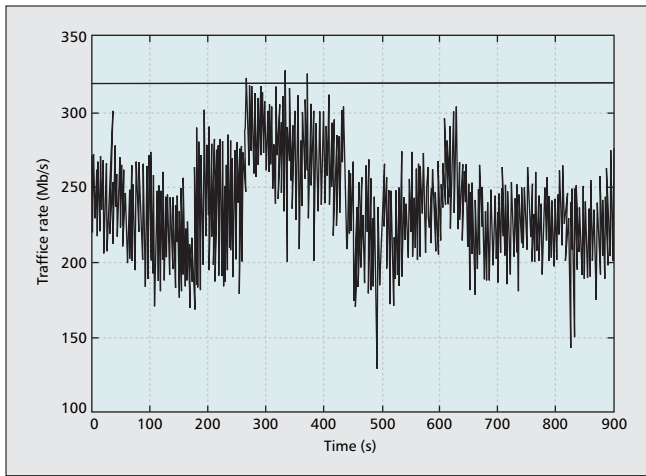
Having introduced our performance criterion, we now proceed with presenting a (quantitative) relation between traffic characteristics, the desired performance level, and the link capacity needed. In earlier papers we have derived (and thoroughly studied) the following formula to estimate the minimum required capacity of an Internet link [2, 3]:

$$C(T, \epsilon) = \frac{1}{T} \sqrt{(-2 \log \epsilon) \cdot v(T)}. \quad (1)$$

This dimensioning formula shows that the required link capacity $C(T, \epsilon)$ can be estimated by adding to the average traffic rate some kind of “safety margin.” Importantly, however, in contrast to equating it to a fixed number, we give an explicit and insightful expression for it: we can determine the safety margin, given the specific value of the performance target and the traffic characteristics. This is in line with the notion of *equivalent bandwidth* proposed in [5]. A further discussion on differences and similarities (in terms of applicability and efficiency) between both equivalent-bandwidth concepts can be found in [3, Remark 1].

In the first place it depends on ϵ through the square root of its natural logarithm — for instance, it says that replacing $\epsilon = 10^{-4}$ by $\epsilon = 10^{-7}$ means that the safety margin has to be increased by about 32 percent. Second, it depends on time interval T . The parameter $v(T)$ is called the *traffic variance*, and represents the variance of traffic arriving in intervals of length T . The traffic variance $v(T)$ can be interpreted as a kind of burstiness and is typically (roughly) of the form αT^{2H} for $H \in (1/2, 1)$, $\alpha > 0$ [6, 7]. We see that the capacity needed on top of is proportional to T^{H-1} and, hence, increases when T decreases, as could be expected. In the third place, the required capacity obviously depends on the traffic characteristics, both through the “first order estimate” and the “second order estimate” $v(T)$. We emphasize that safety margins should not be thought of as fixed numbers, like the 30 percent mentioned in the introduction; instead, it depends on the traffic characteristics (i.e., it increases with the burstiness of the traffic) as well as the strictness of the performance criterion imposed.

It is important to realize that our dimensioning formula assumes that the underlying traffic stream is *Gaussian*. In our research we therefore extensively investigated whether this assumption holds in practice; due to central-limit-theorem type arguments, one expects that it should be accurate as long as the aggregation level is sufficiently high. We empirically found that aggregates resulting from just a few tens of users already make the resulting traffic stream fairly Gaussian; see [8] for precise statistical support for this claim. In many practical situations one can therefore safely assume Gaussianity; this conclusion is in line with what is found elsewhere [5–7].



■ Figure 2. Example from a university access link.

How to Use the Dimensioning Formula

The dimensioning formula presented in the previous section requires four parameters: ϵ , T , \bar{r} , and $v(T)$. As argued above, the performance parameter ϵ and time interval T must be chosen by the network manager and can, in some cases, directly be derived from an SLA. Possible values for these parameters are $\epsilon = 1$ percent (meaning that the link capacity should be sufficient in 99 percent of the cases) and $T = 100$ ms (popularly speaking, in the exceptional case that the link capacity is not sufficient, the overload situation does not last longer than 100 ms). The two other parameters, the average traffic rate \bar{r} and traffic variance $v(T)$, are less straightforward to determine and discussed in separate subsections below.

Example

A short example of a university backbone link will be presented first. In this example we have chosen $\epsilon = 1$ percent and $T = 100$ ms. To find \bar{r} and $v(T)$, we have measured all traffic flowing over the university link for a period of 15 minutes. From this measurement we have measured the average traffic rate for each 100 ms interval within these 15 minutes; this rate is shown as the plotted line in Fig. 2. The figure indicates that this rate varies between 125 and 325 Mb/s. We also measured the average rate \bar{r} over the entire 15 min interval ($\bar{r} = 239$ Mb/s), as well as the standard deviation (which is the square root of the traffic variance) over intervals of length $T = 100$ ms (2.7 Mb),

$$(\sqrt{v(T)}) = 2.7 \text{ Mb.}$$

After inserting the four parameter values into our formula, we found that the required capacity for the university access link should be $C = 320.8$ Mb/s. This capacity is drawn as a straight line in the figure. As can be seen, this capacity is sufficient most of the time; we empirically checked that this was indeed the case in about 99 percent of the 100 ms intervals.

Approaches to Determine the Average Traffic Rate

The average traffic rate \bar{r} can be estimated by measuring the amount of traffic (the number of bits) crossing the Internet link, which should then be divided by the length of the measurement window (in seconds). For this purpose the manager can connect a measurement system to that link and use tools like `tcpdump`. To capture usage peaks, the measurement could run for a longer period of time (e.g., a week). If the busy period is known (e.g., each morning between 9:00 and 9:15), it is also possible to measure during that period only.

The main drawback of this approach is that a dedicated

measurement system is needed. The system must be connected to the network link and be able to capture traffic at line speed. At gigabit speed and faster, this may be a highly non-trivial task. Fortunately, the average traffic rate can also be determined by using the Simple Network Management Protocol (SNMP) and reading the `ifHCInOctets` and `ifHCOutOctets` counters from the Interfaces MIB. This MIB is implemented in most routers and switches, although old equipment may only support the 32-bit variants of these counters. Since 32-bit counters may wrap within a measurement interval, it might be necessary to poll the values of these counters on a regular basis; if 64-bit counters are implemented, it is sufficient to retrieve the values only at the beginning and end of the measurement period. Anyway, the total number of transferred bits as well as the average traffic rate can be determined by performing some simple calculations. Compared to using `tcpdump` at gigabit speed, the alternative of using SNMP to read some MIB counters is rather attractive, certainly in cases where operators already use tools like MRTG [1], which perform these calculations automatically.

Direct Approach to Determine Traffic Variance

Like the average traffic rate \bar{r} , the traffic variance $v(T)$ can also be determined by using `tcpdump` and directly measuring the traffic flowing over the Internet link. To determine the variance, however, it is now *not* sufficient to know the total amount of traffic exchanged during the measurement period (15 min); instead, it is necessary to measure the amount of traffic for every interval of length T , in our example 1500 measurements at 100 ms intervals. This will result in a series of traffic rate values; the traffic variance $v(T)$ can then be estimated in a straightforward way from these values by applying the standard sample variance estimator.

It should be noted that, as opposed to the average traffic rate \bar{r} , now it is *not* possible to use the `ifHCInOctets` and `ifHCOutOctets` counters from the Interfaces MIB. This is because the values of these counters must now be retrieved after every interval T ; thus, in our example, after every 100 ms. Fluctuations in SNMP delay times [9], however, are such that it will be impossible to obtain the precision that is needed for our goal of link dimensioning. In the next subsections we propose a method that avoids real-time line speed traffic inspection by instead inspecting MIB variables.

An Indirect Approach to Determine Traffic Variance

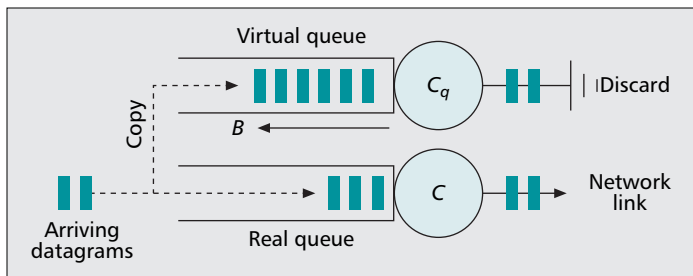
One of the major outcomes of our research [2] is an *indirect* procedure to estimate the traffic variance, with the attractive property that it avoids measurements on small timescales. This indirect approach exploits the relationship that exists between $v(T)$ and the occupancy of the buffer (in the router or switch) in front of the link to be dimensioned. This relationship can be expressed through the following formula [2]: for any t ,

$$v(t) \approx \min_{B > 0} \frac{(B + (C_q - \bar{r})t)^2}{-2 \log \mathbb{P}(Q > B)} \quad (2)$$

In this formula, C_q represents the current capacity of the link, \bar{r} the average traffic rate over that link, and $\mathbb{P}(Q > B)$ the buffer content's (complementary) distribution function (i.e., the fraction of time the buffer level Q is above B). The formula shows that once we know the buffer contents distribution $\mathbb{P}(Q > B)$, we can for any t study

$$\frac{(B + (C_q - \bar{r})t)^2}{-2 \log \mathbb{P}(Q > B)} \quad (3)$$

as a function of B , and its minimal value provides us with an



■ Figure 3. Decoupling the real queue from a virtual queue.

estimate of $v(t)$. In this way we can *infer* $v(t)$ for any timescale t ; by choosing $t = T$, we indeed find an estimate of $v(T)$, which was needed in our dimensioning formula. Theoretical justification of Eq. 2 can be found in [10].

To estimate $\mathbb{P}(Q > B)$, let us assume that a MIB variable exists that represents the amount of data in the buffer located in front of the link. This MIB variable should be read multiple times to collect N “snapshots” of the buffer contents q_1, \dots, q_N . Obviously, from these snapshots we are now able to estimate the buffer contents distribution $\mathbb{P}(Q > B)$. To determine $v(t)$, we have to fill in each possible value of B in the above formula, with $t = T$, and find that specific B for which Eq. 3 is minimal; this minimal value is then the estimate of the traffic variance we are seeking.

The advantage of this indirect approach is that it is no longer necessary to measure traffic at timescale T to determine $v(T)$. Instead, it is sufficient to take a number of snapshots from a MIB variable representing the occupancy of the buffer in front of the link. Based on extensive empirical testing we have empirically observed that the impact of the interval length hardly affects the performance of the algorithm — there is no need to take equally sized intervals, which is an important advantage of the indirect procedure. Further results on the number of buffer snapshots needed to obtain a reliable estimate of $\mathbb{P}(Q > B)$ and the measurement frequency are presented in detail in [2].

Implementation Requirements for the Indirect Approach

The indirect approach requires the existence of a MIB variable representing the length of the output queue, but such a variable has not been standardized by the Internet Engineering Task Force (IETF) yet. The variable that comes closest is `ifOutQLen` from the Interfaces MIB. In the latest specifications of this MIB module the status of this variable has been deprecated, however, which means that this variable is obsolete, although implementers may still implement it to ensure backward compatibility. In addition, the `ifOutQLen` variable measures the length of the queue in packets, whereas our procedure requires the queue length to be in bits. Although this “incompatibility” might be “fixed” by means of some probabilistic computations, our recommendation is to add to the definition of some MIB module a variable representing the length of the output queue in bits (or octets). We stress that implementing such variable should be straightforward; Random Early Detection (RED) queuing algorithms, which are widely implemented in modern routers, already keep track of this information.

A second issue regarding the indirect approach is that it may seem impossible to estimate a “usable” buffer content distribution $\mathbb{P}(Q > B)$. For example, if the capacity of the outgoing link is much higher than the traffic rate, the buffer in front of that link will (nearly) always be empty. Also in case the traffic rate approaches the link capacity, the buffer in front of that link becomes overloaded, so that we do not have any useful

information on the buffer content distribution for small values of B . To circumvent these complications, vendors of switches and routers could implement some kind of “intelligence” within their devices. Such intelligence could simulate the queuing dynamics of a *virtual* queue, with a virtual outgoing line with capacity C_q that can be chosen smaller or larger than the actual capacity. If the link is underloaded, the capacity of the virtual queue should clearly be chosen substantially smaller than the actual capacity, in order to obtain an informative estimate of the buffer content distribution; if the link is overloaded, vice versa. Procedures for detecting appropriate values for the virtual capacity are presented in [2]. Figure 3 shows the structure of such intelligence within a switch or router. Since RED-enabled routers already include much of this intelligence, implementation will be relatively straightforward.

Validation

In this section the correctness of our link dimensioning procedure will be validated in two steps. For each trace:

- First, we validate the correctness of Eq. 2. We do this by comparing the results of the direct approach to determine traffic variance to the results obtained via the indirect approach based on Eq. 2.
- Second, we validate the correctness of Eq. 1. We empirically determine the “correct” value of the link capacity; that is, we empirically find the minimum service rate needed to meet the performance criterion (T, ϵ) . We then compare the outcome of Eq. 1 with this “correct” capacity.

The next subsection starts with providing details about the measurements that were needed to perform the validation. We then present the comparison between the direct and indirect approaches. Finally, we compare the outcome of Eq. 1 with the empirical approach.

Measurements

To enable a thorough validation study, we have collected around 850 TCP/IP packet traces, based on measurements performed between 2002 and 2006. To ensure that the traffic within these traces is representative for large parts of the Internet, we have measured on five different types of links:

- *A*: A 1 Gb/s uplink of an ADSL access network. Several hundreds of ADSL customers are connected to this network; the link capacity for each individual ADSL user varies between 256 kb/s and 8 Mb/s.
- *C*: A 1 Gb/s link between a large college network and the Dutch academic and research network (SURFnet). This college network serves around 1000 students, most of them connected via 100 Mb/s Ethernet links.
- *R*: A 1 Gb/s link between a research institute and SURFnet. The research network is used by approximately 200 researchers, each having a 100 Mb/s link to the research network.
- *S*: A 50 Mb/s Internet access link of a server-hosting company. This company provides floor and rack space to clients who want to connect, for example, their Web servers to the Internet. Internally, most servers are connected via 100 Mb/s links.
- *U*: A 300 Mb/s (three parallel 100 Mb/s Ethernet links) between the residential and core networks of a university. Around 2000 students are each connected via 100 Mb/s links to this residential network; an important share of the traffic generated by these students remains within this residential network and is therefore not visible on the link toward the university’s core network.

Each trace contains 15 min worth of TCP/IP header data;

Trace	$\sqrt{v_{\text{direct}}(T)}$	$\sqrt{v_{\text{indirect}}(T)}$	
loc. A-1	0.969	1.032	147.180
loc. A-2	0.863	0.864	147.984
loc. C-1	0.796	0.802	23.894
loc. C-2	3.263	3.518	162.404
loc. R-1	0.701	0.695	18.927
loc. R-2	0.241	0.249	3.253
loc. S-1	0.447	0.448	14.254
loc. S-2	0.152	0.152	2.890
loc. U-1	1.942	2.006	207.494
loc. U-2	2.704	2.773	238.773

■ Table 1. Direct vs. indirect approach (in megabits per second).

the sizes of these traces range from a few megabytes to a few gigabytes. In total some 500 Gbytes of TCP/IP header data was collected. This data has been anonymized and can be downloaded from our Web server [11].

Traffic Variance: Direct vs. Indirect Approach

In this subsection we compare the traffic variance as can be estimated from direct link measurements (the direct approach) to the traffic variance that can be estimated using Eq. 2, that is, the approach that measures the occupancy distribution of the buffer in front of the link (the indirect approach) with an appropriately chosen value of the virtual queue's link capacity.

MIB variables that represent router buffer occupancy are not yet available. We therefore chose to simulate such a router. The simulator implements a virtual queue similar to the one shown in Fig. 3. The simulator input is the traces discussed in the previous subsection. A sufficiently large number of snapshots of the buffer content are performed to reliably estimate $\mathbb{P}(Q > B)$. We also estimated the average traffic rate of each trace, to use it in Eq. 2.

Table 1 shows, for each of the five locations, the results for two representative traces. It shows, in megabits, the square root of the traffic variance $v(T)$, and thus the standard deviation, for the direct as well as the indirect approach. We note that the table also shows the average traffic rate, which is in megabits per second. To support real-time interactive applications, the time interval T of our performance criterion was chosen to be 100 ms.

The table shows that there is just a modest difference between the traffic variance obtained using Eq. 2 and the one obtained using direct link measurements. In many cases the results using Eq. 2 differ only a few percent from the direct results. The worst result is obtained for location C, example #2; in this case the difference is about 16 percent. Observe, however, that this table may give an overly pessimistic impression, as the dimensioning of Eq. 1 indicates that the error made in the estimation of capacity is substantially smaller: on the basis of the direct variance estimate (with $\epsilon = 1$ percent) the capacity is estimated to be 261.4 Mb/s, and on the basis of the indirect variance estimate 269.2 Mb/s, there is a difference of just 3 percent.

Trace	C_A	C_B	C_C	$\Delta_{B/A}$	$\Delta_{C/A}$
loc. A-1	171.191	176.588	178.480	1.032	1.043
loc. A-2	168.005	174.178	174.218	1.037	1.037
loc. C-1	44.784	48.033	48.250	1.073	1.077
loc. C-2	265.087	261.444	269.182	0.986	1.015
loc. R-1	37.653	40.221	40.020	1.068	1.063
loc. R-2	10.452	10.568	10.793	1.011	1.033
loc. S-1	27.894	27.843	27.873	0.998	0.999
loc. S-2	7.674	7.482	7.532	0.975	0.981
loc. U-1	258.398	266.440	268.385	1.031	1.039
loc. U-2	302.663	320.842	322.934	1.060	1.067

■ Table 2. Link capacity for each of the three approaches (in megabits per second).

For space reasons, Table 1 shows only the results for some traces, but the same kind of results have been obtained for the other traces; for an extensive set of experiments see [2]. Also, results did not change significantly when we selected other values for the time interval T . We therefore conclude that our indirect approach is sufficiently accurate. This also means that for the purposes of this link dimensioning, there is in principle no need for line-speed measurements to determine traffic variance. Our experiments show that simple MIB variables indicating current buffer occupancy are sufficient for that purpose.

Required Link Capacity

Finally, this subsection validates the correctness of Eq. 1, and thus our approach to dimension network links. This is done by comparing the outcomes of three different approaches:

- Approach A: In this approach we have measured all traffic flowing over a certain link and empirically determined the minimum capacity needed to meet the performance criterion; this capacity could be considered the “correct” value. Although it is difficult to perform such measurements at gigabit speed and higher, the estimation of the minimum capacity needed to satisfy our performance criterion is rather straightforward (assuming that the link is not yet overloaded).
- Approach B: In this approach we have used Eq. 1 to determine the required link capacity. The average traffic rate as well as the traffic variance $v(t)$ have been determined in the way described in the previous section (i.e., the variance has been estimated through the direct procedure).
- Approach C: In this approach we have used both Eqs. 1 and 2. Compared to approach B, the traffic variance $v(t)$ has now been derived from the occupancy of the buffer in front of the link, as described previously (i.e., through the indirect procedure).

For all three approaches we have used the same performance criterion: the link capacity should be sufficient in 99 percent of the cases ($\epsilon = 1$ percent); and in the exceptional case that the link capacity is not sufficient, the overload situation should not last longer than 100 ms ($T = 100$ ms). Note that results using other performance criteria can be found in [2]; the findings agree to a large extent with those presented here.

Table 2 shows the outcome for the three approaches, using the same traces as before. The column C_A shows, in megabits per sec-

Traces	avg $\Delta_{C/A}$	stderr $\Delta_{C/A}$	avg $\Delta_{C/A}^*$	stderr $\Delta_{C/A}^*$
loc. A	1.04	0.02	1.04	0.01
loc. C	1.04	0.11	1.05	0.08
loc. R	0.90	0.19	1.00	0.10
loc. S	0.99	0.10	1.01	0.05
loc. U	1.01	0.07	1.03	0.06

■ Table 3. Overall validation results.

ond, the minimal required link capacity to meet the performance criterion that we (empirically) found after measuring all traffic flowing over that link. In fact, this is the actual capacity that would be needed in practice to satisfy our performance criterion; it is therefore our target value. Column C_B shows the capacity that has been estimated using Eq. 1; column C_C shows the capacity that has been estimated if additionally Eq. 2 has been used to determine the traffic variance. As shown in the last two columns, the estimated values divided by the target values are very close to 1; in all cases the differences are less than 7 percent.

Our procedure to determine link capacity has been validated not only for the 10 traces shown in Table 2, but for all 850 traces that were collected as part of our studies. The overall results for the complete procedure, approach C, are shown in columns 2 and 3 (avg $\Delta_{C/A}$ and stderr $\Delta_{C/A}$) of Table 3. For all locations but R, $\Delta_{C/A}$ is very close to 1, indicating that the bandwidth as estimated through our procedure is nearly correct.

The deviation at location R is caused by the fact that at R traffic is on average “less Gaussian” than at the other measurement locations — as our methodology assumes Gaussian traffic, some error in the resulting estimate can be expected when the traffic is “not as Gaussian.” To further investigate this, we recomputed all values, but removed the traces that were “less Gaussian” (in terms of statistics presented in [7, 8], e.g., Kolmogorov-Smirnov distance and goodness-of-fit). Columns 4 and 5 of Table 3 show the results; the differences are now 5 percent or less. It should be noted that in all cases this difference results in a slight overestimation of the required capacity; in practice this may be desirable, in particular if meeting the SLA is valued more than (temporarily) not using all transmission capacity available.

Conclusions

Motivated by the fact that rules of thumb usually lead to unreliable capacity estimates, this article focused on the development of a generic methodology for link dimensioning. It was demonstrated that the capacity of Internet links can be accurately estimated using a simple formula, which requires only four parameters. The first two of these parameters reflect the desired performance level (representing how often the offered load may exceed the available capacity, and for how long this link exceedance may last) and should be chosen by the network manager. The last two parameters reflect the characteristics of the offered traffic, and can be obtained by estimating the average link load and variance. The average link load can easily be determined by reading certain MIB variables via SNMP; tools like MRTG can be used for that purpose. Measuring traffic variance is somewhat more involved, but may be performed in a sophisticated, indirect way, using the distribution of the occupancy of the buffers located (in the router or switch) in front of the link to be dimensioned. The advantage of this indirect approach is that measurements at small timescales (whose reliability cannot be guaranteed) are no longer needed. Although

much of the intelligence to determine the buffer occupancy distribution is already implemented in current routers, the corresponding MIB variables are not yet available. Implementing these variables is argued to be straightforward, however. Our formula has been validated using 850 TCP/IP traces, collected at five different locations, ranging from ADSL access networks, university networks, and college networks to access links of server hosting companies and research institutes. The validation showed that our formula was able to determine the required link capacity with an error margin of just a few percent; our approach therefore clearly outperforms the simple rules of thumb that are usually relied on in practice.

Acknowledgments

The work reported in this article was supported in part by the EC IST-EMANICS Network of Excellence (#26854).

References

- [1] T. Oetiker, “MRTG: Multi Router Traffic Grapher,” 2003; <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>
- [2] R. van de Meent, “Network Link Dimensioning — A Measurement and Modeling-based Approach,” *Ph.D. thesis*, Univ. of Twente, 2006; <http://purl.org/utwente/56434>
- [3] J.L. van den Berg *et al.*, “QoS Aware Bandwidth Provisioning of IP Links,” *Comp. Net.*, vol. 50, no. 5, 2006.
- [4] C. Fraleigh, *Provisioning Internet Backbone Networks to Support Latency Sensitive Applications*, Ph.D. thesis, Stanford Univ., 2002.
- [5] R. Guérin, H. Ahmadi, and M. Naghsineh, “Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks,” *IEEE JSAC*, vol. 9, no. 7, 1991.
- [6] C. Fraleigh *et al.*, “Packet-Level Traffic Measurements from the Sprint IP Backbone,” *IEEE Network*, vol. 17, no. 6, 2003.
- [7] J. Kilpi and I. Norros, “Testing the Gaussian Approximation of Aggregate Traffic,” *Proc. 2nd ACM SIGCOMM Internet Measurement Wksp.*, Marseille, France, 2002, pp. 49–61.
- [8] R. van de Meent, M. R. H. Mandjes, and A. Pras, “Gaussian Traffic Everywhere?,” *Proc. IEEE ICC '06*, Istanbul, Turkey, 2006.
- [9] A. Pras *et al.*, “Comparing the Performance of SNMP and Web Services-Based Management,” *IEEE Elect. Trans. Net. Svc. Mgmt.*, vol. 1, no. 2, 2004.
- [10] M. Mandjes, “A Note on the Benefits of Buffering,” *Stochastic Models*, vol. 20, no. 1, 2004.
- [11] R. van de Meent and A. Pras, “Traffic Measurement Repository,” 2007; <http://traces.simpleweb.org/>

Biographies

AIKO PRAS (a.pras@utwente.nl) is working at the University of Twente, the Netherlands, where he received a Ph.D. degree for his thesis, *Network Management Architectures*. His research interests include network management technologies, Web services, network measurements, and intrusion detection. He chairs the IFIP Working Group 6.6 on Management of Networks and Distributed Systems and is Research Leader in the European Network of Excellence on Management of the Internet and Complex Services (EMANICS). He has organized many network management conferences.

REMCO VAN DE MEENT (Remco.vandeMeent@vodafone.com) received a Ph.D. degree from the University of Twente in 2006 for his thesis, *Network Link Dimensioning: A Measurement & Modeling Approach*. From 2006 to 2007 he worked as R&D manager at Virtu, an Internet and hosting services organization. As of January 2008, he is working at Vodafone NL. He is currently a lead designer working with the High Level Design team of the Technology — Service Delivery Department.

MICHEL MANDJES (mmandjes@science.uva.nl) received M.Sc. (in both mathematics and econometrics) and Ph.D. degrees from the Vrije Universiteit (VU), Amsterdam, the Netherlands. After having worked as a member of technical staff at KPN Research, Leidschendam, the Netherlands, and Bell Laboratories/Lucent Technologies, Murray Hill, New Jersey, as a part-time full professor at the University of Twente, and as department head at CWI, Amsterdam, he currently holds a full professorship at the University of Amsterdam, the Netherlands. His research interests include performance analysis of communication networks, queueing theory, Gaussian traffic models, traffic management and control, and pricing in multi-service networks.

BART NIEUWENHUIS (l.j.m.nieuwenhuis@utwente.nl) is a part-time professor at the University of Twente, holding the chair in QoS of Telematics Systems. He is owner of the consultancy firm K4B Innovation. He is chairman of the innovation-driven research program Generic Communication (part of R&D programs funded by the Ministry of Economic Affairs) and advisor to the Netherlands ICT Research and Innovation Authority.