Generalized Processor Sharing Performance Models for Internet Access Lines

J.V.L. Beckers[1], I. Hendrawan[2], R.E. Kooij[1], R.D. van der Mei[1,3]

[1]KPN Research
QoS Control and Network Planning
Leidschendam, the Netherlands
{J.V.L.Beckers, R.E.Kooij, R.D.vanderMei}@kpn.com

[2]University Twente
Applied Education
Enschede, the Netherlands

[3]Vrije Universiteit Amsterdam
Mathematics and Computer Science
Amsterdam, the Netherlands

Abstract. For Internet service providers (ISPs) the proper dimensioning of Internet access lines is essential. Under-dimensioning generally leads to a less than acceptable end-to-end Quality of Service (QoS) perceived by the subscribers, and over-dimensioning is generally too expensive. In this paper, we demonstrate empirically that the Generalised Processor Sharing (GPS) model can be used to accurately describe the flow-level characteristics of the traffic carried by Internet access lines, based on analysis of IP packet-level traces with hundreds of subscribers connected to the Internet over high-speed access lines. Using properties of the GPS-model, we establish simple relations between the access line capacity and the utilisation of the access lines on the one hand and the download times of Internet objects on the other hand. Subsequently, we study the impact of the TCP slow start algorithm on the download times of objects. We propose simple and fast approximations for the average object download time as a function of the object size, encompassing the effect of the round-trip times, maximum window size and modem speed. Empirical data demonstrate the accuracy of the approximations.

1 Introduction

Over the past few years the use of Internet services has experienced tremendous growth. Internet applications have evolved from standard document-retrieval functionality to advanced multimedia services. One of the main barriers to the continuing success of the Internet in the near future is Internet congestion, which in many cases leads to unacceptably long response times, particularly for real-time applications like the World Wide Web (WWW), PC banking and other on-line services. Consequently, the ability to somehow deliver QoS guarantees to the end users strongly enhances the competitive edge crucial for the business of Internet service providers (ISPs). To cope with the increasing bandwidth requirements, new technologies are being developed for high-speed access networks (e.g., CATV Internet, ADSL). This, in turn, leads to an increase in the amount of bandwidth consumption in both the core network and the access trunk lines connecting the access lines to the core network (cf. [6]). Figure 1 illustrates a typical situation where subscribers have access over a customer line to the core network (where most of the Internet server farms reside) via an access multiplexer (MUX) and an access trunk line.

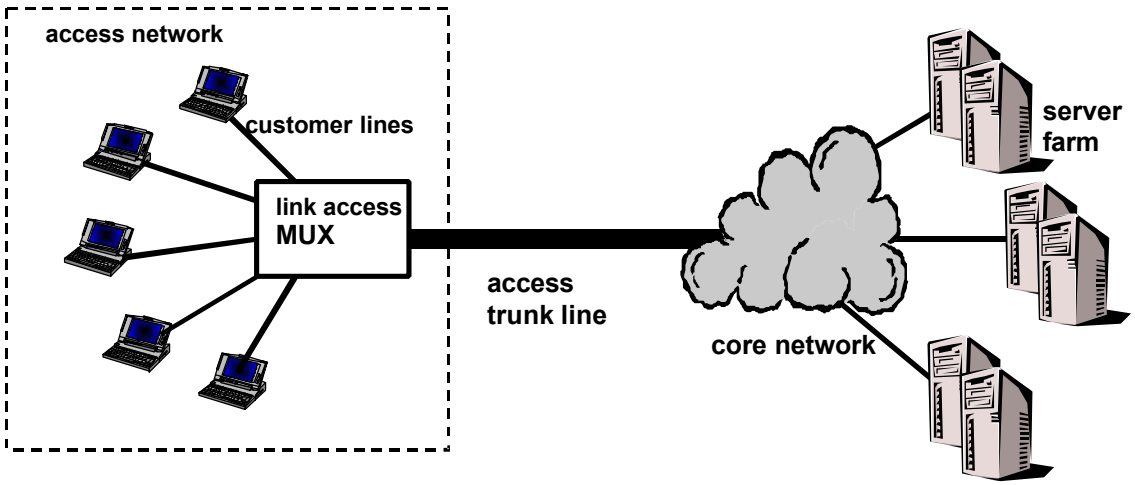

Figure 1: Access trunk line connecting the access network to the core network.

For a cost-efficient design of their infrastructure, ISPs need to properly dimension the trunk lines giving the subscribers access to the Internet: Under-dimensioning generally leads to a less than acceptable QoS perceived by the subscribers, whereas over-dimensioning may be too expensive. In this context, it is important to note that a significant amount of multiplexing gain can be achieved by exploiting the statistical features of the downstream traffic, which is predominantly elastic TCP traffic (cf. [6]). To exploit the potential for statistical multiplexing gain in the dimensioning of the Internet access lines, a proper performance model that accurately describes the relevant characteristics of the downstream traffic is needed.

Traffic characterisation in IP networks has been subject to extensive research over the past few years. Analysis of empirical traffic measurements has revealed that packet-level WAN traffic exhibits two intriguing properties: long-range dependence (LRD) and self-similarity (cf. Park and Willinger [5] for an overview). Processor sharing (PS) models provide a simple and effective way of modelling the elastic properties of traffic carried by transport protocols that adapt the transmission rate to the available capacity in the network. Nabe et al. [4] propose to use an M/G/1-PS queueing model to discuss a design methodology of Internet access networks. Lindberger [3] and Bonald and Roberts [1] propose an M/G/c-PS queueing model to describe the flow-level behaviour for a link that can accommodate c times the peak rate of an individual source. Charzinski [2] defines the so-called fun factor as the expected value of the ratio between the data transfer time at link rate and the actually observed data transfer time. The main shortcoming of these PS-based models is that their validity has not been validated by means of empirical data. The reader is referred to [8] (and references therein) for a fairly complete overview of the state-of-the-art in IP traffic characterisation and modelling.

Nabe et al. [4] have proposed a design methodology for access trunk lines, for architectures as depicted in Figure 1. From now on we use the term “access line” to denote the access trunk line. It is assumed that the capacities in the access and customer lines are small compared to the capacities in the backbone. In [4] it is assumed that a single server (the processor) is servicing all or some sets of jobs in the system at a rate that depends on the state of the system. The processing time is the total time a job spends in the system. More precisely, it is the time from epoch of arrival to epoch of its departure following service completion. The main interest is in the processing time conditioned on the service time (the length) of a job. For the service discipline, Nabe et al. [4] adopted the processor sharing (PS)-scheduling algorithm because multiple HTTP requests can share the access line by virtue of underlying TCP. The processor-sharing is described as follows: when there are $n \geq 1$ jobs present in the system then each job receives service at a rate which is $1/n$ times the rate of service which a solitary job in the processor would receive. Therefore the rate of service received by a specific job fluctuates with time and its processing time depends not only on the jobs in the processor at its arrival, but also on subsequent arrivals. As known from literature, the PS scheduling discipline gives smaller delays for customers with less work demands when compared with FIFO scheduling.

The aim of this paper is to propose and test simple and fast approximations for the conditional download times of objects over Internet access lines. These approximations, in turn, can be used to develop simple rules for dimensioning of Internet access lines with high-speed access rates. To this end, we have performed extensive IP-packet-level measurements in a realistic experimental set-up, where hundreds of customers were connected to the Internet via ADSL. The experimental results demonstrate that the flow-level characteristics of the traffic patterns can be modelled accurately by a Generalised Processor Sharing (GPS) model, with proper choices of the

parameters. Using properties of the GPS-model discussed in [7], we establish simple relationships between the access line capacity and the utilisation of the access lines on the one hand, and the download times of Internet objects on the other hand. Subsequently, we consider the impact of the TCP slow start algorithm on the download times of Internet objects (e.g., Web documents). We propose simple and fast approximations for the average object download time as a function of the object size, explicitly taking into account the impact of the round-trip times, maximum window size and the modem speed. Empirical data demonstrates that the approximations are accurate in realistic situations.

The remainder of this paper is organised as follows. In section 2 we discuss several preliminary results. In section 3, we use these results to propose several simple and fast approximations for the mean download times over Internet access lines. In section 4, the approximations are validated by experimental data obtained from real-life measurements. Finally, in section 5 we give concluding remarks, and address a number of topics for further research.

2 Preliminaries

In this section we briefly review the M/G/1-PS model and the M/G/1-GPS model, and give exact expressions for the conditional mean sojourn times. These results will be used in the next section to develop approximations for the download times of objects over Internet access lines.

2.1 M/G/1-PS model

Consider an M/G/1-PS model, where customers arrive at the system according to a homogeneous Poisson arrival process with rate λ. The service-time requirements have a general distribution with mean β, and the service rate is C. The load of the system is defined by $\rho := \lambda\beta / C$. It is assumed that the stability condition $\rho < 1$ is satisfied and that the system is in steady state. Denote the service-time requirement of an arbitrary customer by T^{PS}, and the sojourn time by S. Then the mean conditional sojourn time is given by the following expression: For $t > 0$,

$$E[S \mid T^{PS} = t] = \frac{t}{C(1-r)}. \tag{1}$$

Notice that (1) depends on the service-time distribution only through the mean $E[T^{PS}]$.

2.2 M/G/1-GPS model

The M/G/1-GPS model is similar to the M/G/1-PS model, except that the rate at which each of the customers is served can be an arbitrary (under weak restrictions) function of the number of customers in the system. More precisely, let $f(n)$ denote the fraction of the total service rate C that each customer receives when there are n customers in the system. Thus, when there are n customers in the system, *each* of these customers is served at rate $f(n)C$. Moreover, define the function $f(.)$ as follows:

$f(0) := 1$,

and

$$f(n) := \left[\prod_{j=1}^{n} f(j) \right]^{-1}, \text{ for } n = 1, 2, \ldots .$$

Cohen [7] gives the following exact expression for the steady state distribution of N^{GPS}, defined as the number of customers in the GPS system: For $n = 0,1,\ldots$

$$\Pr\{N^{GPS} = n\} = \frac{r^n}{n!} f(n) / \sum_{j=0}^{\infty} \frac{r^j}{j!} f(j). \tag{2}$$

Notice that in the standard PS-model (see section 2.1) we have $f(n)=1/n$ ($n = 1, 2,\ldots$), and hence, $f(n)=n!$ ($n = 1, 2,\ldots$), which implies that the steady-state distribution of the number of customers on the standard PS model is

geometrically distributed with parameter *?*. Denoting the service-time requirement of an arbitrary customer by T^{GPS}, Cohen [7] gives the following expression for the mean conditional sojourn time: For $t > 0$,

$$E[S \mid T^{GPS} = t] = \frac{t}{C} \frac{\sum_{n=0}^{\infty} \frac{r^n}{n!} f(n+1)}{\sum_{n=0}^{\infty} \frac{r^n}{n!} f(n)}. \quad (3)$$

Using (2) and several straightforward algebraic manipulations it is easily verified that (3) can be rewritten as follows: For $t > 0$,

$$E[S \mid T^{GPS} = t] = \frac{x}{C} \frac{E[N^{GPS}]}{r}. \quad (4)$$

3 Approximations for the download times over Internet access lines

In this section we apply the results presented in section 2 to develop approximations for the download times of documents over Internet access lines. To relate the processor sharing models to document download times the service capacity C represents the bandwidth of the Internet access line, the customers represent document-retrieval requests, and the download times are modelled by the sojourn times of the customers.

3.1 Approximation based on standard M/G/1-PS model ("PS")

Assume that document-retrieval requests arrive according to a Poisson process with arrival rate and that the bandwidth of the Internet access line is C (bits per second). Denote by D the size of a document (in bytes), and denote the utilisation of the access line by $\rho := \lambda E[D] / C$. Then according to the standard M/G/1-PS model, the mean download times of a document is of size d (bytes) is given by the following expression: For $d > 0$,

$$E[S \mid D = d] = \frac{8d}{C(1-r)}. \quad (5)$$

This approximation, which will throughout be referred to as "PS", was also proposed in [4].

3.2 Approximation based on GPS-model ("GPS")

The main shortcoming of the approximation based on the M/G/1-PS model discussed in 3.1 that it implicitly assumes that a single user can potentially consume all access line capacity. In practice, however, access link capacities are orders of magnitude larger than the individual customer line capacities. Therefore, one may expect that the object download times based on approximation "PS" (see 3.1) significantly under-estimate the real object download times. To cover this phenomenon, in this section we apply the so-called generalised processor sharing (GPS) model that significantly enhances the modelling capabilities of processor sharing models.

Let C_{cust} denote the capacity of the customer line (in bits per second), and define $d := C_{cust} / C$, i.e., the fraction of the access line capacity that a single user can potentially occupy. Throughout it is assumed that $C_{cust} < C$. To model the impact of the limitations of the customer line capacities compared to the access trunk capacities, we assume that the amount of capacity per user when n users are active is given by $f(n)C$, where $f(n) := d$ for $n < 1/d$, and $f(n) := 1/n$, for $n \geq 1/d$, as illustrated by Figure 2.

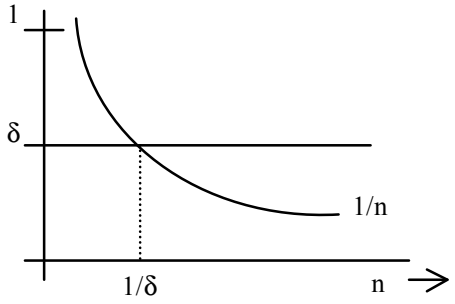

Figure 2: The function f(n)

Define $n_0 := \lceil 1/d \rceil = \lceil C/C_{cust} \rceil$, where $\lceil k \rceil$ denotes the smallest integer equal or greater than k. Then $f(n)=d$ for $n < n_0$, and $f(n) = 1/n$, for $n \geq n_0$, which is readily seen to imply that

$$f(n) = \left[\prod_{j=1}^{n} f(j) \right]^{-1} = d^{-n}, \text{ for } n < n_0, \tag{6}$$

and

$$f(n) = \left[\prod_{j=1}^{n_0-1} d \prod_{j=n_0}^{n} \frac{1}{j} \right]^{-1} = d^{1-n_0} \frac{n!}{(n_0-1)!}, \text{ for } n \geq n_0. \tag{7}$$

It is tedious but fairly straightforward to verify that relations (2), (6) and (7) lead to the following expression for the expected number of customers system:

$$E[N^{GPS}] = \frac{\sum_{n=0}^{n_0-1} \frac{n s^n}{n!} + \frac{[(1-n_0)r + n_0] r s^{n_0-1}}{(n_0-1)!(1-r)^2}}{\sum_{n=0}^{n_0-1} \frac{s^n}{n!} + \frac{d^{1-n_0} r^{n_0}}{(n_0-1)!(1-r)}}, \tag{8}$$

with $s := r/d$.

Combining relations (4) and (8) we obtain the following expression for the conditional download times of a document of size d: For $d > 0$,

$$E[S \mid D = d] = \frac{8d}{C} H_d(r), \text{ where } H_d(r) := E[N^{GPS}]/r. \tag{9}$$

To evaluate $H_d(r)$ according to (8)-(9) a key role is played by the parameter $d = C_{cust}/C$, the ratio between the available bandwidth on the customer line and access line. As a first approach we will assume that C_{cust} is determined by the following two factors:

1. The maximum window size of the TCP flow ($W_{\max}$) and the mean round-trip time (RTT), assuming that the flow start sending at the maximum window size from the start and no losses occur.
2. The modem speed which will be denoted by C_{mod}.

Then it follows that $C_{cust} = \min\{8W_{\max}/RTT, C_{\mathrm{mod}}\}$, where $W_{\max}$ is the maximum window size (in bytes), RTT is the mean round-trip time (in seconds) and C_{mod} the modem speed (in bits per second). This observation leads to the following approximation for the mean conditional download times of an object of size d: For $d > 0$,

$$E[S \mid D = d] = \frac{8d}{C} H_d(r), \text{ with } d = \min\{8W_{\max}/RTT, C_{\mathrm{mod}}\}/C, \tag{10}$$

where $H_d(r)$ is defined in (9).

3.3 Refinement: inclusion of TCP slow-start for large downloads ("GPS-slowstart")

In 3.2 we have neglected the dynamics of the slow-start phase in TCP. To cover the impact of TCP slow-start on the object download times, it seems reasonable to assume that inclusion of TCP slow start leads to an additional 4 RTTs of delay for each object. This "loss" of 4 RTTs caused by TCP slow start is motivated by the following arguments. First RTT: it takes one RTT to set up a TCP connection between the client and the server (the well-known 3-way handshake for TCP connection set-up, see [11]). Second RTT: the client sends 1 packet, the server sends an ACK, and the window size is doubled. Third RTT: client sends two packets, the server sends two ACKs, and the window size is increased to four packets. Fourth RTT: the client sends four packets, the server sends four ACKs, and the window size is increased to 8 packets, which is generally larger than the maximum window size. To this end, notice that in many Operating Systems (e.g., Windows 95 and NT [10]) the maximum window size defaults to 8760 bytes, whereas the maximum packet size is typically on the order of 1460 bytes (see also the experimental results discussed in section 4), so that the maximum window size contains at most 6 packets.

These observations support the observation that TCP slow start goes at the expense of about 4 RTT's. Hence, when the document size D is sufficiently large the mean conditional sojourn time (in seconds) can be approximated as follows: For d large,

$$E[S \mid D = d] = \frac{8d}{C} H_d(r) + 4RTT, \text{ with } d = \min\{8W_{\max}/RTT, C_{\mathrm{mod}}\}/C. \quad (11)$$

The accuracy of the approximation (11), throughout referred to as "GPS-slowstart", is validated in section 4.

3.4 Further refinement: inclusion of TCP slowstart for small downloads ("GPS-slowstart+")

For small documents TCP will actually never leave the slow-start phase. This behaviour is not captured by the approach in the previous sub-section. In order to model processing times for small documents, similar arguments as discussed in section 3.3 can be used to motivate that the number of RTTs needed to transport x packets is given by $\lfloor {}^2\log x + 1 \rfloor$. To this end, it is readily seen that it takes 1 RTT to set up a connection, one additional RTT to transport 1 packet, one additional RTT to transport 2 additional packets, and one additional RTT to transport 4 additional packets, and so on. Noting that an object of size d (bytes) is packetized in $x = \lceil d / P_{size} \rceil$ packets, we obtain the following formula for the expected processing time for sufficiently small documents of size D (in bytes): For d small,

$$E[S \mid D = d] = \lfloor {}^2\log \lceil d / P_{size} \rceil + 1 \rfloor RTT \quad (12)$$

where P_{size} is the packet size (in bytes) and $\lfloor k \rfloor$ denotes the largest integer smaller or equal to k. Combining the models given in (11) and (12) we arrive at the following enhanced processor sharing model: For $d > 0$,

$$E[S \mid D = d] = \min\{\frac{8d}{C} H(r) + 4RTT, \lfloor {}^2\log \lceil d / P_{size} \rceil + 1 \rfloor RTT\}. \quad (13)$$

This approximation will be throughout referred to as "GPS-slowstart+".

To compare the four approximations discussed above, consider the model with the following parameters: C = 10 Mbps, W_{max} = 8 kB, RTT = 105 ms, ρ = 0.2, P_{size} = 500 bytes, C_{mod} = 2.5 Mbps. Figure 3 below shows the mean download times as a function of the document size, for each of the four approximations.

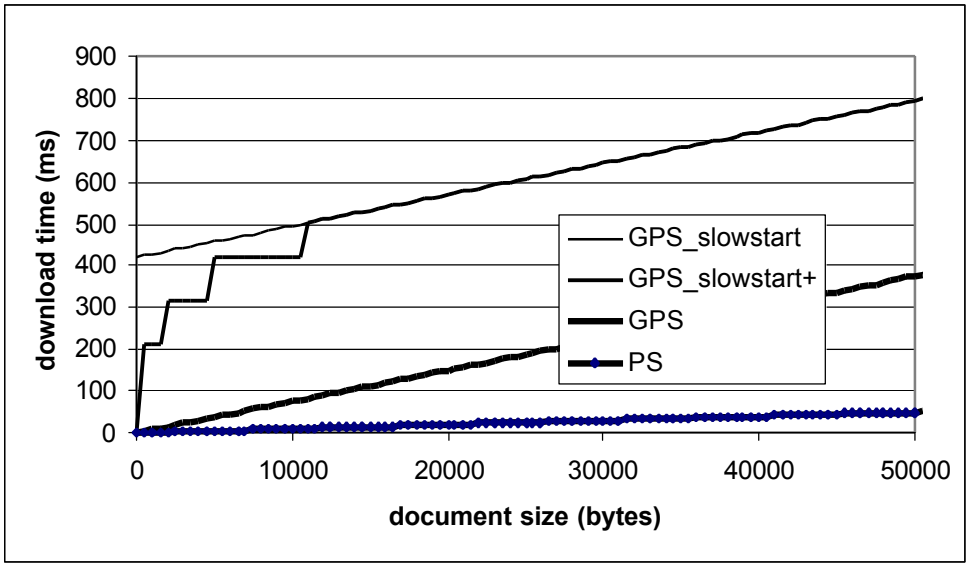

Figure 3: Approximations for the mean download times as a function of the document size.

Here, the four models are denoted by "PS" (see section 3.1), "GPS" (see 3.2), "GPS-slowstart" (see 3.3) and "GPS-slowstart+" (see 3.4). The results in Figure 3 demonstrate that the predicted approximations for the conditional download times may vary significantly among the four approximations. In the next section the accuracy of the approximations will be discussed in detail.

4 Validation

To validate the approximations discussed in the previous section, we have performed extensive performance measurements in a real-life test environment. In 4.1 the experimental set-up is discussed in detail, and in 4.2 the results of the experiments are outlined.

4.1 Experimental Setup

For a two-month period, packet-level traffic measurements have been taken on an ADSL test service of around 500 customers. These measurements were used to analyse the download behaviour of customers. The test set-up is illustrated by Figure 4. The end-users received ADSL-connectivity via Fast-Ethernet connections to the Internet. The modems were 2.5 Mbps downstream.

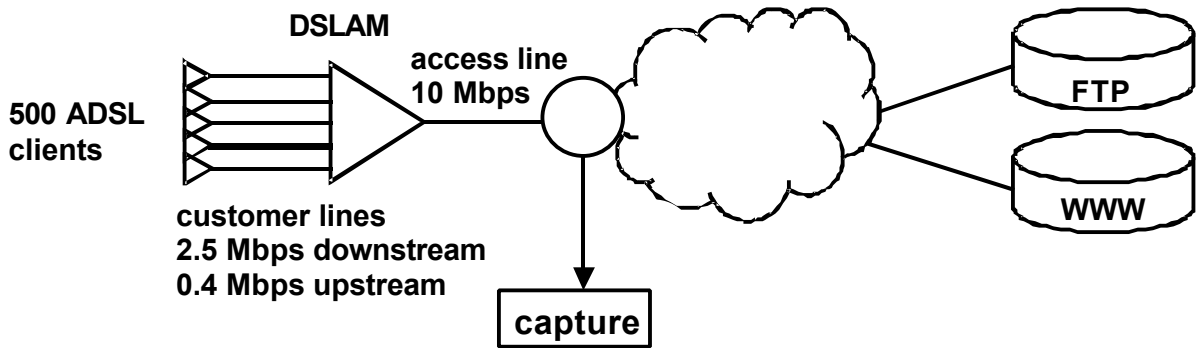

Figure 4: Experimental set-up.

The measurement traces were collected using the sniffer application 'snoop' [9] running on a workstation connected to the central router (see Figure 4). The captured traces consist of the IP header information and a time stamp (to a resolution of 0.01 ms) for each packet. Over 40 measurements were taken during various hours (morning, afternoon, evening) over a two-month period. The duration of each measurement was 0.5 to 3 hours. In total, around 7.5 gigabytes of data was collected and stored in trace files. Several web-browsing sessions of substantial length were selected from the traces for further analysis.

4.2 Measurement results

The sojourn time for a www-download depends on several system parameters. These will come out as a result of the trace analyses described in the sections below.

4.2.1 Comparison between the $GPS_{slowstart+}$ model and experimental data

Two parameters that occur in the $GPS_{slowstart+}$ model that are easily obtained are the access line and customer line capacities. For the experimental set-up, these parameters are equal to C = 10 Mbps and C_{mod} = 2.5 Mbps. The load on the access line was low during our measurements (less than 1%). In Figure 5, the processing time is plotted as a function of the document size for two relatively long download sessions with comparable round-trip times (RTT), together with the prediction of the $GPS_{slowstart+}$ model as given by equation (11). The RTT in the measurements was defined as the response time to a HTTP request. For both downloads in Figure 5 the measured RTT is 105 ms. It is explained in section 4.2.4 that 28 ms should be added to obtain the effective RTT. P_{size} was found to be 1500 Bytes in session A. For session B, P_{size} is somewhat lower (1200 bytes). Furthermore a maximum TCP window size of 8760 bytes is assumed, which is the default value for Windows 95 and NT systems [10]. The document size is defined as the number of downloaded bytes at any time. A single WWW-session thus produces a series of data points in Figure 5. Only downstream bytes are counted.

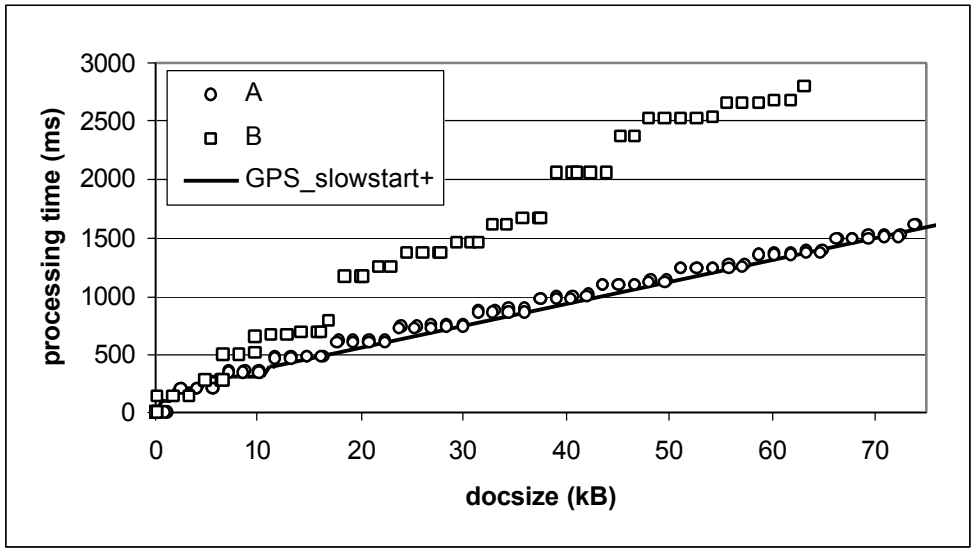

Figure 5: Download time of WWW-pages as a function of the document size.

The results in Figure 5 show that agreement between the $GPS_{slowstart+}$ model and measurement A is good. Measurement B suffers from extra delays. We will come back to this in section 4.2.4.

4.2.2 TCP window size/RTT ratio

We have observed that the download rates reach a maximum of after 500 to 700 ms (i.e. 5 to 7 RTT's). This initial period of low throughput is clearly identified as the slow start of TCP. The slow start will be discussed further in section 4.2.3. Table 1 shows measured maximum rates for several values of the measured RTT, all for a situation of a low load, in the order of 1%. Obviously the maximum download rates for all the traces in Table 1 are much lower than the bandwidth of the customer line (2.5 Mbps) and that the download rates are limited by another factor. We calculate the maximum rates determined by the TCP maximum window size (assumed to be 8760 bytes) and the RTT. The results in Table 1 show that the calculated maximum rates (using the measured RTT) are in the range of the measured values. The results get even better if we ad 28 ms to the measured RTT, see also section 4.2.4. The TCP window size is thus identified as a significant rate-limiting factor in these download-sessions. Other effects, causing additional delays will be discussed in section 4.2.4.

Table 1: Calculated and measured rates of download session during non-busy hours.

Session	Measured RTT (ms)	Measured max rate (kbps)	Calculated max rate (kbps)	Corrected rate (kbps, see 4.2.4)
A	105	487	667	526
B	105	480	667	526
C (not in Figure 5)	105	495	667	526
D (not in Figure 5)	100	580	700	570
E (not in Figure 5)	100	571	700	547
F (not in Figure 5)	180	330	389	337
G (not in Figure 5)	140	400	500	417
H (not in Figure 5)	80	630	876	648

4.2.3 TCP slow-start

The slow start of TCP was already identified in section 4.2.2 as a cause for delay during the first 5 to 10 RTT's of a download. In Figure 6 we have plotted the download of a WWW-page containing several inline objects, where we have reset both the time and the document size after each object. The slow start is clearly visible for the download of the first object. The persistent HTTP connection avoids the slow start in all subsequent downloads. The net delay caused by the slow start is estimated to be 500 ms. We have analysed several traces similar to this example and found that, in general, the slow start causes roughly 4 RTTs additional delay.

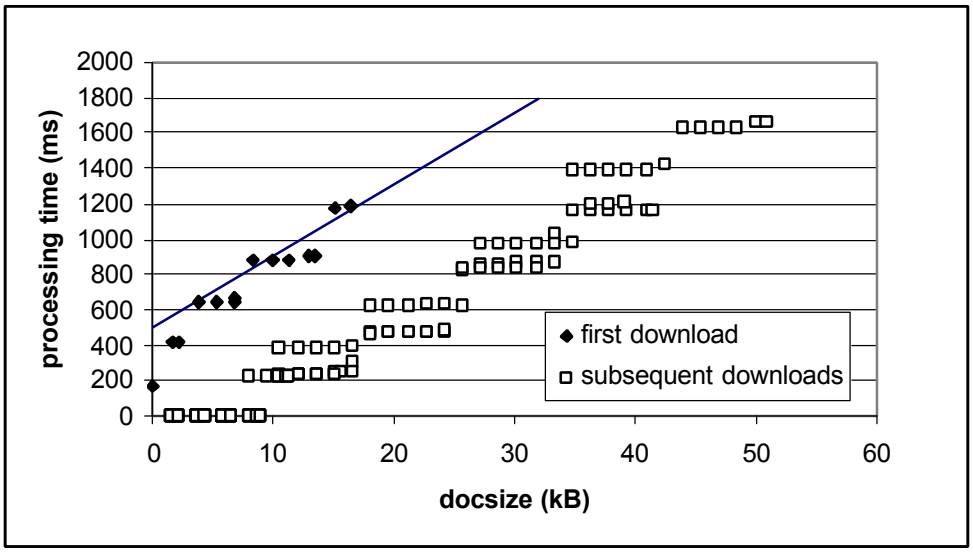

Figure 6: First and subsequent downloads for a persistent HTTP connection.

4.2.4 Other effects

Customer line bandwidth

Although the modem speed (or the bandwidth of the access line) is not the bottleneck for the system under study, it is still a source of delay. The 8760 bytes of the maximum window size are not delivered instantaneously to the client as they leave the central router. This is visualised in Figure 7. It takes 28 ms to carry the 8760 bytes over the 2.5 Mbps modem. Although this is much shorter than the typical RTT (80 to 180 ms), the 28 ms should be taken into account in the calculation of the maximum attainable download rate. We then obtain the numbers in the last column in Table 1. The results are close to the measured download rates, which indicates that we have captured all significant rate-limiting factors in the model.

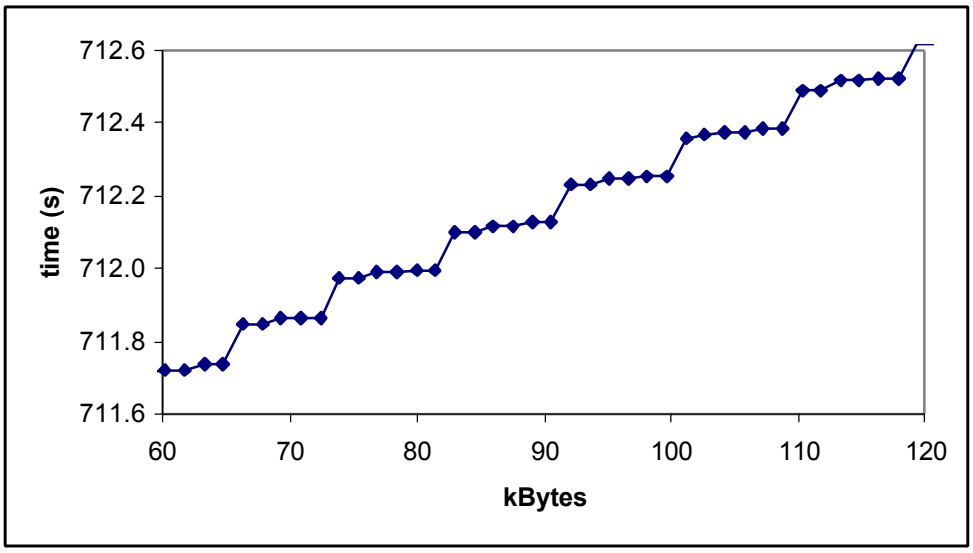

Figure 7: Detail from a session, showing the delay caused by the modem speed relative to the RTT.

Multiple inline objects

Maximum speed is only attained by download sessions of single (large) files. An interesting observation from Figure 5 is that session B seems to suffer from several extra delays, each of around 500 ms. Consequently, the average throughput in session B is significantly lower than the maximum rate. A closer look at the traces of this session (and other similar sessions) reveals that the additional delays occur at the start of new inline objects in www-pages. This is presumably due to delay of the WWW-server (see, e.g., Van der Mei et al [12] for a detailed discussion on Web server delay). The server delay per object varies between 10 and several 100 ms. The magnitude of the delay obviously depends on the type and load of the www-server. The small objects also cause terminating IP packets that are smaller than their maximum size (1500 bytes). For session B in Figure 5 we measured an average packet size of 1200 bytes. This also reduces the average download speed.

Multiple parallel TCP connections

It is well known that multiple TCP connections (maximum 4) can exist in parallel between the WWW-server and the client. The TCP connections are used for simultaneous downloading different inline objects of a WWW-page. In our traces we found several examples of this phenomenon, but never more than two parallel connections. An example is shown in Figure 8. The multiple TCP connections can, in principle, increase the download speed by a factor equal to the number of parallel connections. However, the multiple TCP connections are usually found for downloads of WWW-pages containing many small objects. This causes delay and usually annihilates the effect of the multiple TCP connection.

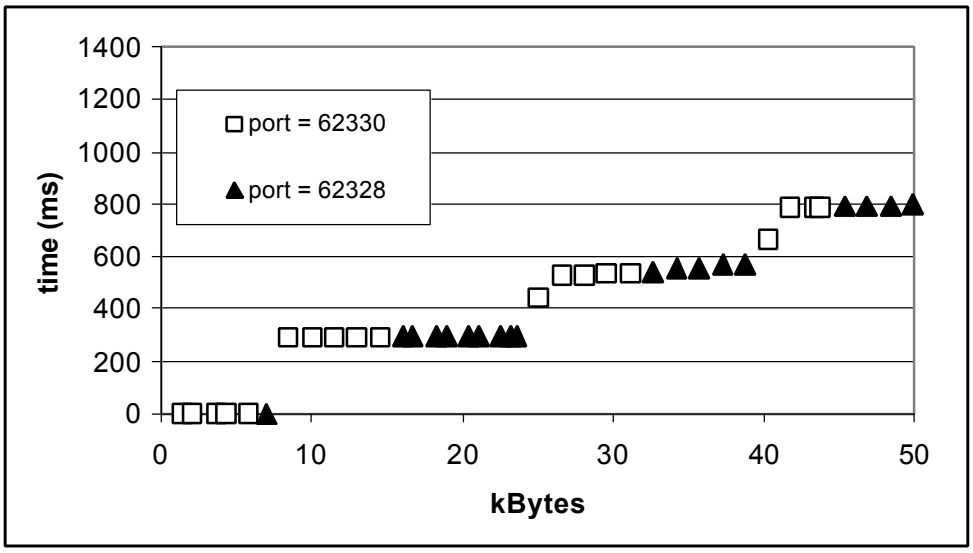

Figure 8: Example of two parallel TCP connections (two ports) increasing the bit rate of the download.

5 Concluding remarks and suggestions for further research

We have proposed a simple and fast approximation for the mean object download time as a function of the document size, for downloads by TCP over Internet access lines. Our approximations cover the impact of a variety of factors, such as (1) the bandwidth of the customer line or modem speed, (2) the TCP-maximum window size, (3) the mean round-trip times, (4) TCP slow start, (5) server delay in case of many objects in the WWW-page, (6) the load on the access line, and (7) the maximum packet size. Experimental data obtained over a two-month period on an ADSL test service of around 500 customers, show that the approximations indeed yield accurate predictions for the document download times.

The results in this paper demonstrate that the predictions of the object download times based on the M/G/1-PS model discussed by Nabe et al. [4] tend to become inaccurate when the customers line speed is significantly smaller than the access trunk line capacity. This observation addresses a significant shortcoming of the M/G/1-PS model. In addition, experimental results show that the GPS-based performance predictions are significantly more accurate. In particular, the "GPS-slowstart+" approximation, explicitly taking into account the impact of TCP slow start, appears to lead to good estimations for the mean object download times.

The GPS-model has also been proposed recently by Bonald and Roberts [1] for modeling flow-level performance over Internet backbone links. The results in this paper give experimental support for the validity of the modelling approach proposed in [1].

We conclude with a number of topics for further research. First, the accuracy of the $GPS_{slowstart+}$ model has not been validated for high loads. (The reason for this addresses the nature of real-life measurements: during the specific night that we installed all the equipment to perform the measurements, the Dutch soccer team was playing a qualification game for the EURO-2000, the European championships. Consequently, the amount of traffic measured during that time frame was significantly less than during normal measurement intervals.) In particular we will need to collect traffic traces where the load at the access trunk load is high. It can be anticipated that at high loads the $GPS_{slowstart+}$ model might turn out to be too simple because it fails to capture the impact of buffering or even loss of TCP packets, which typically occur at high loads. Another shortcoming of the GPS model in general, is that it assumes that the processing speed (bandwidth) is shared equally among the customers (flows). In reality, however, bandwidth is not equally shared among flows, for example caused by different modem speeds and different TCP parameter setting (e.g., specifics of the slow start algorithm, maximum window size, fast retransmit algorithm). Development and analysis of models including unequal sharing of bandwidth among flows is a topic for further research. The results also lead to theoretical challenges related to the analysis of the GPS model. For example, note that for the GPS-model Cohen [7] gives an explicit expression for the probability distribution of the number of customers (flows) in the system. Little's Law then leads to an expression for the mean sojourn time of an arbitrary customer. However, Little's Law can not be applied to obtain expressions for the higher moments and the complete distribution of the sojourn times in the

GPS model. In this context, notice that the sojourn time of a flow can be viewed as a (surrogate) measure for the response times for an object download over the Internet access line. Derivation of (approximate) expressions for the higher moments and the distribution of the sojourn times in the GPS-model is an interesting topic for further research.

6 References

1. T. Bonald and J. Roberts (2000). Performance of bandwidth sharing mechanisms for service differentiation in the Internet. In: *Proceedings 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management* (Monterrey, September), 22-1 – 22-10.
2. J. Charzinski (2000). Fun factor dimensioning for elastic traffic. In: *Proceedings 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management* (Monterrey, September), 11-1 – 11-9.
3. K. Lindberger (1999). Balancing Quality of Service, pricing and utilization in multiservice networks with stream and elastic traffic. In: *Proceedings ITC16* (Edinburgh, UK), 1127-1136.
4. M. Nabe, M. Murata and H. Miyahara (1998). Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines. *Performance Evaluation* 34, 249-271.
5. K. Park and W. Willinger (eds.). *Self-Similar Network Traffic and Performance Evaluation* (Wiley, New York), 2000.
6. N. Vicari and S. Kohler (2000). Measuring Internet user traffic behavior dependent on access speed. In: *Proceedings 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management* (Monterrey, September), 5-1 – 5-7.
7. J.W. Cohen (1979). The multitype phase service network with generalized processor sharing. *Acta Informatica* 12, 245-284.
8. Y. Levy and J. Roberts (eds.). *Proceedings 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management* (Monterrey, September 2000).
9. http://www.enteract.com/~lspitz/snoop.html
10. http://www.networkcomputing.com/1013/1013ws1side1.html
11. W.R. Stevens (1996). TCP/IP Illustrated, vol. 1 (Addison-Wesley).
12. R.D. van der Mei, R. Hariharan and P.K. Reeser (2001). Web server performance modeling. *Telecommunication Systems* 16, 361-378.