End-to-End Queuing Delay Assessment in Multi-service IP Networks

D. De Vleeschauwer, M.J.C. Büchli, A. Van Moffaert
Alcatel Bell, Network Strategy Group
Francis Wellesplein 1
B-2018 Antwerp, Belgium
E-mail: {danny.de_vleeschauwer, maarten.buchli, annelies.van_moffaert}@alcatel.be

and

R.E. Kooij
KPN Research Leidschendam
St. Paulusstraat 4
2264 XZ Leidschendam
Postbus 421, 2260 AK Leidschendam, The Netherlands
E-mail: R.E.Kooij@kpn.com

Abstract. Packet-based networks are more and more used to transport interactive streaming services like telephony and videophony. To guarantee a good quality for these services, the queuing delay and delay jitter introduced in the transport of voice or video flows over the packet-based network should be kept under control. Because data sources tend to increase their sending rate until (a part of) the network is congested, mixing real-time traffic and data traffic in one queue would lead to unacceptable high delays for real-time services. Therefore, voice and video packets need to get preferential treatment (e.g. head-of-line priority) over data packets in the network nodes. Therefore, the queuing behavior of the voice and video packets can be studied more or less independently from the traffic generated by data services. Simple methods to assess the end-to-end delay are primordial. Since it is well known that an aggregate of voice (and CBR video) sources is accurately modeled by a Poisson arrival process and that delays in consecutive nodes are more or less statistically independent, this boils down to developing methods to calculate quantiles of the total queuing delay through a system of $N$ statistically independent M/G/1 nodes. This paper develops four methods to calculate quantiles of the total queuing delay: a Gaussian method, a method based on the numerical inversion of the moment generating function of the total queuing delay developed by Abate & Whitt and two methods based on the assumption that the tail distribution of the individual queuing delay of one node is approximately exponential. The Gaussian method is the simplest, but only gives crude results. The method of Abate & Whitt is the most complex and breaks down for large quantiles. The methods based on the assumption of an exponential tail produce results that are more or less equally accurate as long as there is a node where the load is high enough.

Keywords. Queuing delay, packet-based voice/video transport, voice over IP.
1. Introduction

Since the Internet Protocol (IP) is considered to be an enabling technology for multi-service networks, the past few years also real-time services have been gradually deployed over IP networks, which were traditionally used for data services. However, the current Internet model provides only a best effort service. Delay and packet loss may be introduced depending on the congestion state on the end-to-end route through the network, especially when voice, video and data are transported concurrently over the network. To offer the possibility to implement reliable and predictable real-time services (e.g. telephony) over IP networks, the Internet Engineering Task Force (IETF) has defined two approaches to support Quality of Service (QoS) in IP networks: the Integrated Services model (Braden et al, 1994) (IntServ) and the Differentiated Services (Blake et al, 1998) (DiffServ) model.

The delay incurred on the end-to-end path in the network can be decomposed into a deterministic (e.g. propagation) and a stochastic (e.g. queuing) part. The deterministic delay is quite easy to assess, as it depends on factors such as the packetization delay, propagation delay and service delay. Queuing delay is more difficult to assess because it depends on the congestion state of the network. It is well known that deterministic upper bounds for queuing delay are easily obtained, see (Charny et al, 2000), (Parekh et al, 1993), (Parekh et al, 1994), but these worst-case upper bounds lead to unrealistically high values. Instead in this paper statistical “upper bounds” (i.e. quantiles) are derived for the queuing delay, leading to more realistic values, see also (Brichet et al, 1997), (De Vleeschauwer et al, 2000), (Mandjes et al, 1999).

Obviously there is a need to predict the delay before large-scale deployment of real-time services over IP networks. As these services are not implemented yet on a large scale, we follow a theoretical approach for assessing the delay performance that could be obtained under certain traffic assumptions.
Assessing the delay is important for two reasons. First of all in order to deliver voice with sufficient quality, strict delay bounds should be met. For example according to (ITU-T Rec. G.114, 1999) the end-to-end delay should not exceed 150 ms for telephony. In order to keep the queuing delay for real-time traffic small enough the load on the queues (devoted to real-time traffic) should be kept under control. The formulae developed in this paper are invaluable tools in network dimensioning and admission control mechanisms that are required to keep the load on those queues under control.

Secondly, in order to cope with variation in delay (often referred to as jitter) at the destination of every end-to-end route a dejittering buffer is implemented. The dejittering buffer retains the first arriving packet of a flow for some time (referred to as the dejittering delay) in the buffer before it starts to deliver the packets to the decoder at the rate with which the packet were originally generated. Packets that arrive too late in the buffer are discarded. Dimensioning the dejittering delay at the receiver side is of great importance. It involves a trade-off between delay and packet loss. The larger the dejittering delay is chosen the smaller is the amount of packets that arrive too late for play-out at the expense of more delay. When the dejittering delay is chosen equal to the $(1-P)$-quantile of the end-to-end queuing delay, the value of $P$ indicates the fraction of packets that can be expected to arrive too late for play-out, and hence, that are considered to be lost.

One traditional First-In-First-Out (FIFO) queue (per output interface) is not sufficient for supporting voice, video and data services on a single network, because data sources making use of the Transmission Control Protocol (TCP) tend to increase their sending rate until (a part of) the network is congested. Packets with real-time requirements should get priority over data traffic in order to minimize the queuing delay. This can be done, for example, with two queues (per output interface) served by a non-pre-emptive Head-of-Line (HoL) scheduler illustrated in Figure 1. Real-time packets get absolute priority over data packets. Data packets are only
served when the real-time queue is empty. In case DiffServ is used packets with different transport requirements can be distinguished (and classified into the correct queue) by the code-point in the header of the IP packet.

Figure 1

The payloads (i.e. voice or video code words) of the real-time services are transported over the Real-time Transport Protocol (RTP) over the User Datagram Protocol (UDP) over IP. Hence, the overhead for each packet consists of (12 byte for RTP, 8 byte for UDP and 20 byte for IP) 40 byte.

In this paper we study the real-time queue in isolation, because the queuing delay of the real-time services and the queuing delay due to the residual service of the data packet are assumed to be statistically independent (Kleinrock, 1975). We do not study the performance of TCP-controlled data.

In (De Vleeschauwer et al, 2000) and (Mandjes et al, 1999) different approaches have been introduced to determine a quantile of the queuing delay. The aim of this paper is threefold.

First of all, different approaches for determining quantiles of the queuing delay are discussed and compared.

Secondly, this paper includes the case where in each node the high-priority queue may have to treat a mix of several packet sizes. This corresponds with a situation where both voice and video are present with distinct packet sizes, hence, generalizing the results discussed in (De Vleeschauwer et al, 2000) and (Mandjes et al, 1999).
Finally, we also give results for networks where links have distinct capacities and/or loads, thus again generalizing the results discussed in (De Vleeschauwer et al, 2000) and (Mandjes et al, 1999).

The analytical solutions can be used to dimension dejittering buffers and to assess the capacity needed in each node for telephony and videophony services.

2. The queuing model

In order to calculate a quantile of the queuing delay for the real-time queue we assume that all voice and video flows are Constant Bit Rate (CBR) flows. However, as the traffic flows interfere with other flows, the CBR character of the flows disappears while traversing the network crossing multiple routers.

In (Brichet et al, 1997) it is conjectured that flows that are initially CBR cannot be disturbed to a stream that has more burstiness than a Poisson stream, as long as it interferes only with flows that were originally CBR. A recent paper (Bonal et al, 2001) presents analytical and simulation results that support the conjecture in (Brichet et al, 1997).

Hence, in order to calculate the delay we model each node as an M/G/1 queuing system. This also results in the convenient fact that each node $n$ can be characterized with a single parameter, namely the load $\rho_n$. Obviously, modeling an aggregate as a Poisson process is worst case.

Note that there is recent evidence that even TCP-controlled data traffic is also very well approximated by a Poisson arrival process if the load is high enough (Cao et al, 2001).

When assessing the queuing delay on the end-to-end path it is assumed that all nodes are statistically independent from each other. This can be justified by the fact that the aggregate traffic flow splits at each node and is mixed with traffic arriving from other input interfaces. The independence assumption has been validated in (Kruskal et al, 1984), (Lau et al, 1997) and
(Van Den Berg et al, 1995) and appears to be (for most cases) a worst case assumption, see also (Bonald et al, 2001).

Hence, to calculate the probability density function (pdf) of the end-to-end queuing delay the pdfs of the individual delays are to be convolved. From the resulting end-to-end delay pdf a small enough quantile has to be determined.

The theory presented here applies for arbitrary distributed packet sizes (i.e. M/G/1 model). However, the examples and results are worked out for a small number of different packet sizes, i.e. an M/(D_1+D_2+...+D_k)/1 model. Different packet sizes occur because voice and video sources use codecs with different bit rates and different values for the packetization delay are used. For the case of heterogeneous nodes results are shown for two types of nodes.

3. Approximating the (1-\(P\))-quantile of the total queuing delay

3.1. General

We define the moment generating function (mgf) of the random variable \(d\) as

\[
D(s) = E\{\exp(-sd)\}.
\]  

(1)

Because each node is modeled as an M/G/1 queue, the mgf of the delay in the \(n\)-th node loaded up to load \(\rho_n\) is given by (Kleinrock, 1975)

\[
D_n(s) = \frac{1 - \rho_n}{1 - \rho_n \left[ \frac{B_n(s) - 1}{B_n'(0)s} \right]},
\]  

(2)

where \(B_n(s)\) is the mgf and \(-B_n'(0)\) the average service time in the \(n\)-th node (with the prime \(^'\) indicating the derivative). Because the delays \(d_n\) incurred in all \(N\) nodes are assumed to be statistically independent, the mgf of the total queuing delay

\[
d = \sum_{n=1}^{N} d_n
\]  

(3)
is given by

\[ D(s) = \prod_{n=1}^{N} D_n(s) \quad . \quad (4) \]

In the following sections we discuss and compare different approximations for the computation of quantiles of the total queuing delay. Calculating the quantiles boils down to inverting the tail distribution function (tdf) of the total queuing delay, defined as

\[ T_d(x) = \text{Prob}[d > x] \quad , \quad (5) \]

i.e. the \((1-P)\)-quantile is the (smallest) value \(x\) for which \(T_d(x) \leq P\).

3.2. The approximation based on the dominant pole associated with each node

The approximation discussed in this section is based upon the assumption that the mgf of the delay in one node can be written as

\[ D_n(s) = \frac{H_n(s)}{s - p_n} \quad , \quad (6) \]

with \(p_n\) the root with smallest absolute value of the denominator of (2) and \(H_n(s)\) an analytical function in the neighborhood of the dominant pole \(p_n\). It follows from (2) that \(p_n\) satisfies:

\[ \rho_n (B_n (p_n) - 1) = p_n B_n \cdot (0) \quad . \quad (7) \]

We can approximate \(D_n(s)\) as

\[ D_n(s) \approx \lim_{s \to p_n} \frac{H_n(s)}{s - p_n} + C_n \quad , \quad (8) \]

where \(C_n\) is a constant chosen such that the right-hand side of (8) evaluates to 1 at \(s = 0\), which is mandatory because \(D_n(s)\) represents the mgf of a pdf.

An elementary calculation yields
\[ D_n(s) \approx 1 - R_n \frac{p_n R_n}{s - p_n} , \quad (9) \]

where \( R_n \) satisfies

\[ R_n = \frac{(1 - \rho_n)}{\rho_n \frac{B_n'(p_n)}{B_n'(0)} - 1} . \quad (10) \]

In (De Vleeschauwer et al, 2000) it is shown that \( p_n \) tends to \(-1/\sigma_n\) and \( R_n \) tends to 1, if the load \( \rho_n \) tends to 1, where \( \sigma_n \) is the standard deviation of the queuing delay in the node.

Since some of the \( N \) nodes may be identical the mgf of the total queuing delay can be written as

\[ D(s) = \prod_{l=1}^{L} \left[ 1 + \frac{(1 - R_l)\phi_l s}{1 + \phi_l s} \right]^{N_l} , \quad (11) \]

with \( \phi_n = -1/p_n \) and

\[ N = \sum_{l=1}^{L} N_l . \quad (12) \]

In the Appendix it is shown that the tdf \( T_d(x) \) associated with an mgf of this form is a weighted sum of tdfs of Erlang variables. To find the desired quantiles this tdf \( T_d(x) \) needs to be inverted.

In the next paragraphs the explicit formulae are given for the examples considered in Section 4.

### 3.2.1. \( N \) identical nodes serving a mix of different packet size

Let \( P_k \) denote the fraction of the packets with size \( S_k \), where \( k = 1, 2, \ldots, K \). Since all nodes are identical we omit the subscript \( n \) in this section. In this case eqs. (7) and (10) that define \( p \) and \( R \), become

\[ \rho \left( \sum_{k=1}^{K} P_k \exp(-pS_k) - 1 \right) = -p \sum_{k=1}^{K} P_k S_k , \quad (13) \]
\[ R = \frac{(1 - \rho)}{\sum_{k=1}^{K} P_k S_k \exp(-pS_k)} - 1 \] 

respectively. If there is only one packet size, these reduce to well-known formulae for the M/D/1 model already reported in (COST 224, 1991).

In order to determine the \((1-P)\)-quantile we need to invert the tdf \(T_d(x)\), which with the method described in the Appendix can be readily calculated as

\[ T_d(x) = \sum_{i=0}^{N-1} h[i] E_{N-i} \left( \frac{x}{\phi} \right), \] 

with

\[ h[i] = \frac{N!}{(N-i)! i!} (R)^{N-i} (1-R)^i, \]

and \(E_M(x)\) the tdf of an Erlang variable of \(N\) stages

\[ E_N(x) = \exp(-x) \sum_{n=0}^{N-1} \frac{x^n}{n!}. \]

### 3.2.2. \(N_1\) M/D/1 nodes under load \(\rho_1\) and \(N_2\) M/D/1 nodes under load \(\rho_2\)

In this case eqs. (7) and (10) become

\[ \rho_n (\exp(-p_n S_n) - 1) = -p_n S_n \] 

\[ R_n = \frac{(1 - \rho_n)}{\rho_n \exp(-p_n S_n) - 1}, \] 

with \(n = 1\) and 2 for the first and second group of nodes respectively.

In order to determine the \((1-P)\)-quantile we need to invert the tdf \(T_d(x)\), which with the method described in the Appendix can be readily calculated as
\[ T_d(x) = \sum_{i=0}^{N_i-1} h_i[i]E_{N_i-i} \left( \frac{x}{\phi} \right) + \sum_{i=0}^{N_{r}-1} h_{2}[i]E_{N_{r}-i} \left( \frac{x}{\phi_{2}} \right), \] (20)

with

\[ h_1 = f_{1,2} \otimes g_{1,1} \otimes g_{1,2}, \] (21)

\[ h_2 = f_{2,1} \otimes g_{2,1} \otimes g_{2,2}, \] (22)

\[ f_{k,j}[i] = \frac{(N_j -1 + i)!}{(N_j -1)!i!} \left( \frac{1}{1 - \phi_i} \right)^{N_j} \left( \frac{1}{1 - \phi_k} \right)^i, \quad k \neq l, \] (23)

and

\[ g_{k,i}[i] = \frac{N_i^{-1}}{(N_i -i)!i!} \left( 1 - (1 - R_l) \frac{\phi_i}{\phi_k} \right)^{N_i-i} \left( 1 - R_l \frac{\phi_i}{\phi_k} \right)^i, \quad i \leq N_i \] (24)

and \( \otimes \) denoting the convolution.

### 3.3. Gaussian approximation

Inspired by the central limit theorem the Gaussian approximation assumes that the distribution of the total queuing delay, which consists of \( N \) statistically independent queuing delays, is governed by a Gaussian distribution. A Gaussian distribution is completely determined by its first two central moments, which are given by

\[ \mu = \sum_{n=1}^{N} \mu_n, \] (25)

\[ \sigma = \sqrt{\sum_{n=1}^{N} \sigma_n^2}, \] (26)

where \( \mu_n \) and \( \sigma_n \) are the central moments of the queuing delay in the \( n \)-th node. These can be calculated using Takács recurrence relations for the moments (Kleinrock, 1975). For the \( n \)-th node we obtain
\[
\mu_n = \frac{\rho_n}{2(1-\rho_n)} \cdot \frac{E[S_n^2]}{E[S_n]} \cdot E[S_n] \quad ,
\]

and

\[
\sigma_n = \sqrt{\left(\frac{\rho_n}{2(1-\rho_n)} \cdot \frac{E[S_n^2]}{E[S_n]}\right)^2 + \frac{\rho_n}{3(1-\rho_n)} \cdot \frac{E[S_n^3]}{E[S_n]^3} \cdot E[S_n]} \quad ,
\]

where \(E[S_n^m]\) denotes the \(m\)-th moment of the service time of a packet at the \(n\)-th node.

The \((1-P)\)-quantile is then approximated by

\[
Q(P) \approx \mu + \text{erfc}^{-1}(P)\sigma \quad ,
\]

where \(\text{erfc}^{-1}\) is the inverse of the complementary error function (i.e., the tdf of a Gaussian variable)

\[
\text{erfc}(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{\tau^2}{2}\right) d\tau \quad .
\]

It is expected that the Gaussian approach will only form a good approximation if the number of nodes \(N\) is high enough.

**3.3.1. \(N\) identical nodes serving a mix of different packet size**

Again we drop the subscript \(n\) in (27) and (28), as all nodes are identical. The \(m\)-th moment of the service time can be calculated as

\[
E[S^m] = \sum_{k=1}^{K} p_k (S_k)^m \quad .
\]

Eq. (27) and (28) give the average and standard deviation of the queuing delay in each node. Using these individual contributions the central moments of the total queuing delay can be calculated with eq. (25) and (26) and the \((1-P)\)-quantile with eq. (29).
3.3.2. $N_1$ M/D/1 nodes under load $\rho_1$ and $N_2$ M/D/1 nodes under load $\rho_2$

The average and standard deviation of the total waiting time is given by

$$\mu = N_1\mu_1 + N_2\mu_2,$$

$$\sigma = \sqrt{N_1\sigma_1^2 + N_2\sigma_2^2},$$

respectively, where the individual averages and standard deviations are given by eq. (27) and (28), respectively, with $E[S_n^m] = E[S_n]^m$. Eq. (29) is then used to calculate the (1-P)-quantile.

3.4. Heuristic formula

The heuristic formula approximates the (1-P)-quantile similar to the Gaussian approach as the average of the total queuing delay plus a number of times the standard deviation of the total queuing delay, i.e.

$$Q(P) \approx \mu + \alpha(P, \sigma_1, \ldots, \sigma_N)\sigma,$$

with $\mu$ and $\sigma$ given by (25) and (26), but this time the weighting factor $\alpha$ is chosen such that, if the individual delays where exponentially distributed the formula would be exact. Hence, it follows that

$$\alpha(P, \sigma_1, \ldots, \sigma_N) = \frac{T^{-1}(P, \sigma_1, \ldots, \sigma_N) - \sum_{n=1}^{N} \sigma_n}{\sqrt{\sum_{n=1}^{N} \sigma_n^2}},$$

with $T^{-1}$ the inverse of the function $T$ defined as tdf of the sum of $N$ exponentially distributed variables each with their own average (and standard deviation) $\sigma_n$. Notice that not all the $\sigma$'s are necessarily different. This tdf $T$ can be calculated with the method discussed in the Appendix (with $R_l$ set equal to 1). So, in principle the weighting factor $\alpha$ could be tabulated once and for all. Remark that the weighting factor $\alpha$ does not need to be calculated for all possible combinations of $\sigma_n$ as
\[
\alpha(P, \vartheta \sigma_1, \ldots, \vartheta \sigma_N) = \alpha(P, \sigma_1, \ldots, \sigma_N) \quad \forall \vartheta > 0
\] (36)

Once the function \( \alpha \) is tabulated, the heuristic formula is closed-form.

Similar as in (De Vleeschauwer, 2000) it is expected that the heuristic formula (34) only will form a good approximation, if the load is high enough.

### 3.4.1. \(N \) identical nodes serving a mix of different packet size

The first two central moments are calculated as in Section 3.3. In (De Vleeschauwer, 2000) it is proven that if all nodes are identical, the weighting factor is given by

\[
\alpha \left( P, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}} \right) = \frac{E_N^{-1}(P) - N}{\sqrt{N}}.
\] (37)

### 3.4.2. \(N_1 \) M/D/1 nodes under load \( \rho_1 \) and \( N_2 \) M/D/1 nodes under load \( \rho_2 \)

The first two central moments are again calculated as in Section 3.3. In this case finding the weighting factor \( \alpha \) involves the inversion of

\[
T(x) = \sum_{i=0}^{N_1-1} h_1[i] E_{N_1-i} \left( \frac{x}{\sigma_1} \right) + \sum_{i=0}^{N_2-1} h_2[i] E_{N_2-i} \left( \frac{x}{\sigma_2} \right),
\] (38)

with

\[
h_1 = f_{1,2},
\] (39)

\[
h_2 = f_{2,1},
\] (40)

and

\[
f_{k,l}[i] = \frac{(N_l - 1 + i)!}{(N_l - 1)! i!} \cdot \frac{1}{\sigma_l} \cdot \frac{1}{\sigma_k} \cdot \left( 1 - \frac{\sigma_l}{\sigma_k} \right)^{N_l} \cdot \left( 1 - \frac{\sigma_k}{\sigma_l} \right)^i \quad \text{for} \quad k \neq l.
\] (41)
3.5. Method of Abate & Whitt

3.5.1. General methodology

Before we compare the different approximations to compute the dejittering delay, we first discuss how, starting with the Laplace transform of the queuing delay distribution given in (4), we can obtain \((1-P)\)-quantiles numerically by applying the inversion method for Laplace transforms by Abate & Whitt (Abate et al, 1992).

Let \(f(t)\) denote a continuous pdf, with Laplace transform \(F(s)\).

In (Abate et al, 1992) it is shown that for any constant \(a > 0\)

\[
f(t) = \frac{\exp\left(\frac{a}{2}\right)}{2t} F\left(\frac{a}{2t}\right) + \frac{\exp\left(\frac{a}{2}\right)}{t} \sum_{k=1}^{\infty} (-1)^k \Re e F\left(\frac{a + 2k\pi i}{2t}\right) - \varepsilon(a, t) \quad \forall t > 0,
\]

where \(\Re e(z)\) denotes the real part of the complex variable \(z\). Eq. (42) is based upon Poisson’s summation formula.

The error term \(\varepsilon(a, t)\) can be bounded by

\[
\varepsilon(a, t) \leq \frac{\exp(-a)}{1 - \exp(-a)} \quad \forall t > 0.
\]

Abate & Whitt propose to approximate (42) by

\[
s_n(t) = \frac{\exp\left(\frac{a}{2}\right)}{2t} F\left(\frac{a}{2t}\right) + \frac{\exp\left(\frac{a}{2}\right)}{t} \sum_{k=1}^{n} (-1)^k b_k(t) + E(m, n, t),
\]

where

\[
b_k(t) = \Re e F\left(\frac{a + 2k\pi i}{2t}\right),
\]

and \(E(m, n, t)\) denotes the Euler sum of the first \(m\) terms after an initial \(n\), i.e.

\[
E(m, n, t) = \sum_{k=0}^{m} \binom{m}{k} 2^{-m} s_{n+k}(t).
\]
The Abate & Whitt algorithm is completely specified by (44), (45) and (46).

3.5.2. Tuning the parameters in the method of Abate & Whitt

The Abate & Whitt algorithm contains three parameters, $a$, $n$ and $m$. We have tuned these parameters by comparing the outcome of the algorithm (implemented in a Matlab program) with the exact $(1-P)$-quantiles for convolutions of $M/M/1$ queues (implemented in a Maple program). For the network scenario we considered a network consisting of $N = 8$ nodes, loaded up to $\rho = 0.5$, with a link rate of $R_{\text{link}} = 34$ Mb/s and an average packet size of $S = 440$ byte.

We have chosen the following parameter values in the algorithm: $a = 25$, $n = 38$ and $m = 11$. The results are given in Figure 2, which depict the exact $(1-P)$-quantiles of the queuing delay and the ones calculated with the Abate & Whitt algorithm.

It can be concluded that with these settings for $a$, $m$ and $n$, and for the scenario described above, the relative error for $(1-P)$-quantiles is at most $2 \times 10^{-7}$ for $P \geq 10^{-6}$. However, for very small values of $P$ (not depicted in Figure 2) Abate & Whitt diverges enormously from the exact quantiles.

There are three sources of errors in the inversion algorithm: the discretization error $\epsilon(a,t)$, the truncation error by applying Euler summation and finally, the rounding error which result from subtracting positive numbers that are very close to each other.

Figure 2

The choice for $a = 25$ is based on the network scenario where $N = 1$, $\rho = 0.5$, $R_{\text{link}} = 2$ Mb/s and an average packet size of $S = 440$ byte and $P = 10^{-6}$. The result for this scenario is given in Figure 3.
An exact error bound for the Euler sum is in general difficult to give. It is suggested in the literature (Abate et al, 1992) to use $|E(m-1,n)-E(m,n)|$ as an estimate of the error caused by using the Euler sum, but for some cases this is a poor estimate, see the example in Section 11 of (Abate et al, 1992).

Throughout the remainder of this paper we will assume the parameter settings: $a = 25$, $n = 38$ and $m = 11$. Furthermore, the Abate & Whitt algorithm will be used as a benchmark for the methods described in Section 3.2, 3.3 and 3.4.

4. Results and discussion

4.1. Comparison of the statistical and the deterministic upper bound approach

It was stated in the introduction that the statistical upper bound estimation of the queuing delay is more appropriate for practical purposes than the worst-case upper bound estimation. In this section we validate this statement.

As an example we first consider a single M/D/1 node with $\rho = 0.8$, $R_{\text{link}} = 34$ Mb/s and a packet size of $S = 240$ byte.

The $(1-P)$-quantile of the queuing delay (in seconds) is given by

$$T_{\text{stat}} = q(\rho, P) \cdot T_{\text{ser}}$$

(47)

where $q(\rho, P)$ denotes the normalized $(1-P)$-quantile of the queuing delay expressed in number of packets and $T_{\text{ser}}$ the serialization delay, i.e. $T_{\text{ser}} = 8S/R_{\text{link}}$.

The worst-case upper bound for the queuing delay corresponds to the assumption that all sources feed a packet in the queue at precisely the same instant. In order to quantify this we
need to know the codec bit rate $R_{\text{cod}}$. Then the worst-case upper bound is proportional to the number of phones $M$ that can be simultaneously supported on a link of capacity $R_{\text{link}}$ loaded up to $\rho$

$$M = \rho \frac{R_{\text{link}}}{R_{\text{gross}}}$$

(48)

where $R_{\text{gross}} = R_{\text{cod}} S / (S - 40)$ denotes the gross bit rate of the codec, i.e. including the overhead of the 40 byte header. The worst-case upper bound (in seconds) now becomes

$$T_{\text{wc}} = M * T_{\text{ser}}.$$  

(49)

Here we consider codec bit rates of 64 kb/s (ITU-T Rec. G.711 format), 32 kb/s (ITU-T Rec. G.726 format), 16 kb/s (ITU-T Rec. G.726 format), 8 kb/s (ITU-T Rec. G.729 format) and 5.3 kb/s (ITU-T Rec. G.723 format). Table I and II give the quantiles calculated with the statistical approach and the worst-case upper bounds respectively. The quantiles in Table I are computed with the method of Abate & Whitt. Comparing the tables clearly indicates the difference between the statistical approach and the worst-case approach. To interpret what the (1-$10^{-6}$)-quantile means in practice, consider the example of $S = 240$ byte and $R_{\text{cod}} = 64$ kb/s. In this case if someone would talk over the phone continuously, then on average only 1 in $10^6$ packets, i.e., only 1 packet every 6.9 hours, would experience a delay of more than 1.79 ms. This indicates that the worst case bound of 20 ms is not very tight.

Note that if a route over several nodes is considered the worst-case upper bound becomes even less firm because for the statistical approach convolutions are used to calculate the quantiles whereas for the worst-case approach the contribution of all individual nodes are simply added.

The conclusion of the section is that for practical purposes statistical upper bounds for queuing delay are to be preferred over worst-case upper bounds.
### Table I: Quantiles of the M/D/1 node loaded up to 0.8 serving packet of size $S = 240$ bytes.

<table>
<thead>
<tr>
<th>$P$</th>
<th>Delay [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>0.58</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>0.89</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>1.19</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>1.49</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>1.79</td>
</tr>
</tbody>
</table>

### Table II: Worst-case upper bounds encountered in one node loaded up to 0.8 with codecs of bit rate $R_{cod}$ producing packets of size $S = 240$ bytes.

<table>
<thead>
<tr>
<th>$R_{cod}$ [kb/s]</th>
<th>Delay [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>20</td>
</tr>
<tr>
<td>32</td>
<td>40</td>
</tr>
<tr>
<td>16</td>
<td>80</td>
</tr>
<tr>
<td>8</td>
<td>160</td>
</tr>
<tr>
<td>5.3</td>
<td>242</td>
</tr>
</tbody>
</table>

4.2. $N$ Identical M/D$_1$+ ...+ D$_k$/1 nodes

In this and the next section we use the label “AW” for the method of Abate & Whitt, “HE” for the heuristic formula, “GA” for the Gaussian approximation and “DP” for the method based on the dominant pole in each node.

For each of the considered cases we present the $(1-P)$-quantile $q_#$ calculated with the four considered methods ($# = AW$, GA, DP or HE) and the relative error. As the AW method is used as benchmark the relative error (in percent) is defined as

$$100 \frac{q_# - q_{AW}}{q_{AW}}.$$  \hspace{1cm} (50)

We always consider the same link rate $R_{link} = 34$ Mb/s and the case where $N = 8$ nodes are traversed.

The packet mix consists of packets of $S_1 = 140$ byte, packets of $S_2 = 240$ byte, packets of $S_3 = 440$ byte and packets of $S_4 = 1500$ byte. The first three packet sizes result if a (voice) codec of bit rate $R_{cod}$ of 16 kb/s, 32 kb/s and 64 kb/s, respectively, uses a packetization interval
of 50 ms and an IP/UDP/RTP header of 40 byte is added. The latter packet size could stem e.g. from a video codec.

The GA method is the method that requires by far the least computation. The DP method is more complex than the HE method for two reasons. First, the DP method needs to determine $p$ and $R$ (see eq. (13) and (14)), where the former involves the solution of a transcendent equation, while the average and standard deviation of the total delay are readily calculated for the HE method. Second, the DP method requires the inversion of a tdf defined by eqs. (15) and (16), while the HE method is closed form (see eq. (37)). The AW method is the most complex method.

4.2.1. Influence of the load $\rho$

Figure 4 shows the 0.999-quantile ($P = 10^{-3}$) and the relative error as function of the load $\rho$. The considered packet mix consists of one third of packets of 140 byte, one third of packets of 240 byte and one third of packets of 440 byte (i.e. $p_1 = p_2 = p_3 = 1/3$, $S_1 = 140$ byte, $S_2 = 240$ byte, $S_3 = 440$ byte).

The GA method has the worst performance. Apparently $N = 8$ is not large enough to make the tdf Gaussian. This is consistent with the findings in (De Vleeschauwer et al, 2000).

For low loads the HE method (and to a lesser extent the DP method) do not give accurate results. For the HE method the absolute value of the relative error is smaller than 1 % if the load is larger than 0.7, while for the DP method it is already smaller than 1 % for a load larger then 0.5.

Figure 4
4.2.2. Influence of the parameter $P$

Figure 5 shows the $(1-P)$-quantile and the relative error as function of the parameter $P$. The considered packet mix is the same as in the previous paragraph (i.e. $p_1 = p_2 = p_3 = 1/3$, $S_1 = 140$ byte, $S_2 = 240$ byte, $S_3 = 440$ byte). The load $\rho$ on each node was set to 0.8.

Again it can be seen that the GA method heavily underestimates the $(1-P)$-quantile. The relative error of the DP and the HE method is smaller than 1 % for all values of $P$. Also it can be seen that for low values of $P$ the DP method performs slightly better than the HE method. For $P = 10^{-9}$ the $(1-P)$-quantile calculated with the AW method (used as benchmark) is no longer reliable (see Figure 2) which makes that the monotonous character of the relative error curves is broken there.

4.2.3. Influence of the packet mix

Figure 6 shows the influence of the traffic mix. The load is again $\rho = 0.8$. As before the mix consists of equal portions of 140 byte, 240 byte and 440 byte packets, but now there is also a fraction $\pi$ of 1500 byte packets, (i.e. $p_1 = p_2 = p_3 = (1-\pi)/3$, $p_4 = \pi$, $S_1 = 140$ byte, $S_2 = 240$ byte, $S_3 = 440$ byte, $S_4 = 1500$ byte). Figure 6 can e.g. be used to predict the influence of mixing real-time video traffic in a queue dedicated to real-time voice.

Again the GA method underestimates the $(1-P)$-quantile and the DP method slightly outperforms the HE method, although for the latter two the relative error is again smaller than 1 %.

Figure 6
4.3. $N_1$ M/D$_1$/1 nodes under load $\rho_1$ and $N_2$ M/D$_2$/1 nodes under load $\rho_2$

In this section all nodes are no longer identical. To limit the number of different parameters we consider the case where the nodes can be divided into two groups, with identical nodes in each group. Also for simplicity we only discuss cases where all packets are of the same size ($S = 440$ byte). We always consider the same link rate $R_{\text{link}} = 34$ Mb/s and the case where $N = 8$ nodes are traversed. But remember that the methodology developed in Section 3 is generally valid.

Again the GA method is by far the method that requires the least computation. The DP method is more complex than the HE method for two reasons. First, the DP method needs to determine $p_n$ and $R_n$ (see eq. (18) and (19)), the former involves the solution of transcendent equations, while the average and standard deviation of the total delay are readily calculated for the HE method. Second, compared to the HE method the DP method requires additional convolutions: compare eqs. (20), (21), (22), (23) and (24) with eqs. (38), (39), (40), and (41). The AW method is again the most complex method.

4.3.1. Variable load $\rho_2$

Figure 7 shows the influence of the load $\rho_2$ of the second group of nodes, while the load $\rho_1$ on the first group of nodes is fixed to 0.8. Again the GA method has a bad performance. Both the DP and the HE method have a relative error smaller than 1 %, even for small loads $\rho_2$ (a conclusion that is seemingly opposed to Section 4.2.1 where we concluded that the load should be high in order to have accurate results). Apparently it is sufficient that the load on one (or a few) of the nodes is sufficiently high ($\rho_1 = 0.8$ here) for the DP or the HE method to closely approximate the $(1-P)$-quantile. Comparing the DP and the HE method for this case we see that for small loads $\rho_2 < 0.7$ the HE method slightly outperforms the DP method for, while for larger loads the DP method is slightly better.
Figure 7

4.3.2. Variable number of nodes of one kind

Figure 8 shows the influence of the number of nodes that are under high load. There are two groups of nodes. All nodes in the first group are under load $\rho_1 = 0.8$, the nodes of the second groups under load $\rho_2 = 0.9$. The total number of nodes is always $N = 8$.

The conclusions here are similar to the ones drawn above. The GA method heavily underestimates the $(1-P)$-quantile and the DP and HE method approximate the quantile very well (i.e., with a relative error smaller than 1%).

Figure 8

4.4. Convergence

In (De Vleeschauwer et al, 2000) it is proven that the HE and the DP method produces quantiles that converge to the correct values as the load $\rho$ tends to 1 for the case of $N$ identical M/G/1 nodes. This proof can be readily extended to the case of two groups of identical nodes.

As for the accuracy of both methods the rule of thumb resulting from the examples shown in the previous sections is that the relative error of the DP method is smaller than 1%, if the load $\rho$ on one of the nodes is larger than 0.5, while for the HE method to obtain the same accuracy the load on one of the nodes needs to be larger than 0.7.

Notice that the HE and the GA method are the same except for the multiplication factor with which the standard deviation needs to be multiplied (see eq. (29), (34) and (35)). Figure 9 illustrates that the factor $\alpha_N(P)$ converges to $\text{erfc}^{-1}(P)$ as $N$ tends to infinity for the case of $N$
identical nodes. Hence, also the quantiles produced by the HE method converge to the ones produced by the GA method as \( N \) tends to infinity. Figure 10 shows for the case of \( N \) identical M/D/1 nodes how large \( N \) needs to be in order to attain an accuracy of at least 1%, i.e. for values smaller than the values given in Figure 10 there is a difference of at least 1% between the GA and the AW method. It is clear from this figure that \( N \) should be of the order of a few hundred to a few thousand, before the GA method produces accurate results.

Moreover, since the HE method converges to the exact value (i.e., also the one produced by the AW method if \( P \) is not too small) as the load \( \rho \) tends to 1 and since \( \alpha_{M}(P) \) considerably differs from \( \text{erfc}^{-1}(P) \) if \( N \) is too small (see Figure 9), the GA method cannot converge to the exact value as the load \( \rho \) tends to 1, if the number of nodes \( N \) is too small.

Measurements performed in (Vanhaestel et al., 1999) show that a route over the Internet rarely traverses more than 30 nodes. Hence, the GA method is not accurate enough for our purposes.

5. Conclusions

This paper has developed several methods to calculate quantiles of the total queuing delay incurred over a route through several packet-based nodes. All methods are based on two assumptions. First, the packet arrival process in each node is approximated by a Poisson arrival process. Second, the queuing delays in consecutive nodes are assumed to be statistically independent. All methods are general in the sense that each node can have its individual
characteristics, e.g. link rate, packet mix, and load, hence, generalizing the work in (De Vleeschauwer et al, 2000) and (Mandjes et al, 1999).

It was first shown that for practical purposes quantiles (i.e. statistical bounds) for the queuing delay are to be preferred over worst-case bounds.

Then the paper assessed the accuracy and complexity of the four methods.

The Gaussian method assumes that the total queuing delay through many nodes has a Gaussian distribution. This turned out to be too simplistic. The number of nodes traversed on a normal route is by no means large enough for the Gaussian method to produce accurate results.

The method of Abate & Whitt is based on the explicit inversion of the Laplace transform with a numerical method based on Poisson’s summation formula. It was concluded that this method works very well for \((1-P)\)-quantiles as long as \(P\) is not too small \((P > 10^{-7})\). Furthermore, the method of Abate & Whitt is of the considered methods by far the most complex and does not give insight in how the parameters, like load and packet mix, influence the total queuing delay.

The method based on the dominant poles and the heuristic method make use of the fact that the total queuing delay is the sum of individual queuing delays that have approximately an exponential tail distribution function. The former method calculates the rate of decay of the tail and the residue in an exact way, while the latter method approximates these by using the first two moments of the queuing delays. Hence, the heuristic method is less complex than the method based on the dominant poles. Both methods produce very accurate results, i.e. relative errors smaller than 1 \%, as soon as the load on at least one node is high enough. In particular the method based on the dominant poles has in a relative error smaller than 1 \%, if the load on at least one node is larger that 0.5, while for the heuristic formula the load on at least one node needs to larger than 0.7 to attain the same accuracy. Moreover, the heuristic formula and to a lesser extent also the method based on the dominant poles, explicitly show how the load and the packet mix influence the quantile of the total queuing delay.
Appendix

This appendix shows how to calculate the tdf $T_{d}(x)$ of the total queuing delay, if its mgf is of the form

$$D(s) = \prod_{l=1}^{L} \left[ \frac{1+(1-R_l)\phi_l s}{1+\phi_l s} \right]^{N_l}. \quad (51)$$

This kind of calculation is needed for the heuristic formula and in the approximation based on the dominant poles.

Let's first consider the contribution to the tdf of the $k$-th pole $-1/\phi_k$. Using the residue theorem and the substitution $s=s'/\phi_k$ this contribution can be written as

$$\lim_{s \to -1} \frac{d}{ds} \left( \prod_{l=1}^{L} \left[ \frac{1+(1-R_l)\phi_l s}{1+\phi_l s} \right]^{N_l} \right) \left( -\frac{1}{s} \right) \exp \left( s \frac{t}{\phi_k} \right). \quad (52)$$

Using

$$\frac{d^K}{ds^K} (fgh) = \sum_{i=0}^{K} \sum_{j=0}^{K-i} \frac{K!}{i! j!(K-i-j)!} \frac{d^i f}{ds^i} \frac{d^j g}{ds^j} \frac{d^{K-i-j} h}{ds^{K-i-j}}, \quad (53)$$

and

$$\frac{d^n}{ds^n} \left( \frac{-1}{s} \right) = n! \quad , \quad (54)$$

and

$$\frac{d^n}{ds^n} \left[ \exp \left( s \frac{t}{\phi_k} \right) \right]_{s=-1} = \left( \frac{t}{\phi_k} \right)^n \exp \left( -\frac{t}{\phi_k} \right) \quad , \quad (55)$$

the contribution of the $k$-th pole to the tdf can be written as
\[
\sum_{i=0}^{N-1} h_k[i] E_{N_k-i} \left( \frac{x}{\phi_k} \right),
\]

with \( E_N(x) \) the tdf of an Erlang variable of \( N \) stages

\[
E_N(x) = \exp(-x) \sum_{n=0}^{N-1} x^n n!
\]

and

\[
h_k[i] = \frac{1}{d^i} \frac{d^i}{d s^i} \left\{ \prod_{j=1}^{L} \frac{1}{s (1 + \phi_j s)} \prod_{l=1}^{L} \left[ 1 + (1 - R_l) \frac{\phi_l}{\phi_k} \right]^{N_l} \right\} \right|_{s=1}.
\]

Via its Z-transform it can be proved that \( h_k \) is a convolution

\[
h_k = \left( \bigotimes_{j=1}^{L} f_{k,j} \right) \otimes \left( \bigotimes_{i=1}^{L} g_{k,i} \right),
\]

with

\[
f_{k,j}[i] = \frac{(N_l-1+i)!}{(N_l-1)! (1-R_l)!} \left[ 1 \frac{1}{\phi_j \phi_k} \right]^{N_l-i} \left[ 1 - \frac{\phi_j}{\phi_k} \right]^i k \neq l,
\]

and

\[
g_{k,i}[i] = \frac{N_l!}{(N_l-i)! (1-R_l)!} \left( 1 - (1-R_l) \frac{\phi_l}{\phi_k} \right)^{N_l-i} \left( 1 - \frac{\phi_l}{\phi_k} \right)^i i \leq N_l.
\]

Collecting the contribution of all the poles, this proves that the tdf of the total queuing delay is a weighted sum of Erlang tdfs

\[
T_d(x) = \sum_{k=1}^{L} \sum_{i=0}^{N_k-1} h_k[i] E_{N_k-i} \left( \frac{x}{\phi_k} \right).
\]
Acknowledgement

This work was partly carried out within the framework of the project CoDiNet sponsored by the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT).

References


Captions

Figure 1: Head-of-Line queuing system.

Figure 2: Comparing Abate & Whitt (AW) with the exact solution for an 8-fold convolution of M/M/1 queues.

Figure 3: The influence of the parameter $a$ in the Abate & Whitt algorithm for a quantile in a M/M/1 queue.

Figure 4: $(1-P)$-quantiles calculated with the 4 methods (left) and relative error of the heuristic formula and the method based on the dominant poles with respect to the method of Abate & Whitt as a function of the load $\rho$. The parameters were $R_{\text{link}} = 34$ Mb/s, $N = 8$, $P = 10^{-3}$, $p_1 = p_2 = p_3 = 1/3$, $S_1 = 140$ byte, $S_2 = 240$ byte, $S_3 = 440$ byte.

Figure 5: $(1-P)$-quantiles calculated with the 4 methods (left) and relative error of the heuristic formula and the method based on the dominant poles with respect to the method of Abate & Whitt as a function of $P$. The parameters were $R_{\text{link}} = 34$ Mb/s, $\rho = 0.8$, $N = 8$, $p_1 = p_2 = p_3 = 1/3$, $S_1 = 140$ byte, $S_2 = 240$ byte, $S_3 = 440$ byte.

Figure 6: $(1-P)$-quantiles calculated with the 4 methods (left) and relative error of the heuristic formula and the method based on the dominant poles with respect to the method of Abate & Whitt for different packet mixes. The parameters were $R_{\text{link}} = 34$ Mb/s, $\rho = 0.8$, $N = 8$, $P = 10^{-3}$, $p_1 = p_2 = p_3 = (1-\pi)/3$, $p_4 = \pi$, $S_1 = 140$ byte, $S_2 = 240$ byte, $S_3 = 440$ byte, $S_4 = 1500$ byte.

Figure 7: $(1-P)$-quantiles calculated with the 4 methods (left) and relative error of the heuristic formula and the method based on the dominant poles with respect to the method of Abate & Whitt as function of the load on the second group of nodes. The parameters were $R_{\text{link}} = 34$ Mb/s, $P = 10^{-3}$, $\rho_1 = 0.8$, $N_1 = N_2 = 4$, $P = 10^{-3}$, $S = 440$ byte.
Figure 8: $(1-P)$-quantiles calculated with the 4 methods (left) and relative error of the heuristic formula and the method based on the dominant poles with respect to the method of Abate & Whitt as function of number of nodes under load $\rho_2 = 0.9$. The parameters were $R_{\text{link}} = 34$ Mb/s, $P = 10^{-3}$, $\rho_1 = 0.8$, $\rho_2 = 0.9$, $N_1 + N_2 = 8$, $P = 10^{-3}$, $S = 440$ byte.

Figure 9: The multiplication factor $\alpha_M(P)$ (solid lines) used in the HE method converges to the one $\text{erfc}^{-1}(P)$ (dashed lines) used in the GA method as the number of nodes $N$ increases.

Figure 10: The minimal number of nodes $N$ for which the relative error between the GA and the AW method is smaller than 1% for the case of $N$ identical M/D/1 nodes for two values of the parameter $P$. 
Figures

Figure 1

Figure 2

Figure 3

Figure 4
Figure 5

Figure 6

Figure 7

Figure 8
Figure 9

Figure 10