End-to-end delay models for interactive services on a large-scale IP network

Michel Mandjes ♠, Kees van der Wal ♠, Rob Kooij ♠, Harrie Bastiaansen ♠.

Abstract

In addition to the traditional best-effort Internet service, the Internet community now explores new Internet service models with guaranteed QoS levels. Work in this area has in particular been co-ordinated by the IETF working groups on Integrated Services (intserv) and on Differentiated Services (diffserv). The goal of these working groups is to define new QoS mechanisms for the Internet to be able to provide IP transport services with sufficient quality to support interactive services.

A key aspect is the end-to-end delay that may be expected from using these services. Because no widely deployed networks exist which provide these low-delay services, this article addresses models for assessing the major components in the end-to-end delay. The models can be applied to both the intserv and the diffserv approach.

An IP network configuration is considered consisting of a high-speed IP core network and two lower-speed access parts. For the core network queuing models are presented. In addition, it is concluded that, if some 'loose requirements' are fulfilled by the core network, the delay due to queuing can be kept below 8 ms; the propagation delay is to be added to this figure.

For the access parts, very simple models can be used if it is assumed that only a single interactive application uses the access. Simple expressions are presented covering various protocols (including PPP with suspend/resume) to be used on the access parts.

This paper has been presented at the 7th Workshop on Performance Modelling and Evaluation of ATM and IP Networks at the University of Antwerp, June 28-30, 1999.

The Working Conference was organised by IFIP TC6 (Communication Systems), Working Group 6.3 (Performance of Communication Systems) and Working Group 6.4 (High Performance Networks).

The proceedings are edited by Guido H. Petit.

♣ KPN Research, Sint Paulusstraat 4, 2264 XZ Leidschendam, the Netherlands. Fax: +31 70 332 6477. J.C.vanderWaal@research.kpn.com
♢ KPN Research, Groningen, the Netherlands. H.J.M.Bastiaansen@research.kpn.com
♠ Bell Labs/Lucent Technologies, Murray Hill, USA. Michel@research.bell-labs.com
1 Introduction

The Internet traditionally provides a service that is commonly characterised as a best-effort service. The network tries to deliver the IP datagrams at their destination but no guarantees are given. Many applications (e.g. web browsing) happily run using this traditional best-effort service model. Some areas of the Internet may be heavily congested and, consequently, a considerable fraction of datagrams is discarded by the network. Usually, additional higher layer protocols (e.g. TCP for error detection, retransmission and flow control) compensate for the lost datagrams. At the application level this is noticed as a reduced throughput.

However, not all applications are able to reduce the application rate to an arbitrary low level and can tolerate the delay due to retransmissions. For example, interactive communications such as interactive voice communication, video telephony or video conferencing would suffer from additional delay. Thus, even if the IP service is augmented with additional higher layer protocols, such as TCP, it cannot provide the application with guarantees for throughput and for a low transfer delay. Therefore, a new service model is necessary to meet the customer's demand for a consistent Quality of Service (QoS).

1.1 Guaranteed QoS

Work in this area has in particular been co-ordinated by the IETF working groups on Integrated Services (intserv) and on Differentiated Services (diffserv). The goal of these working groups is to define new QoS mechanisms for the Internet to be able to provide IP transport services with sufficient quality to support interactive services.

- The IETF intserv working group has defined two new Internet services: the Controlled Load Network Element Service [22] and the Guaranteed Quality of Service [23]. In particular the latter may be expected to provide strict end-to-end QoS guarantees in terms of throughput, data loss, delay and delay variation. The services may be invoked by using the RSVP protocol [20,21,29].

- The IETF diffserv working group is defining a service architecture based on a simpler model where traffic entering a network is conditioned at the edges of the network, and assigned to different behaviour aggregates [25]. Each behaviour aggregate is identified with a combination of bits, currently called the DS (Differential Service) code point [24]. Within the core of the network, packets are forwarded according to the per-hop behaviour associated with the DS code point. In particular the Expedited Forwarding behaviour [27] is expected to provide a 'guaranteed' throughput, low data loss, and minimum delay and delay variation.

1.2 End-to-end delay assessment

For interactive applications the foremost question is “How good can these new services expected to be?”, in particular in terms of end-to-end delay. As these services are not yet implemented on a wide scale, we follow a theoretical approach for assessing the delay performance that could ideally be obtained under certain traffic assumptions. Earlier work [28] showed that, to support interactive applications, a sufficiently low delay can be achieved if some conditions are met. The conditions relate to the link rate, the load on each link and the size of the IP datagrams using the guaranteed service. Such conditions can be enforced with the approach outlined by the intserv working group. It also showed that the link rate used on the access has a major effect on the delay and may even jeopardise the interactivity of some applications.
This paper provides a calculation model which enables to assess the end-to-end delay in a large-scale IP network. The approach is similar to but more comprehensive than the earlier work [28]. The calculation model is insensitive to whether the intserv or the diffserv approach is used; it is sufficiently generic to cover the delay performance achieved by either of these proposed approaches.

1.3 Structure of this paper

The structure of this article is as follows. Section 2 describes several scenarios and underlying assumptions, separately for the core network, the access network and the applications.

Section 3 targets the actual models for the delay performance. In particular, Section 3.2 provides the theoretical foundation and the practical application to arrive at a small but realistic upper bound for the queuing delay experienced in the core network. Section 3.3 similarly provides the calculation models for the delay in the access parts. Details are given for the calculations applied for several protocol options used in the access.

Section 4 summarises the main conclusions.
2 Scenarios and assumptions

The delay performance of new Internet service models is assessed for an IP network configuration consisting of an access network and an IP core network as illustrated in Figure 1.

An IP datagram is generated by an application. Such applications may, for example, be interactive voice and interactive video (see Section 2.3 for more details about the applications). The IP datagram, travelling from source to destination, traverses only two hops on low speed links; a single hop between the source and the core network (upstream) and a single hop from the core network to the destination (downstream). In between, it is assumed that there is a high speed core network with multiple router hops. The assumptions on the core network are listed in Section 2.1; those for the access parts in Section 2.2. In addition to the IP scenarios, also a full ATM scenario is evaluated where the native ATM application produces ATM cells and where both the core and the access network carry ATM cells.

2.1 Core network

The current Internet has a distributed organisation; no single provider is responsible for its operation. Providing a hard end-to-end delay guarantee in such an environment is a real challenge. Here, we argue that a ‘good’ delay performance can be obtained from a concatenation of independent networks (providers) if each satisfies some ‘loose requirements’. It is assumed that these requirements are fulfilled and that there is a high-speed core network with the following properties.

- The IP routers are interconnected with a capacity of STM-1 (155 Mbit/s) or higher.
- The core network can be traversed in no more than 15 hops.
- The IP datagrams in the core network are no larger than 1500 byte.
- The routers are able to distinguish between IP datagrams from interactive applications (priority traffic) and from other applications (best-effort traffic).
- The load in the core network generated by the interactive applications is not more than 60% of the capacity between the routers. The remainder of the capacity may be saturated with best-effort traffic.

Each of these assumptions is briefly elaborated on below.

2.1.1 At least STM-1 capacity between routers

It is assumed that the high-speed IP backbone is specifically designed for high throughput. A capacity equivalent to STM-1 (or OC-3) provides a gross bit rate of 155.52 Mbit/s; a net capacity of 149.74 Mbit/s is available to the routers. With the advent of gigabit routers with even higher
interface rates, this appears to be a realistic assumption. Using a lower interconnecting capacity (e.g. 34 or 45 Mbit/s) has been shown [28] to yield significantly more delay.

### 2.1.2 Number of hops in the core network

It is assumed that the high-speed IP backbone is specifically designed for low delay by reducing the number of routers to a minimum required to implement the network. A provider may choose to implement a network architecture with two levels of hierarchy. Crossing a single provider’s network appears to be achievable in at maximum 5 hops: twice a single hop at the lower hierarchical level and 3 hops at the top level assuming that the routers at the top level are nearly meshed. The networks of all co-operating providers are interconnected. If the interconnection between any of two co-operating providers can be done via a single (third) transit network, no IP datagram needs to traverse more than three networks and no more than 5 hops in each network, resulting in a maximum of 15 hops.

### 2.1.3 Maximum size of IP datagrams

It is assumed that IP datagrams are not larger than 1500 byte. This equally applies to the best-effort and to the priority datagrams. Because Ethernet allows no larger packets, a limit of 1500 byte is not an additional restriction as long as it remains a popular protocol. The assumption matches quite well with the packet size observed in the current Internet where datagram sizes beyond 1500 byte are rare.

### 2.1.4 Distinction between priority and best-effort traffic

It is assumed that the network is able to distinguish between one class of IP datagrams to be treated as best-effort and another class of datagrams to be treated with preference. The latter class is typically used by applications with stringent delay requirements. For the calculation model it is of no relevance whether the distinction is based on state information (for example, resulting from RSVP messages) or from a Differentiated Services code point carried in the IP header. Consequently, the calculation model applies to both the intserv and to the diffserv approach (see Section 1.1).

### 2.1.5 Load from interactive applications

It is assumed that, even on the long term, a significant part of the network capacity is consumed by best-effort traffic. With a successful introduction of new services it is expected that today’s elastic applications, such as email, file transfer and web browsing, will remain and will co-exist with priority traffic from interactive and other delay-sensitive applications. Further observing the current trend that ‘data traffic’ is growing faster than ‘telephony traffic’, it is expected that the average demand from priority traffic will not exceed half the available link capacity. To allow for some statistical variations in the momentary load, it is assumed that the momentary load of priority traffic on any core network link may be slightly higher than 50%, say 60%. The remainder of the link capacity is available for best-effort traffic and it is assumed that there may be sufficient best-effort traffic to saturate the links.

In case the diffserv approach (see Section 1.1) is used, network dimensioning is expected to assure that the average load of priority traffic during busy hour remains below half the interconnection capacity. In case the intserv approach is used, additional reservation requests may be denied (blocked) if the momentary aggregated reserved capacity (priority traffic) exceeds 60% of the interconnecting capacity.

### 2.1.6 Comparison with a native ATM environment

In the comparison with a full ATM case, the same assumptions are made.
2.2 Access network

The access part of each route is assumed to consist of two low-speed hops: a single hop from the source to the core network (upstream) and a single hop from the core network to the destination; see Figure 1. The options assessed for the access part are to be divided according to the access rate and the protocol used on the access.

2.2.1 Access rate

The following technologies are believed to be relevant for the access part:

- ISDN with a capacity of 64 kbit/s in both the up- and downstream part;
- ADSL with a capacity of 640 kbit/s in the upstream part and 3 Mbit/s downstream. Because the use of Forward Error Correction is known to introduce an additional delay (latency) of several tens of ms, this option is not considered.
- T1/E1 with a capacity of about 1.5 Mbit/s (T1) or 2 Mbit/s (E1) in both access parts.

These transmission rates in the access parts are very low compared to the ≥155 Mbit/s rate in the high-speed IP backbone. The possibilities listed above are expected to be representative for residential and for small- and medium enterprise customers.

2.2.2 Access protocol

Regarding the protocols used on the access parts, the following options are considered: full IP, two PPP options and IP over ATM in two variants: on the access only and end-to-end.

- Full IP. The most straightforward option is to transmit IP datagrams using separate queues for best-effort datagrams and for priority datagrams. The priority queue is served with priority over the best-effort queue but once transmission of a best-effort datagram has commenced, this is not interrupted if a priority datagram arrives.

- PPP options. It has been identified that the presence of large (best-effort) datagrams on low-rate links jeopardises the delay objectives for interactive applications [2]. For example, a 1500 byte best-effort IP datagram on an 64 kbit/s access link makes this link unavailable for the transmission of real-time information for 187.5 ms. Thus, a full IP solution as described above has a significant drawback. Therefore, the IETF Integrated Services over Specific Link Layers (issll) working group is defining mechanisms to resolve this issue. Two approaches are further explored in the IETF and included in this assessment.

  - A fragment-oriented solution for PPP packets has been described in [3]. In this methodology, large IP datagrams are broken up into smaller fragments and queued in a separate best-effort queue. The datagrams from delay-sensitive applications are queued separately and served with priority over the best-effort fragments. Hence, datagrams from the priority queue can be interleaved between fragments of the best-effort datagrams. In this solution the (maximum) size of the fragments remains a parameter which value needs to be selected.

  - A suspend/resume-oriented solution for best-effort packets is proposed to be added to the PPP multilink fragmentation protocol [4]. In this methodology, transmission of a best-effort datagram is suspended (pre-empted) on arrival of a priority datagram. The pre-emption may be done after any byte of the best-effort datagram by sending an escape code followed by a (compact) header and the priority data. After serving the priority datagram, transmission of the remainder of the best-effort datagram is resumed.

In both cases the fragments or the suspended frames are re-assembled at the receiving end of the access link. Hence, the high-speed IP backbone operates on full-size IP datagrams and not on fragments.
• **IP over ATM on the access.** Using ATM under IP, the IP datagram is converted into a sequence of ATM cells using AAL type 5. A sequence of cells carrying a best-effort datagram may be interrupted at any cell boundary by a sequence of cells carrying a priority datagram. At the receiving end of the access, the IP datagrams are reassembled from the ATM cells.

• **IP over ATM end-to-end.** In case IP over ATM is used end-to-end the IP datagrams are carried in ATM cells as described above. However, the ATM cells are not reassembled at the receiving end of the access link but carried end-to-end over an ATM network and reassembled at the receiving terminal.

### 2.3 Applications

For simplicity, it is assumed here that at the user terminal, there is a **single real-time application and one or more other applications that use the traditional best-effort service.** For a business environment, the model needs to be expanded to cover multiple real-time applications, but this is not treated here. Even for a residential user the assumption about a single real-time application is disputable. In this paper the assumption serves as a starting point for the calculations.

#### 2.3.1 Interactive applications

Two examples for interactive applications are given here: voice over IP and a video application. With a single interactive application per user terminal, the interactive application is assumed to be either of the following two example applications.

- **A voice over IP application at 32 kbit/s.** This is used as an example of an interactive application producing a relatively low bit rate. Several standardised and proprietary coding techniques do exist for such applications that produce a lower information rate than standard telephony at 64 kbit/s PCM, but maintain a reasonable quality, for example 32 kbit/s ADPCM. With more advanced compression and coding techniques the source bit rate may be further reduced towards, 16 kbit/s or 8 kbit/s. The possibility of silence detection has not been included in the calculations. Thus, each voice source is assumed to produce a continuous bit stream at the given rate which is packetised into equally sized IP datagrams (20 byte header) using RTP (12 byte) and UDP (8 byte). The IP datagram size (payload size) used for voice is one of the design parameters.

- **A video over IP application at 384 kbit/s.** This is used as an example of an interactive application producing a relatively high bit rate. With an information rate of 384 kbit/s, a good quality video telephony or video conferencing can be achieved. The payload, containing both the video and the voice component of the application, is assumed to be packed into equally sized IP datagrams with the same 40 byte overhead as specified above for voice. The IP datagram size used for video is one of the design parameters; the size may be different from the size for voice datagrams.

For interactive applications the delay should be kept to a low value. Applications with much interactivity are known to be the most demanding but there is no hard boundary between acceptable and unacceptable delays. An end-to-end delay of less than about 100 ms is known to provide a good communication quality for even the most demanding interactive tasks [13,15]. Therefore, we assume an objective to restrict the end-to-end delay to 100 ms, but we allow users to set their own objective depending on the application or on user preferences.

#### 2.3.2 Best-effort applications

It is assumed that, in addition to the interactive application, there is always at least one application that uses the traditional IP best-effort service, for example a data application. The datagrams generated by these applications are assumed to be no larger than 1500 byte, a rather common MTU size in the current Internet. As the applications use the best-effort service, there is no control over the momentary load of the data produced; the load generated by these applications may saturate all available capacity.
2.3.3 Native ATM applications

In case the applications use native ATM, the continuous bit stream is mapped into ATM cells using AAL type 1 with a 47 octet payload. Cells are assumed to be completely filled; partially filling the cell payload or the use of AAL type 2 has not been considered in this paper.
End-to-end delay models for interactive services on a large-scale IP network
3 Models for delay performance

This section presents the models used to assess the delay performance. Separate models are used for the high-speed IP backbone (the core network), and for the access parts. In addition, the delays introduced by the application itself need to be accounted for, which is not covered in this paper.

First, a pre-amble (Section 3.1) addresses deterministic bounds versus statistical bounds. Then, Section 3.2 quantifies the delay components of the core network. In addition, it shows that eventually a single value may be used as an upper bound to the core network delay. Note that a single application cannot autonomously influence this delay component.

Then, Section 3.3 derives straightforward upper bounds for the delay components of the access parts. Note that these delay components may be influenced by the user, for example by choosing the access rate, the access protocol and the application’s packet size.

3.1 Pre-amble: deterministic or statistical bounds on queuing delay?

The existing literature on performance studies of packet-switched networks (in particular end-to-end delays) can be roughly divided into two proposed approaches.

- In the first place, deterministic upper bounds on the end-to-end delay have been derived; the key reference here are the seminal articles by Parekh and Gallager [17,18], related work is by Chang [6], Cruz [9], and Le Boudec [16]. This type of delay bounds are based on ‘worst-case scenarios’: it corresponds to the delay of the last packet to be served when all streams feed a packet into the queue at precisely the same instant. If many streams are multiplexed, it becomes unlikely that all (unsynchronised) sources offer a packet at a same instant; it is more likely that the packet arrivals are randomly ’spread’ over time. Consequently, the actual delay experienced by an arbitrary packet is smaller than the deterministic upper bound.

- In particular in case a large number of flows are multiplexed, statistical upper bounds were proposed to provide more realistic values for the delay. There, the stochastic mechanism behind packet arrivals at the queues is explicitly modelled. The end-to-end delay is the sum of the individual delays at the hops. The 10^-6 quantile of the distribution of this end-to-end delay is typically regarded as ‘statistical upper bound’, see [28]. A prerequisite, however, is knowledge on the change in burstiness of the traffic streams during their paths through the network. In [28] it was assumed that the traffic streams hardly changed while traversing a number of hops; Briche, Massoulié and Roberts [5] postulate the conjecture that under specific conditions the delay at any individual queue is bounded by the delay in an appropriate queue with Poisson arrivals and deterministic service (M/D/1).

In this paper, we advocate using statistical bounds for the core network, as deterministic bounds will give unrealistically high values. More particularly, as will be explained in Section 3.2, we will use the M/D/1-based upper bound proposed in [5]. On the access parts only a limited number of traffic flows is present simultaneously. Consequently, we will rely on deterministic bounds for these parts.

3.2 Method and calculations applied to the queuing delay in the core network

This subsection deals with the queuing delay experienced by a priority datagram in the core network. First, Section 3.2.1, describes the architecture of the routers. These architecture allows us to split up the delay into a part due to collision with other priority streams and a part due to best-
effort traffic; statistical upper bounds to these delay components are given in Sections 3.2.2 and 3.2.3, respectively. An upper bound on the aggregated core delay is provided in Section 3.2.4. Section 3.2.6 gives the comparison with ATM.

### 3.2.1 Components of the queuing delay at the routers

In every router, packets from different inputs contend for the same output. As only a single packet at a time can be handled, other packets have to be queued. This results in a stochastic queuing delay. There are two reasons for queuing delay.

- A priority packet might have to wait for other priority packets to be served. The priority datagrams are handled in a FIFO manner.
- A priority packet might have to wait for non-priority (best-effort) traffic being in service. We assume no pre-emption, i.e., the server is available for priority datagrams as soon as the in-service best-effort packet has departed. The value of this delay component is not more than the maximum size of a best-effort packet (1500 byte, see Section 2.1.3) divided by the link rate.

A priority stream enters the network as continuous stream of packets (CBR), but the stream is disturbed due to interference with best-effort streams, which temporarily occupy the server (the lack of pre-emption, as described above). It is straightforward to see that the delay for a priority datagram at a particular router is bounded by:

- the delay the (disturbed) priority stream would experience in the model with pre-emption at that router (this delay will be referred to as the priority delay in the remainder of the paper),
- increased by the service time of a best-effort packet.

For that reason, we will first consider in Section 3.2.2 the model that a priority stream arrives at a router and does not see the best-effort at all (i.e., there is strict pre-emption). Then, we will treat the queuing delay due to the best-effort traffic, and show that we can do slightly better than taking just the service time of the best-effort packet. This delay component is evaluated in Section 3.2.3.

![Figure 2. The router output buffer architecture.](image)

### 3.2.2 Priority delay

This subsection deals with the calculation of the priority delay. We will justify the use of a convolution of M/D/1 delays as upper bound.

#### 3.2.2.1 Disturbance of the CBR streams.

We assume (see Section 2.3.1) that the traffic from interactive applications is (almost) CBR at the entrance of the network. The queuing in the first queue of the series can consequently be described by the model N×D/D/1 (assuming that all priority streams have the same bit rate). However, as the traffic streams interfere with other streams, the CBR character of the streams will disappear while traversing the network crossing multiple routers. Two extreme views on this phenomenon are possible.

- We can assume – as we did in our earlier work [28] – that the CBR character of the streams is hardly affected. Then, the end-to-end delay can be calculated as the convolution of the
delays of the individual $N \times D/D/1$ queues, assuming that these individual delays are independent. Notice that this approach is optimistic. In reality the traffic streams will be disturbed, and thus induce a higher delay than calculated based on $N \times D/D/1$.

- On the other hand, we can assume that the traffic streams are ‘maximally disturbed’. In [5], Brichet, Massoulié, and Roberts present support for the conjecture that streams that were initially (almost) CBR, cannot be disturbed to a stream that has more burstiness than a Poissonian stream, as long as it interferes only with streams that were originally (almost) CBR. In other words, the conjecture states that the delay in any individual queue is bounded by the delay of the M/D/1 queue with corresponding offered load.

In our queuing model the interfering stream cannot be characterised as a stream that was originally (almost) CBR, as it is triggered by the idle periods of the priority traffic. However, notice that the time it takes to serve a best-effort packet is maximally 80.13 µs (on a 155 Mbit/s link), while even for small priority packets the inter arrival times are orders of magnitude larger: a 100 bytes packet gives inter arrival times of about 40 ms for voice and 3.3 ms for video. Consequently, we believe that the resulting priority streams will have at most Poissonian burstiness.

We expect the M/D/1 approach to be pessimistic. In reality the traffic streams will be less disturbed and thus induce a lower delay than calculated based on M/D/1.

Therefore, we will consider the following two specific cases: the individual queuing delays corresponding to (1) an $N \times D/D/1$ queue and (2) an M/D/1 queue. The end-to-end queuing delay experienced by an IP packet can be seen as the sum of the individual queuing delays, caused by the consecutive routers. This is equivalent to calculating a convolution, assuming the individual delays being mutually independent. This independence assumption can be justified by reasoning analogously to Kleinrock [14].

### 3.2.2.2 Calculation of convolutions of M/D/1 and $N \times D/D/1$, Bahadur-Rao estimate

Convolutions can be performed by standard techniques. In case of ‘M/D/1-delays’, reflections on this convolution are given by Gravey, Romoeuf, Sevilla, and Blaabjerg [11]. As this solution is only valid in the case of heavy load, we propose a different technique in this subsection.

First note that, we are in fact not interested in the entire distribution of the end-to-end delay, but rather in its $10^{-6}$ quantile, i.e., that value of the end-to-end delay for which it holds that only one out of a million packets will experience a longer queuing delay. This quantile value can be regarded as a kind of ‘statistical upper bound’ to the end-to-end queuing delay. As this quantile refers to a rare event probability, the suitable technique here is to use large deviations theory, in particular the Bahadur-Rao estimate [1].

Let $W_i$ denote the queuing delay in a single queue, and let $d$ be the number of queues the path consists of. Then our goal is to find the quantile value $x$ such that

$$P\{W > x\} = \varepsilon, \quad (1)$$

with $W = \sum_{i=1}^{d} W_i$, and $\varepsilon$ typically $10^{-6}$.

Bahadur and Rao [1] present an efficient and accurate method to estimate this kind of rare event probabilities for the case that the random variables $W_i$ are independent and identically distributed. More concretely, let $M(\theta)$ be the *moment generating function* of the waiting time in one single queue:

$$M(\theta) = \int e^{\theta x} dF(x) = \int e^{\theta x} dP\{W_i > x\}.$$  

Here $F(x)$ is the distribution function of $W_i$. Suppose that this moment generating function exists. In other words: the queue length distributions does not have a ‘heavy tail’, which is indeed the case for both $N \times D/D/1$ and M/D/1. Define also the *large deviations rate function* as
End-to-end delay models for interactive services on a large-scale IP network

\[ I(x) = d \cdot \sup_{\theta} \left( \theta \cdot x - \log M(\theta) \right) = d \cdot \left( \theta^* \cdot x - \log M(\theta^*) \right), \]

where \( \theta^* \) is the (unique) root of

\[ \frac{M'(\theta)}{M(\theta)} = x. \]

The large deviations rate function \( I(x) \) measures the ‘likelihood’ of the end-to-end delay having value \( d \cdot x \). Inserting the mean end-to-end queuing delay, the function \( I(x) \) will give value 0; for unlikely large and small values of \( d \cdot x \), the function gives high positive values. Furthermore, define the tilted variance by

\[ \sigma^2 = d \cdot \left( \frac{M'(\theta^*)}{M(\theta^*)} \right)^2 - d \cdot \left( \frac{M(\theta^*)}{M(\theta^*)} \right)^2 \]

Having introduced this notation, we have the following accurate estimate for the tail distribution of the end-to-end delay:

\[ P(W > x) \approx \frac{1}{\sigma \sqrt{2\pi}} e^{-I(x)}, \quad (2) \]

As we are pursuing a statistical upper bound, we are interested in the value of \( x \) for which expression (2) equals \( \varepsilon \) of expression (1). Due to monotonicity properties of the right hand side, this can easily be done by a bisection procedure.

To apply the above formulas, the moment generating function \( M(\theta) \), and its first and second derivative have to be calculated. This requires the density \( dP(W_i > x) \) of the individual delay distributions \( W_i \), as we showed above. This density is known more or less explicitly for \( M/D/1 \).

However, for \( N \times D/D/1 \) only the distribution function is known. That problem can be circumvented as follows.

- In the \( N \times D/D/1 \) queue, \( N \) independent periodic sources (with the same period \( D \)) feed into a queue that is emptied at constant rate \( c \). Notice that – applying a rescaling – we can choose \( c = 1 \).

For \( N \times D/D/1 \) we only have a distribution function available, see [8, p. 118]; as we said, we lack an explicit formula for the density. The ‘complementary distribution function’ is given by

\[ 1 - F(x) = \sum_{n \leq N} \left( \begin{array}{c} N \\ n \end{array} \right) \left( \frac{n-x}{D} \right)^n \left( \frac{1-n-x}{D} \right)^{N-n} \frac{D-N+x}{D-n+x}. \]

A numerical value for the moment generating function in \( \theta \) can be found as follows from \( (1 - F(x)) \):

\[ M(\theta) = \int_0^\infty e^{x\theta} dF(x) = -\int_0^\infty e^{x\theta} d(1-F(x)) = 1 + \int_0^\infty e^{x\theta} \cdot \theta \cdot (1-F(x)) dx, \]

where the last step is done through integration by parts. This results in an integral that can be easily evaluated numerically, for instance by trapezoidal techniques. In the same way we obtain the following expression for the first and second derivative phrased in terms of the complementary distribution function \( (1 - F(x)) \).

\[ M'(\theta) = \int_0^\infty e^{x\theta} (1+x\theta) \cdot (1-F(x)) dx, \quad \text{and} \]

7th IFIP workshop June 28-30, 1999, Antwerp Belgium
In the M/D/1 queue, arrivals are Poisson, and services take a constant time. Now, load $\rho$ and link rate $c$ are the input parameters. Again, we can rescale to $c = 1$. Notice that the delay in the M/D/1 queue is the limit of the delay in the $N\times$D/D/1 queue, where $N$ grows large, and the load is held fixed, see [8, pp. 113-114]. Moreover, the delays in the $N\times$D/D/1 approach the delays in the M/D/1 from below in this asymptotic regime. In other words: the delay quantiles derived from the M/D/1 queue are an upper bound to the delay quantiles derived from the $N\times$D/D/1 queue.

Exact methods to determine the queuing time distribution of the M/D/1 queue are available, but usually lead to numerical problems. Therefore, an easy and explicit approximation [8] is used which has proven to be very accurate.

$$1 - F(x) = C_0 e^{-r_0 x},$$

(3)

where $C_0$ and $r_0$ are the solution to:

$$\rho \left( e^{r_0} - 1 \right) - r_0 = 0 \quad \text{and} \quad C_0 = \frac{1 - \rho}{e^{r_0} \rho - 1}.$$  

From (3) explicit expressions are obtained for the moment generating function and its derivatives. Note that the point mass in $x = 0$ has to be included separately in $M(\theta)$ since $F(0)$ should equal $1 - C_0$. This gives

$$M(\theta) = 1 - C_0 + \frac{r_0 C_0}{r_0 - \theta}, \quad M'(\theta) = \frac{r_0 C_0}{(r_0 - \theta)^2}, \quad M''(\theta) = \frac{2 r_0 C_0}{(r_0 - \theta)^3}, \quad \text{for } \theta \in [0, r_0).$$

According to Petrov [19], a similar procedure can also be applied if the $W_i$ do not have identical distributions. Thus, for a network with a concatenation of links with different capacities (e.g., 34, 155 and 622 Mbit/s) the approach is still valid. In this paper this refinement is not required as all links in the core network are assumed to have the same rate of 155 Mbit/s.

### 3.2.2.3 Choice between the models M/D/1 and N×D/D/1

We found that $N\times$D/D/1 queues give an optimistic impression of the delay quantile under investigation, while the M/D/1 based value tends to be pessimistic. We also demonstrated how to calculate the convolution of these delays. This enables us to make the following choice in favour of the model M/D/1. We start at the assumptions given in Section 2.1 and then make the following steps:

- **Assume in the calculations that all packets are 1500 byte.** In IP networks – unlike ATM networks – packets may have different sizes. We assume that there is a 1500 byte limit on the packet size. For a network with given load (load being defined as load caused by payload and header), from delay point of view it is worst case if all packets have the maximum size. To safely estimate the delay, we therefore assume all packets (payload plus header) to be 1500 byte.

- **Compare the queuing models M/D/1 and N×D/D/1.** First suppose homogeneous input: the network’s only interactive application is voice. A link loaded with 60% of priority traffic and assuming this load to include the headers, means that there are

$$\frac{149.76 \cdot 10^6}{32 \cdot 10^3 \cdot (1500 / 1460)} = 2733$$

voice flows simultaneously present. If the input of priority traffic were exclusively 384 kbit/s video streams, there are 227 video flows (cf. Section 2.1.5 on the 60%
load limit). If homogeneous priority traffic (i.e., all traffic being either voice or video) is offered, then the queuing delay experienced by a particular priority flow due to other priority flows can be described by either the convolution of $N \times D/D/1$ queues (optimistic) or $M/D/1$ queues (pessimistic), as argued in Section 3.2.2.2. First we will investigate whether the upper and lower bound are far apart for these homogeneous scenarios.

We already know [8, pp. 117-119] that the queue length distribution of one a single $N \times D/D/1$ queue converges to the queue length distribution of an $M/D/1$ queue, as the number of connections $N$ goes to $\infty$ while the load remains fixed. Figure 3 shows that also the quantiles of the 15-fold convolution of $N \times D/D/1$ queues converges to the 15-fold convolution of $M/D/1$ queues. Notice in addition that the convergence is rather fast because the load is relatively low (60%).

![Figure 3. The $10^{-6}$ and $10^{-3}$ quantiles of the queuing delay [ms] after 15 hops as a function of the number of priority streams and a load of 60%.](image)

From Figure 3, we can also see that, if all priority traffic were video (corresponding with $N = 227$), the $10^{-6}$ delay quantile has a value 2.86 ms. If all priority traffic were voice ($N = 2733$) this would result in a delay of 2.99 ms. A heterogeneous scenario with any mix of voice and video traffic resulting in a 60% load, will have a quantile in between those values.

It is illustrative to compare this value with the deterministic upper bound [17,18]. For a single hop the deterministic worst-case queuing delay calculates to 18.1 ms and 218.9 ms for video and voice respectively. These values are multiplied by the number of hops to result in unrealistic high values: 0.27 s (video) and 3.28 s (voice).

- **Use the queuing model M/D/1.** We can conclude that it appears both reasonable and practical to rely on the M/D/1 value.

  - It is reasonable since the M/D/1 value is a rather tight upper bound to the corresponding $N \times D/D/1$ values corresponding to the homogeneous scenarios, as we see from Figure 3. Notice that these M/D/1 and $N \times D/D/1$ delays are close due to the fact that the load is relatively low (60%); for higher loads, e.g., 90%-95%, the differences between the quantiles (based on $N \times D/D/1$ and M/D/1, respectively) are still considerable.
It is practical since the M/D/1 value is an upper bound characterised by only a single parameter, namely the offered load ($\rho$), caused by all interactive applications together. The upper bound does not depend on the number of flows. In addition, the upper bound applies to any heterogeneous mixture of flows (resulting in a load $\rho$) where each source generates an approximately regular stream of packets possibly different from the rate or any other source.

Having decided to use the M/D/1 queue to model the delay experienced in a single queue, Figure 4 can be generated. For various values of the (10^-5 quantile of the) queuing delay in the core network, the maximum allowable load as a function of the number of hops is given.

![Figure 4. The allowable load [% of 155 Mbit/s link capacity] for priority traffic, as a function of the number of hops, for different values of the maximum queuing delay.](image)

We conclude that, for the conditions in the high-speed IP backbone (see Section 2.1: 155 Mbit/s links, 60% load, 15 hops) the end-to-end queuing delay due to priority traffic is no more than 3 ms. In addition, it illustrates that the effect of a few hops beyond the target maximum of 15 does not lead to a dramatically higher delay. Moreover, it is compensated if the momentary load is slightly lower than the target maximum of 60%.

### 3.2.3 Queuing delay due to non-priority traffic

We assume in the core network that an in-service low-priority packet is not pre-empted. Consequently, for the priority traffic there is also a queuing component due to collision with non-priority streams, bounded by the time it takes to serve one non-priority packet:

$$\frac{1500 \cdot 8}{149.76 \cdot 10^5} \text{ s} = 80.13 \mu s.$$

As we assumed that there were at most 15 routers in a path, the total delay due to non-pre-emption is bounded by $15 \times 80.13 \mu s = 1.2$ ms.

This deterministic upper bound is rather crude because it is unlikely that in each queue this worst-case delay is experienced. A simple refinement can be made as follows, taking into account the part of the low-priority packet that was already served. Taking the priority streams Poissonian, the model with priority as well as best-effort traffic reduces to the M/D/1 model with service interruptions: as soon as the server gets idle and no priority packet is queued, it starts serving a best-effort datagram (consequently being unavailable during a deterministic time of 80.13 $\mu$s).
The M/D/1 model with service interruptions is studied extensively in the literature, e.g., by Cooper [7] and Fuhrmann [10]. Their most significant result is the following decomposition. The waiting time of an arbitrary packet can be written as the sum of two independent random variables: the waiting time in the ordinary M/D/1 and the residual duration of the service interruption. In fact, the former component is dealt with above, the latter is (per queue) distributed uniformly on the packet service time. The $10^{-6}$ quantile of this delay component over 15 hops is again calculated via the Bahadur-Rao estimate (again assuming independence between the routers). This yields a slightly tighter bound of $12.4 \times 80.13 \mu s \approx 1$ ms.

Figure 5 shows both the deterministic and a statistical upper bound ($10^{-6}$ quantile, 100% best-effort load) for this delay component.

3.2.4 Other components of the core delay

Routing time per router. Even in the absence of other traffic, and hence zero queuing delay, a packet experiences some delay in the router. It is assumed that this component is less than a few hundred $\mu$s. Therefore, this factor is not taken into account for the calculations.

Propagation delay in the core network. The propagation delay is 5 $\mu$s per kilometre. In case of a regional scenario of 250 km, this delay is 1.25 ms; for a world-wide scenario, spanning a distance of 10 000 km, the propagation delay is 50 ms.

Jitter compensation delay. It is assumed that all information is to be converted back into a constant bit stream. As the components in the queuing delay may vary from packet to packet, short delays have to be compensated by additional delay. Ideally, this compensation does not contribute to the worst case end-to-end delay. This ideal is not easy to achieve. A simpler method is to delay the first information by a sufficient amount to prevent buffer starvation. This delay should be at least the maximum value of the variable part of the delay. For the calculation of the jitter compensation delay, the same value is used as the total variable part of the delay, i.e., the queuing delays calculated above (due to the priority stream colliding with both other priority streams and non-priority streams). Notice that this jitter compensation does not actually take place in the core network (but rather at the receiver), but we incorporate it here as it directly relates to the (variable part of) the delay incurred in the core; there will also be a jitter compensation component in the access.
3.2.5 Aggregated delay due to the core network

The models presented in Section 3.2.2 and 3.2.3 allow to determine an upper bound to the delay due to priority traffic and best-effort traffic respectively. Section 3.2.4 reasons that the stochastic (variable) delay is to be compensated. A straightforward approach is to add the calculated upper bounds and to multiply the result by 2. Given the network parameters for the core network as listed in Section 2.1, we conclude that an upper bound to the $10^{-6}$ quantile of the delay in the core network is:

$$2 \times (3 \text{ ms} + 1 \text{ ms}) + \text{propagation delay} = 8 \text{ ms} + \text{propagation delay}.$$  

Note that there is room for a slight improvement by, rather than taking the sum of both individual quantiles as we did above, applying convolution to the probability density functions of the delay due to the priority traffic and that of the best-effort traffic.

3.2.6 Comparison to an ATM core network

As comparison for the IP results presented above, we derive the $10^{-6}$ quantile of the core delay for a full ATM scenario. For the ATM environment there are fewer variables: ATM uses fixed length cells. Using the same techniques as above, we find a $10^{-6}$ quantile of the M/D/1 queuing delay due to priority traffic that amounts to 0.11 ms, while the worst-case queuing delay due to non-priority traffic results in 43 µs. The total calculated queuing delay thus is about 0.15 ms; more than an order of magnitude smaller than the 3 ms value specified by ITU-T for the 2-point CDV in the stringent QoS class 1 [12]. Including the jitter compensation, this results in approximately 0.3 ms for the end-to-end delay due to queuing in the ATM core network.

3.3 Method and calculations applied to the delay in the access parts

For the access parts, the following components are considered in the delay calculations: the queuing delay, the jitter compensation delay and the input delay.

**Queuing delay.** In the router or user terminal where traffic from different sources at the customer premises is aggregated, queuing delay can occur, as only a single packet can be handled at a time. We will assume that at the customers premises only a single application generates priority datagrams. The maximum queuing delay depends on the access protocol used and is further elaborated on in the separate subsections. Note that the queuing delay in the access part occurs both at the upstream and at the downstream access part.

**Jitter compensation delay.** This type of delay is also discussed in Section 3.2. Similarly, this delay is set equal to the maximum value of the variable part of the delay, i.e., the maximum queuing delay.

**Input delay.** It is assumed that only complete IP datagrams are handled by the router i.e., no cut-through switching. The delay equals the IP datagram size divided by the rate of the relevant link:

$$\text{delay}_{\text{access transmission}} \ [\text{ms}] = P_{\text{gross}} \cdot \left(\frac{8}{r_{\text{up}}} + \frac{8}{r_{\text{down}}} \right),$$  

where

- $P_{\text{gross}}$: the overall IP datagram size [byte];
- $r_{\text{up}}$: uplink rate in kbit/s;
- $r_{\text{down}}$: downlink rate in kbit/s.

In case of a high rate in the access parts, this delay is negligible.

For each of the access protocols (see Section 2.2.2) simple expressions are given to calculate the aggregate of the delay components in the access parts.
3.3.1 Full IP (no pre-emption or fragmentation)

In case full IP is used, best-effort packets (≤1500 byte) are not pre-empted or fragmented. Thus, the aggregate of the delay components incurred in the access part is given by:

\[
\text{delay}_{\text{access,appl}} [\text{ms}] = 2 \cdot MTU \cdot \left( \frac{8}{r_{\text{up}}} + \frac{8}{r_{\text{down}}} \right),
\]

where \(MTU\) is the maximal size of a best-effort packet (here: 1500 byte). The factor 2 corresponds to adding the jitter compensation delay.

Values considered for the parameters \(r_{\text{up}}\) and \(r_{\text{down}}\) are given in Section 2.2.1.

- For 64 kbit/s access links, the delay induced by the access parts results in 750 ms, which is unacceptably high for interactive applications.
- For ADSL access, the access delay components result in 45.5 ms, which is substantial compared to the delay budget of an interactive application.
- For an access with T1 or E1 rate, the result is 32 ms and 25 ms respectively.

3.3.2 IP with PPP multilink fragmentation

If the fragment-oriented solution for PPP is used [3], the maximum queuing delay is the delay that corresponds to the time it takes to process a maximum-sized fragment. Expression (4) can be used where the maximum size of a fragment is used instead of the MTU value. It is easily seen that a smaller maximum fragment size leads to proportionally better values for this delay component compared to the full IP case (see Section 3.3.1).

3.3.3 IP with pre-emption

If the suspend/resume-oriented solution for PPP is used [4], a best-effort datagram can be interrupted (pre-empted) and suspended almost immediately. In the calculations this has been evaluated as a delay corresponding to 3 byte: one to account for the current byte of the best-effort datagram that cannot be pre-empted, one for the fragment suspend escape byte [4] and one for the compact fragment header format.

3.3.4 IP over ATM

The concept of using fixed size ATM cells on the access is very similar to using PPP with fragments (see 3.3.2) the size of an ATM cell. The calculations have been made assuming a fragment size of 53 byte.

In case IP over ATM is used end-to-end and in the full ATM case, the same values are used.

3.4 Method and calculation of delays incurred by the source

Independent of the technology used in the core and access network, some delay components are determined by the source; the \textit{coding delay} and the \textit{packetisation delay}.

\textit{Coding delay}. The time it takes from the moment the video (light) or sound (pressure wave) hits the transducer and the moment that the corresponding bits leave the coder. At the receiving end the reverse process may similarly incur some decoding delay. Because this paper deals with end-to-end delays, the term coding delay will be used to indicate the aggregate of both the coding and the decoding delay. For PCM and ADPCM voice codecs this delay is usually less than 1 ms. For the ADPCM examples it is regarded negligible. More recent codings, such as various CELP variants, allow lower bit rates but require 10 ms to 30 ms for its coding. Video coders usually show more coding delay, at least in the order of several tens of ms to more than 100 ms. Because the coding delay is more related to the type of coder used than to the Internet service, it is not taken into
account for the network-related calculations but remains to be added to the end-to-end application delay.

*Packetisation delay.* The time to fill an IP packet. It is assumed that the source itself produces a constant bit stream. Then, the packetising delay equals the IP packet payload size divided by the source information rate. The packetisation delay occurs only once. This results in:

\[
\text{delay}_{\text{packetisation}} [\text{ms}] = \left( \frac{8}{r_{\text{appl}}} \right) \cdot P_{\text{appl}}
\]

where

\[r_{\text{appl}}: \text{rate of application in kbit/s};\]
\[P_{\text{appl}}: \text{payload of IP datagram in byte}.\]

Note that for more advanced coding techniques the constant bit rate assumption no longer holds.
4 Summary and conclusions

As has been argued in the introduction, before implementing a large-scale IP network with new IP services which provide throughput and delay guarantees, it is of crucial importance to assess whether a delay can be achieved that is sufficiently low for interactive applications. The following detailed conclusions can be drawn.

- **Methodology.** It has been shown that it is beneficial to separate the calculations for the core network from the calculations for the access parts and in the source. We use deterministic bounds for the delay in the access parts and statistical bounds for the queuing delay in the core network.

- **Core network delay contribution.** It has been assumed that there is a high-speed IP backbone with two priority classes: priority and best-effort. Priority datagrams in this core network experience delay from other datagrams (priority and best-effort).
  - It is concluded that if the load of priority traffic is not too high (e.g. less than 60% of the link capacity) the contribution to the delay due to contention with other priority traffic is suitably modelled by a series of M/D/1 queues in tandem. A graph (Figure 4) shows the contribution to the end-to-end delay depending on the number of routers traversed and on the maximum link load.
  - It is concluded that the contribution to the delay due to contention with best-effort traffic is suitably modelled by a uniformly distributed delay depending on the maximum size of the best-effort datagrams.
  - Given some assumptions regarding the design parameters of the core network (155 Mbit/s links, ≤60% link load with priority traffic, ≤15 hops, see Section 2.1), the model allows to calculate a universal and small upper bound for the delay of priority datagrams in the core network: 8 ms plus the propagation delay.
  - From this small delay bound calculated for the IP core network it is subsequently concluded that introducing layer 2 switching techniques, for example MultiProtocol Layer Switching (MPLS) [26], is expected to yield no more than a marginal improvement in the end-to-end delay.
  - The calculated upper bound applies to both the intserv and the diffserv approach.
  - Applying the same model to a full ATM core network under comparable conditions, shows that the contribution of such a core network to the end-to-end delay is less than 0.3 ms. This is significantly less than for an IP network.

- **Access part delay contribution.** It has been assumed that only a single real-time application is active on the user’s terminal and access part. Straightforward expressions have been derived to establish an upper bound to the delay induced by the access parts. Different protocol options have been included: full IP, PPP with fragmentation, PPP with suspend/resume.

  An extension of the model is necessary to cover situations with multiple real-time applications originating and/or terminating at the user terminal.
- **Application delay contribution.** For sources generating a low rate continuous bit stream, the packetisation delay is a significant contribution to the end-to-end delay which is straightforward to calculate.

- **End-to-end delay estimate.** An estimate of the upper bound to the end-to-end delay is obtained by adding the contributions from the core network, the access parts and the application itself. The core network contribution can be assessed to a single value due to queuing plus the propagation delay. The application and access parts are to a large extend proportional to the size of the IP packet selected (either to the payload size or to the overall size). Hence, a trade-off can be made between:
  - a low end-to-end delay combined with a low efficiency (small packet payload) and
  - a larger end-to-end delay combined with a better efficiency (larger packet payload).

The approach and the underlying models allow to set an end-to-end application delay objective and to allocate parts of that delay budget to the contributing parts. For example, it can be regarded as a set of tools which can be used when choosing values of application parameters. Examples are the determination of the optimal IP payload size observing the end-to-end delay objective or considerations regarding a trade-off between applying more source rate compression which reduces the source bit rate but may result in additional coding delay.
References


[21] RFC 2210: *The Use of RSVP with IETF Integrated Services*; Wroclawski, J.; September 1997 (proposed standard)

[23] RFC 2212: Specification of Guaranteed Quality of Service; Shenker, S; Partridge, C; Guérin, R; September 1997. (proposed standard).

[24] RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers; Nichols, K; Blake, S; Baker, F; Black, D; December 1998.


