*Appl. Math. and Comp. Sci., 1992, vol.2, No.2, 171–197*

# ANALYTICAL MODELS FOR FILE TRANSFERS IN PACKET–SWITCHED COMPUTER NETWORKS

R. FEIX*, B. WOLFINGER*

**One of the most important application–oriented services in computer networks is file transfer and one of the most important performance characteristics from a user's point of view is delay (e.g. for data transfer requests). This makes delay analyses for file transfer highly desirable. In this paper, we present an approximative analytical model which makes it possible to calculate file delays in packet–switched data networks as well as in interconnected LAN–WAN–networks. Besides homogeneous applications of file transfer type, the models take into account mixed traffic situations, where file transfer is overlayed by some interactive traffic of higher priority. Starting with a model for a transient first moment analysis, steady–state investigations are carried out thereafter. Model validation results indicate the high degree of accuracy of the models for a variety of system configurations and load situations.**

## 1. Introduction

One of the most important application–oriented services in computer networks is file transfer (FT) service. Consequently, performance evaluation for this type of service is of considerable interest. However, despite this demand, there is still a lack of analytical models which reflect FT traffic adequately. The difficulty of obtaining suitable FT models seems to be a consequence of the fact that FT (and more general bulk data transfer) yields to burst arrivals which indeed represents a quite nasty arrival pattern for queueing theorists. (As is common, *bulk data* here corresponds to some large quantity of data).

Models which take into account some kind of burst arrivals have been presented e.g. in (Rubin, 1974; Rubin, 1975) for the case of message transfer via message–switched networks or in (Evequoz and Tropper, 1987; Wolfinger, 1986) for the case of FT via packet–switched networks. Bursty traffic modeling for local–area networks is partially covered by Spirn *et al.*, (1984). The present paper considerably generalizes the results published by Wolfinger, (1986). Moreover, as opposed to earlier studies, we will not restrict ourselves to the steady state case only, but carry out transient model analyses, too.

In this paper we consider a class of computer network configurations and services of the following properties:

* Computer Science Department, University of Hamburg, Vogt-Kölln-Straße 30, D–2000 Hamburg 54, E-mail: feix@rz.informatik.uni-hamburg.dpb.de, wolfinger@rz.informatik.uni-hamburg.dpb.de

a) Data transfer between endsystems is accomplished by an (irregularly) meshed packet–switched network consisting of a set of switching nodes connected by a set of links.
b) Communicating application processes exist in endsystems exchanging some kind of bulk data (e.g. relatively large files in the case of FT, programs and input data in the case of remote job entry, or texts in the case of electronic mail).
c) For transfer of bulk data, sessions are established between application processes on endsystems in such a way that each session is permanently mapped on a fixed path.
d) Bulk data is segmented into data units of limited length (which we call *messages*; alternatively we could denote such fragments as *packets*). Messages of a single session are transfered via a fixed path to their addressed endsystem where they are reassembled. (Note that there is no reassembly in intermediate switching nodes as would be the case in a traditional message–switched network).

We shall assume that the influence of hop level (link level) flow control, (cf. Gerla and Kleinrock, 1980), can be neglected. In addition, we suppose that the exchange of control information (such as acknowledgement) within Data Link and Network Layer (cf. e.g. Schwartz, 1987) shall be neglectable, too. It is evident that our assumptions are satisfied particularly well by packet–switched networks with fixed routing between endsystems or with unlayered virtual circuits (cf. Lampson *et al.*, 1981) between communicating processes in endsystems. (For a more detailed discussion of network characteristics assumed, cf. section 5).

The paper is organized as follows. The basic queueing model used to determine FT delays will be specified and analyzed in section 2. Model analysis directly yields to an algorithm to calculate mean transfer delay for a sequence of files exchanged within a given session. On one hand, the basic model is specialized (cf. section 2.5) for the cases of deterministic service times (i.e. constant message lengths) and of exponential service times. On the other hand, the basic model is generalized (cf. section 2.5 and 4) in order to take into account mixed traffic resulting e.g. from FT and from dialog users, the latter having higher priority for their data transfer. The results of several series of simulation experiments are used (cf. section 3) to validate the analytical models presented in section 2. Eventually, areas of applicability as well as limits in use for the analytical models are indicated (cf. section 5).

## 2. A Queueing Model for the Approximative Analysis of FT Delays

### 2.1. Preliminaries

As our bulk transfer analysis is mainly based on the notion of *unfinished work*, *virtual waiting time*, and *virtual delay* we will shortly summarize here the definition of these terms and reformulate a well–known formula for their computation. It should be noted that all of the results given in this section support transient model analysis as well as general service time distributions, as is required in the following.

Consider an M/G/1/FCFS queueing system fed by a Poisson input stream with rate $\lambda$. The customers' service times are assumed to be independent and identically distributed with probability distribution function (PDF) $B(t)$ and first and second moment $T$ and $T^{(2)}$, respectively. The ***unfinished work*** $v$ at time $t$ is the server's workload at time $t$, i. e. the amount of work (in units of time) needed to empty the queueing system, when no new customers are permitted to enter the queue after time $t$. For an FCFS–service–strategy the unfinished work at $t$ and the ***virtual waiting time*** at $t$ (i.e. a customer's waiting time if he would have arrived at $t$) are the same. Let the service process start at $t_0 = 0$ with an unfinished work of $v = v_0$ time units. A formula for the expected virtual waiting time at $t$ valid for arbitrary rates $\lambda$ and $T^{-1}$ can then be given as follows (Cohen, 1982)

$$W(\lambda, T, v; t) := v + \int_I \pi(v, u)\, du - (1 - \lambda T)t \qquad (1)$$

where $I := I(v,t)$ is the interval $[v,t]$, and $\pi(v,u) = Pr(v_u = 0|v)$ denotes the probability that a customer arriving at time $u$ need not queue given a workload of $v$ time units at $t_0$, so that the integral expression in (1) defines the queueing system's expected idle time accumulated during $[0,t]$. Due to Benes and Cohen it holds (Benes, 1957; Cohen, 1982)

$$\pi(v,t) = \sum_{n\geq 0} e^{-\lambda t}(\lambda t)^n/n! \int_J (t-u)/t\, dB^{n^*}(u) \quad (t > v) \qquad (2)$$

where $J := J(v,t)$ is the interval $[0, t-v]$, $B^{n^*}(\cdot)$ denotes the $n$–fold convolution of the service time PDF with itself, and $B^{0^*}(\cdot)$ the unit step function at zero. Of course, $\pi(v,t) = 0$ for $t \leq v$. Rather than (1) we shall use the expected ***virtual delay*** at $t$

$$V(\lambda, T, v; t) := W(\lambda, T, v; t) + T. \qquad (3)$$

### 2.2. Model Assumptions

Before elaborating a model for the analysis of bulk transfer delays we have to specify our model assumptions and introduce necessary definitions and notations.

**Computer network topology.** Let a *computer network* CN be a tuple $(\boldsymbol{N}, \boldsymbol{L})$, where $\boldsymbol{N} = \{N_1, ..., N_n\}$ is a set of computer network nodes and $\boldsymbol{L} = \{L_1, ..., L_l\}$ a set of (unidirectional) links. Thus, one bidirectional link of the real computer network is represented by two countercurrent links in the model.

**File transfer sessions.** A constant number of FT sessions is supposed for each source–destination–pair $(N_i, N_j)$, $i \neq j$. This assumption implies that sessions either last permanently or if a session terminates, it is immediately replaced by an equivalent session of the same load characteristics.

**Paths.** The given computer network comprises a set $\boldsymbol{P} = \{P_1, ..., P_p\}$ of paths. Formally a *path* in CN is an ordered sequence

$P = (N_{i,1}, L_{i,1}, N_{i,2}, L_{i,2}, ..., L_{i,k}, N_{i,k+1})$

where $N_{i,j} \in \boldsymbol{N}$, $L_{i,j} \in \boldsymbol{L}$ and $N_{i,r} \neq N_{i,s}$ for $r \neq s$. The routing problem is supposed to be solved, i.e., each session is already mapped on a fixed path $P \in \boldsymbol{P}$.

**Resources modeled.** Only links (transmission lines) are considered as resources in the basic model described. The reader will notice, however, that the model can be applied in a straightforward way to take into account the processing of messages by nodes, too. In any case, buffers of switching nodes are not considered to represent some limiting resources of the communication system.

**Load.** For a FT session $S$ mapped onto path $P = (N_i, ..., N_j)$ it is assumed that:

(a1) $S$ generates files with a constant arrival rate $\lambda_f = \lambda_f(S)$.
(a2) Each file f of session $S$ is segmented into say $\nu = \nu(f, S)$ messages; on the average a file of session $S$ creates $\overline{\nu}$ messages $m_1, m_2, ..., m_{\overline{\nu}}$.
(a3) All messages generated within session $S$ are transmitted via path $P$ independently. Messages belonging to the same file are reassembled at the destination node $N_j$.
(a4) All messages of one given file $f$ (created by $S$) are passed to the communication system with rate $\lambda_m >> \lambda_f$ (see Figure 1.).

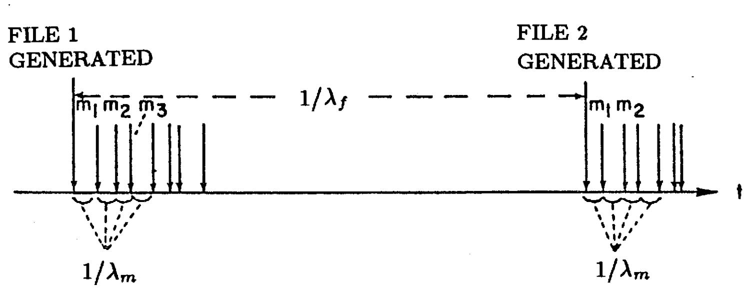



Fig. 1. Generation of files and messages during session $S$.

(a5) At each link on $P$ the arrival stream of messages not created by $S$ is (approximately) Poisson.

The assumptions concerning the load which is imposed to the network by the FT sessions imply that, in particular, each session $S$ induces a long–term load $\lambda_{\text{eff}} = \lambda_{\text{eff}}(S)$ equal to

$$\lambda_{\text{eff}}(S) = \lambda_f(S)[file/sec] \times \overline{\nu}(S)[msg/file] = \lambda_f \times \overline{\nu}[msg/sec] \quad (4)$$

To complete our list of load assumptions we have to make sure that for each link on $P$ the load rate cannot exceed the link's service capacity so that we claim for each $L \in \boldsymbol{L}$:

(a6) $\sum_{S \in S_L} \lambda_{\text{eff}}(S) T_L < 1$, where $S_L$ denotes the set of all sessions transmitting their files via a path containing $L$, and $T_L$ is the mean service (transmission) time at link $L$ (cf. Kleinrock, 1976).

**Remark.** Assumption (a5) is weaker than the message independence assumption (Kleinrock, 1976). We do not claim that messages belonging to the same file $f$ created by $S$ arrive at intermediate nodes of $P$ according to a Poisson stream. Only the interfering traffic is supposed to form a Poisson stream, and this will be a realistic assumption at least in cases where the interference is a result of superposition of traffic from a large set of independent sources, *feeding in* their traffic via several different links. Of course, any fluctuations in traffic of some sources are typically smoothed, anyway, along its path through a meshed communication system.

### 2.3. Model Elaboration and Analysis

Having discussed the basic model assumptions now, the next step towards the elaboration of a suitable FT model is the analysis of the typical dynamic behavior of a FT. We restrict the model to take into account the data exchange phase of a transfer, thus not describing the potential exchange of attributes at the beginning of the transfer and also neglecting possible termination actions at the end. Consider an arbitrary (*tagged*) session $S$ using path $P = (N_1, L_1, N_2, L_2, ..., N_k, L_k, N_{k+1})$ of length $k$ with sending node $N_1$ and receiving node $N_{k+1}$.

We first need some further notations summarized in the table below:

| | |
|---|---|
| $T_i$ | – mean service time at link $L_i$ |
| $T_i^{(2)}$ | – second moment of service time at $L_i$ |
| $\lambda_i^{\text{ext}}$ | – arrival rate of *external* messages (i.e. messages not created by $S$) at link $L_i$ |
| $d_f$ | – average transfer time for file $f$ |
| $\Delta_{i,r}(f)$ | – average interdeparture time at $L_i$ of consecutive messages $m_r, m_{r+1}$ belonging to file $f$ |
| $\delta_i(f)$ | – average interarrival time at $L_i$ between the last message of $f$ and the first of the succeeding file |
| $d_{i,r}(f)$ | – average delay for the $r-$th message of file $f$ at $L_i$ |
| $d_i(f)$ | – average delay for the first message of $f$ at $L_i (= d_{i,1}(f))$. |

Figure 2 shows an example of a typical FT via a three–hop path. Interfering traffic is not depicted in this example. The assumptions we made about FT sessions suggest the following model. The path used by a transfer session is mapped onto a tandem queueing network, where link $L_i$ is modeled as queue $Q_i$ (cf. Figure 2).

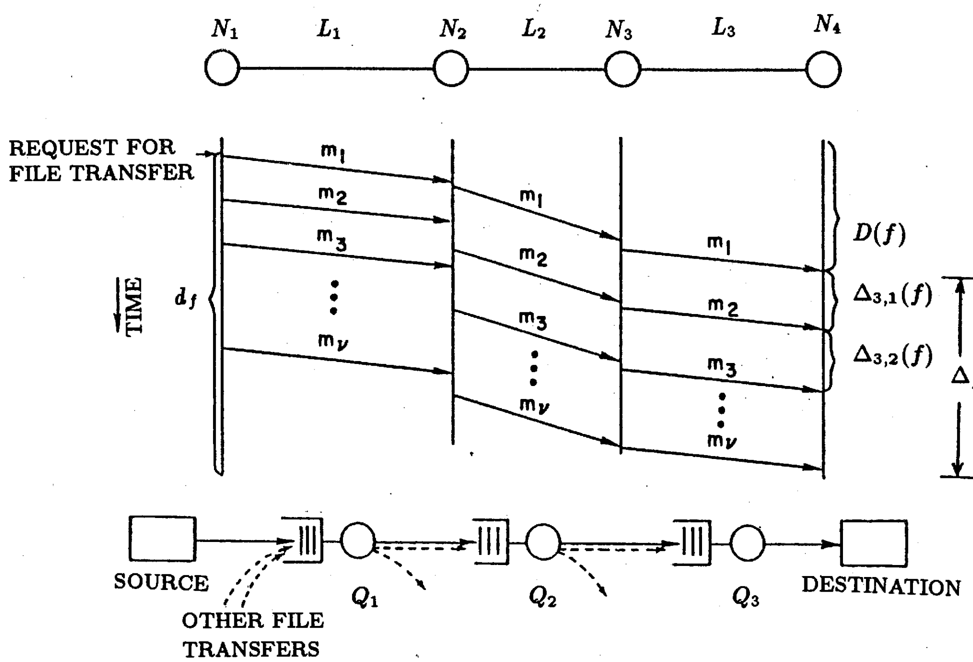



Fig. 2. Dynamic behavior of a FT.

A transfer delay analysis can now be performed as follows. Let $f$ be a file created by session $S$ segmented into messages $m_1, ..., m_\nu$. Taking into account that the unfinished work at the first link immediately after the $r$–th message's arrival is just $d_{i,r}(f)$, for the first link on $P$ using (a5) we then obviously find:

$$\Delta_{1,r}(f) = t_{1,r+1} - t_{1,r} \qquad (1 \le r \le \nu - 1)$$

where

$$a_{1,1} := 0,\ d_{1,1}(f) := d_1(f),\ a_{1,r} := a_{1,r-1} + \lambda_m^{-1},\ \text{and}\ t_{1,r} := a_{1,r} + d_{1,r}(f) \qquad (5)$$

$$d_{1,r}(f) = V(\lambda_1^{\text{ext}}, T_1, d_{1,r-1}(f); \lambda_m^{-1}) \quad (2 \le r \le \nu), \quad (\text{cf.}(3)).$$

The values $a_{1,r}$ and $t_{1,r}$ can be viewed as the expected $r$–th message arrival and departure epochs at queue $Q_1$, respectively, provided that the queueing process of $Q_1$ starts at time $t_0 = 0$.

For the computation of interdeparture times for a queue $Q_i$ $(i > 1)$ on path $P$ we assume that $m_1$ arrives at time $t_0 = a_{i,1} := 0$ at $Q_i$. The departure time of $m_1$ is then expected to be $t_{i,1} = d_i(f)$. Let for an index $r$ $(1 < r \leq \nu - 1)$ $a_{i,r-1}$, $t_{i,r-1}$ and $d_{i,r-1}(f)$ already be calculated. An expression for the interdeparture times at $Q_i$ in terms of interdeparture times at the preceding queue can then be derived as follows

$$\Delta_{i,r}(f) = t_{i,r+1} - t_{i,r} \quad (1 \leq r \leq \nu - 1),$$

where

$$a_{i,r} := a_{i,r-1} + \Delta_{i-1,r-1}(f), \;\; t_{i,r} := a_{i,r} + d_{i,r}(f) \quad (2 \leq r \leq \nu), \tag{6}$$

and

$$d_{i,r}(f) = V(\lambda_i^{\text{ext}}, T_i, d_{i,r-1}(f); \Delta_{i-1,r-1}(f)).$$

Obviously, the mean transfer delay for file $f$ is (see Figure 2)

$$d_f = D(f) + \Delta_f = (\nu - 1)/\lambda_m + \sum_{i=1,\dots,k} d_{i,\nu}(f) \tag{7}$$

where $\Delta_f := \sum_{r=1,\dots,\nu-1} \Delta_{k,r}(f)$ (which is sometimes called *reassembly delay* (Evequoz and Tropper, 1987)), and

$$D(f) := \sum_{i=1,\dots,k} d_i(f)$$

As our delay analysis is based on the first equation of (7) and the terms $\Delta_{k,r}(f)$ are known from (5), (6), what remains is to compute the terms $d_i(f)$.

**Computation of $d_i(f)$.** Let $(f_n)_{n=1,2\dots}$ be the sequence of files generated by $S$ and $m_{1,n}, m_{2,n}, \dots, m_{\nu,n}$ the messages belonging to the $n$–th file $(\nu := \nu(f_n))$; in the sequel; to simplify notation we will drop $f_n$ in our notation of $\nu$ in all cases where this correspondence is obvious. At each queue $Q_i$ the message $m_{1,1}$ has to queue only because of traffic caused by sessions others than $S$ so that the average delay $d_i(f_1)$ of $m_{1,1}$ at $Q_i$ may be estimated by the well-known mean value formula for M/G/1–queueing systems:

$$d_i(f_1) = T_i + \frac{1}{2}\lambda_i^{\text{ext}} T_i^{(2)} (1 - \lambda_i^{\text{ext}} T_i)^{-1}, \qquad i = 1, \dots, k, \tag{8}$$

and the average transfer time for $f_1$ can be computed from (7).

Suppose that for $n > 1$ $d_{i,\nu}(f_{n-1})$ is known from a previous iteration step. The unfinished work at that point of time, when all messages of $f_{n-1}$ have just been received by $Q_i$, is expected to be $d_{i,\nu}(f_{n-1})$ and the average first message delay for file $f_n$ at $Q_i$ is then given by:

$$d_i(f_n) = V(\lambda_i^{\text{ext}}, T_i, d_{i,\nu}(f_{n-1}); \delta_i(f_{n-1})), \qquad i = 1, \dots, k, \tag{9}$$

where

$$\delta_i(f_{n-1}) = \begin{cases} \lambda_f^{-1} - (\nu - 1)\lambda_m^{-1}; & i = 1 \\ \delta_{i-1}(f_{n-1}) + d_{i-1}(f_n) - d_{i-1,\nu}(f_{n-1}); & i > 1 \end{cases} \tag{10}$$

The last expression follows from the fact that for $i > 1$ $\delta_i(f_{n-1})$ is merely the average interdeparture time of messages $m_{\nu,n-1}$ and $m_{1,n}$ at $Q_{i-1}$.

### 2.4. Resulting Algorithm for Delay Calculations of Files

Provided that computational forms for the computation of virtual delays (as defined by (3)) are available (cf. section (2.5)), a sketch of an algorithm for the evaluation of mean transfer delays based on the iterative schema (5), (6), (9), (10) can then be given in a PASCAL–like notation as follows.

**Algorithm FTA:** Computes mean FT delay of the $n$–th file in sequence
given: Length $k$ of communication path;
service capacities $T_i, T_i^{(2)}$ $(i = 1, ..., k)$;
load characteristics $\lambda_f, \lambda_m, \nu = \bar{\nu}, \lambda_i^{ext}$ $(i = 1, ..., k)$.

```
f := 0;
Repeat          */Loop L1 over all files/*
  f := f + 1;
  For i := 1 to k do       */Loop L2 over all queues/*
    If f = 1, then compute d_i(f) from (8);
    Else
    Begin
      If i = 1, then
        δ_i(f) := λ_f^{-1} - (ν - 1)λ_m^{-1};
      Endif;
      compute d_i(f) from (9);
      compute δ_i(f) from (10);
    Endif;
    set a_{i,1} := 0; t_{i,1} := d_i(f);
    For r := 2 to ν do */Loop L3 over all messages of current file/*
      If i = 1, then compute d_{i,r}(f) from (5);
        Else compute d_{i,r}(f) from (6);
      Endif;
      compute a_{i,r}, t_{i,r}, Δ_{i,r-1}(f) from (5) (resp. (6));
    End r;
  End i;
  compute d_f from (7);
Until (f = n);
```

**Remark.**

i) Complexity of algorithm FTA is determined by $k(\nu-1)$ calculations of virtual waiting times (9), (5) due to the nested loops L2 and L3. To be more specific, all our delay calculations in section 3 and 4 required less than 5 [sec] using a VAX 11/780.

ii) FTA provides means for a first moment analysis of file transfers in the transient as well as in the steady state case: In the transient case an analyst is interested in the transfer characteristics of a special file (say the $n$–th in sequence). Consequently, FTA can be applied to that file directly, whereas in the steady state case a session is assumed to create files permanently, so that consecutive files cause a monotonously increasing, and because of (a6), convergent sequence of transfer delays and FTA has simply to be applied to the $n$–th generated file for sufficiently large $n$. To give specific examples here too for typical values of $n$ leading to convergence, it can be reported that in all of our numerous experiments a value of $n \leq 10$ has been sufficient to obtain the value of mean FT delay for the steady state case

(stopping rule: $|d_{f_n} - d_{f_{n-1}}| d_{f_{n-1}}^{-1} \leq \varepsilon$, where $\varepsilon$ has been chosen as $\varepsilon = 10^{-3}$).

iii) Starting from the second equation of (7) a transfer delay algorithm arises which has the complexity of FTA. The reason for that is the fact that computation of mean message delays requires mean interdeparture times at the preceding link (c.f. (6)).

## 2.5. Special Cases for Service Time Distributions

In this section $V(\lambda, T, v; t)$ will be determined for all service time distributions (with mean $T$), which are of interest in this paper. Deterministic service times (cf. special case I) are highly relevant because segmentation in packet–switched networks usually generates data units of fixed length (equal to the network's maximum packet length). Exponential service times (cf. special case II) have been (successfully) applied in the past to approximate strongly varying packet lengths with shorter packets occurring more frequently than larger ones. Exponentially distributed packet lengths, for example, have been assumed when modeling interactive traffic (Kleinrock, 1976). The assumption of different traffic types (fixed length packets being mixed with those of exponentially distributed length, cf. special case III) will be required in section 4 to model computer networks with different types of application–oriented services. The following analysis refers to a fixed but arbitrary queue $Q$ of the network. Throughout this section empty sums are supposed to be 0, and empty products are supposed to be 1.

**Special case I**: Deterministic service times

The unit impulse function $\delta_D$ at value $D$ can be viewed as the pdf (probability density function) for a deterministic (i.e. constant) service time of $D$ time units. Using the simple convolution property $\delta_a * \delta_b = \delta_{a+b}$ we can easily evaluate $\pi(v, t)$

for an M/D/1 queueing system with arrival rate $\lambda = \lambda^{\text{ext}}$ and deterministic service time $T = D$ from (2) for $v := v_0 < t$

$$\pi(v,t) = \begin{cases} e^{-\lambda t}, & c = 0 \\ e^{-\lambda t}(S_c(t) - \lambda D S_{c-1}(t)), & c > 0 \end{cases} \tag{11}$$

where $S_c(t) := \sum_{n=0,\ldots,c} (\lambda t)^n/n!$ and $c := c(v,t;D)$ is defined to be the greatest index $n$ such that (for $J := J(v,t)$ defined as in section 2.1) $\int_J (1 - u/t)\delta_{nD}$ du is nonzero (or equivalently $nD \leq t - v$). Substitution of $\pi(v,t)$ into (1) and integration by parts yields a simple expression for the expected virtual delay at time $t$ starting from an unfinished work of $v$ time units

$$V(\lambda, D, v; t) = D + v - (1 - \lambda D)t$$
$$+ \sum_{j=0,\ldots,c} (\lambda^{-1} H_c(v(j), v(j+1)) - D H_{c-1}(v(j), v(j+1))) \tag{12}$$

where $v(j) := jD + v$ $(j = 0, \ldots, c)$, $v(c+1) := t$, and $H_i(x,y) := 0$ for $i < 0$, $H_i(x,y) := \sum_{n=0,\ldots,i} \sum_{r=0,\ldots,n} (x^r e^{-\lambda x} - y^r e^{-\lambda y})\lambda^r/r! = e^{-\lambda x} \sum_{n=0,\ldots,i} S_n(x) - e^{-\lambda y} \sum_{n=0,\ldots,i} S_n(y)$ for $i \geq 0$.

**Special case II**: Exponential service times

Appendix B proves that for an M/M/1 queueing system with arrival rate $\lambda$ and service rate $\mu$, the probability $\pi(v,t)$ is given by the following expression for $v < t$

$$\pi(v,t) = e^{-\lambda t} e^{-\mu(t-v)}$$
$$\times \left[ \sum_{n \geq 0} (\lambda t)^n/n! \left( \sum_{j \geq n} \mu^j (t-v)^j/j! - \lambda/\mu \sum_{j \geq n+2} \mu^j (t-v)^j/j! \right) \right] \tag{13}$$

Moreover, Appendix B introduces an efficient algorithm for numerical evaluation of $\pi(v,t)$ according to equation (13).

As in case I of deterministic service times, we can now substitute $\pi(v,t)$ into (1) in order to get an expression for the expected virtual delay at time $t$ starting with unfinished work of $v$ time units. The result we obtained after some lengthy and tedious calculations (which are roughly indicated in Appendix B), is the following.

$$V(\lambda, \mu^{-1}, v; t) = \mu^{-1} + v - (1 - \lambda/\mu)t + \alpha^{-1} \sum_{n \geq 0} (\lambda/\alpha)^n g(n;n)$$
$$+ \alpha^{-1} \sum_{n \geq 0} (\lambda/\alpha)^n g(n+1;n)$$
$$+ (1 - \lambda/\mu)\alpha^{-1} \sum_{n \geq 0} (\lambda/\alpha)^n \sum_{j \geq n+2} g(j;n) \tag{14}$$

where $\alpha := \lambda + \mu$, and $g(j;n) := (-\mu v)^j \sum_{r=0,\ldots,j} \binom{n+r}{r} 1/(j-r)!(-\alpha v)^{-r} e^{\mu v} \times \sum_{i=0,\ldots,n+r} [e^{-\alpha v}(\alpha v)^i/i! - e^{-\alpha t}(\alpha t)^i/i!]$.

**Remark.** A different method for the computation of virtual waiting times for queueing systems with Erlangian, in particular exponential service times is presented in (Kim, 1988). The time complexity of the algorithm presented there is of the same order as the complexity of the algorithm given in Appendix B.

**Special case III:** Two types of service time distributions: Deterministic and exponential service times.

In section 4 computer networks are considered with two different types of sessions (for example FT and dialog sessions) generating messages which require deterministic and exponential service times, respectively. Let $\lambda^{(1)}$ be the arrival rate of type–1–messages and $\lambda^{(2)}$ the arrival rate of type–2–messages at queue $Q$. Further, let type–1–messages have deterministic service times ($D$ time units) and type–2–messages exponential service times (with parameter $\mu = T^{-1}$). Then $Q$ behaves exactly the same way as a queue with the following input and service characteristics.

Messages (of only one type) arrive according to a Poisson stream with rate $\lambda = \lambda^{(1)} + \lambda^{(2)}$. The service strategy remains FCFS. At time epochs, when a message is selected for service, its service time with probability $p := \lambda^{(1)}/\lambda$ is deterministic (i.e. $D$ time units) and with probability $q := 1 - p$ is drawn from an exponentially distributed population with rate $\mu$. For the pdf $f(t)$ of the service time and its mean $T$ holds:

$$f(t) = pD + q\mu e^{-\mu t}, \quad t \geq 0, \quad T = pD + q\mu^{-1} \tag{15}$$

Set $b(n,i) := t - v - D(n-i) \qquad (0 \leq i \leq n)$, and

$C(n,i) := \binom{n}{i} p^{n-i} q^i [(1 - t^{-1}(i/\mu + D(n-i)))(1 - \sum_{r=0,\ldots,i-1} b(n,i)^r e^{-\mu b(n,i)} \mu^r / r!)$
$\quad + b(n,i)^i e^{-\mu b(n,i)} \mu^{i-1} / (t(i-1)!)]$.

Now, if $c = c(v,t;D)$ and $v(j)$ $(j = 0, ..., c+1)$ are defined as in (11) and(12), respectively, expressions for $\pi(v,t)$ and the virtual waiting time at time $t > v$ can be given as follows (cf. Appendix A):

$$\pi(v,t) = e^{-\lambda t} \left[ 1 + \sum_{n=1,\ldots,c} p^n (1 - nDt^{-1})(\lambda t)^n / n! \right.$$
$$\left. + \sum_{n \geq 1} (\lambda t)^n / n! \sum_{i=1,\ldots,n} \psi_{n,i} C(n,i) \right] \tag{16}$$

where $\psi_{n,i}$ is defined to be 1 in case of $D(n-i) \leq t - v$, and 0, otherwise. For fixed $v, D$ $(v \geq 0,\ D > 0)$ $\gamma(u) := c(v,u;D)$ (for $u \geq v$), $\gamma(u) := 0$

(for $u \leq v$) is a monotonously increasing step function jumping at $v(j)$ ($j = 1, ..., c$) and constant on the half-open intervals $A(j) := [v(j), v(j+1))$, $j = 0, ..., c-1$, and on $A(c) := [v(c), v(c+1)]$. We can thus write

$$W(\lambda, T, v; t) = v - (1 - \lambda T)t + \sum_{j=0,\ldots,c} W_j \tag{17}$$

where: $W_j := \int_{A(j)} \pi(v, u)du$. For the computation of $W_j$ we need some further abbreviations.

Let $f_j := e^{-\lambda v(j)} - e^{-\lambda v(j+1)}$, and

$g_{j,r}(\xi) := (v(j)^r e^{-\xi v(j)} - v(j+1)^r e^{-\xi v(j+1)})\xi^r / r!$ (for any real number $\xi$).

Furthermore, set $F_j := \lambda^{-1} f_j$, $G_j(n) := \lambda^{-1} \sum_{r=0,\ldots,n} g_{j,r}(\lambda)$,

$M_j(n) := \sum_{r=0,\ldots,n-1} g_{j,r}(\lambda)$ ($n \geq 1$),

$N_j(n, i) := X_j(n, i) - Y_j(n, i) + Y_j'(n, i, i)\mu^{i-1}/(i-1)!$,

where $X_j(n, i) := G_j(n) - \sum_{r=0,\ldots,i-1} X_j'(n, i, r)\mu^r / r!$

$$\begin{aligned} X_j'(n, i, r) : &= \lambda^n e^{\mu v(n-i)} \sum_{k=0,\ldots,r} \binom{r}{k} (-1)^k v(n-i)^k \; (n+r-k)!/n! \\ &\times \; (\lambda + \mu)^{-(n+r-k+1)} \sum_{s=0,\ldots,n+r-k} g_{j,s}(\lambda + \mu) \end{aligned}$$

$Y_j(n, i) := (D(n-i) + i/\mu)(n^{-1} G_j(n-1) - \sum_{r=0,\ldots,i-1} Y_j'(n, i, r)\mu^r / r!)$, and

$$\begin{aligned} Y_j'(n, i, r) : &= n^{-1} \lambda^n e^{\mu v(n-i)} \sum_{k=0,\ldots,r} \binom{r}{k} (-1)^k v(n-i)^k \\ &\times \; (n+r-k-1)!/(n-1)!(\lambda + \mu)^{-(n+r-k)} \sum_{s=0,\ldots,n+r-k-1} g_{j,s}(\lambda + \mu) \end{aligned}$$

It now follows that

$$\begin{aligned} W_j = F_j &+ \sum_{n=1,\ldots,c} p^n (G_j(n) - D M_j(n)) \\ &+ \sum_{n \geq 1} \sum_{i=1,\ldots,n} \binom{n}{i} p^{n-i} q^i \psi_{n,i} N_j(n, i) \end{aligned} \tag{18}$$

Although the computation of mean virtual delays from (17) seems to be rather complex, only elementary operations are involved, so we find it worth the effort in view of section 4 having considered the *mixed traffic* case, too. Here as in the case of exponential service times the infinite series in (18) converge rather fast, so that (17) enables an approximative evaluation of mean virtual delays. If $p = 0$ ($q = 0$), then (16) and (17) simplify the results for the exponential and deterministic service times, respectively.

## 3. Model Validation

In order to get estimations for the accuracy and robustness of our modeling approach, the analytical model is validated by means of simulation models. Using the RESQ– and the MAOS modeling system (Jobmann, 1985; Sauer and MacNair, 1983) two series of validation experiments were carried out for the homogeneous traffic case (i.e. all sessions in CN are FT sessions). Both intend to analyze for the steady state case, how a tagged session $S$ mapped upon a two–hop–path $P$ is affected by additional transfer sessions $S_1, ..., S_r$ partly using $P$. All service times at links belonging to $P$ are supposed to be deterministic (cf. special case I of section 2.5). The chosen tree–like configuration is depicted in Figure 3.

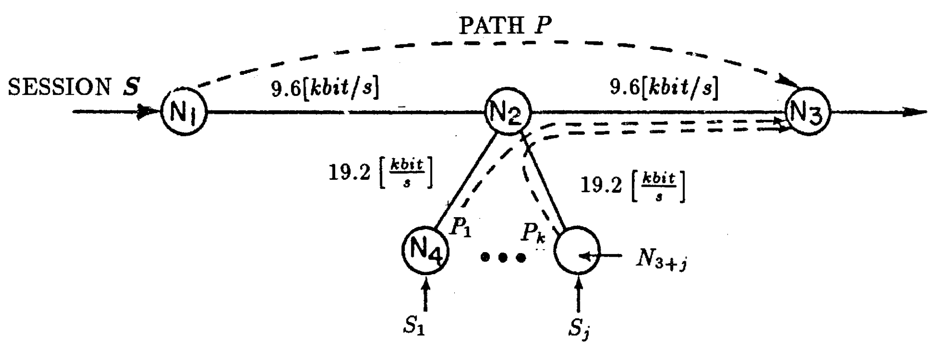



Fig. 3. Network configuration for series I, II.

Load assumptions for both series of experiments are:

- Session $S$ (permanently) transmits files from node $N_1$ to $N_3$ via path $P$.
- Session $S_i$ (permanently) transmits files from node $N_{3+i}$ to $N_3$ via link $(N_2, N_3)$ $(i = 1, ..., r)$.
- $\lambda_f(S) = 1/400[files/sec]$,
- $\lambda_m(S) = 10[msg/sec]$, and
- $\nu(S) = 40[msg/file]$.

Additionally, for series I the sessions $S_i$ are supposed to have the same load characteristics as $S$. Series I of experiments was carried out to study the consequences of varying the number $r$ of additional transfer sessions, which create traffic interfering with that of the tagged session $S$. Furthermore, for each r the message length was varied (from 4 *kbit* to 30 *kbit*), thus varying the utilization

$\zeta_{2,3}$ of link $(N_2, N_3)$. As can be seen from Figure 4, the analytic results for all $r$ agree with the simulated ones fairly well. Although assumption (a5) is violated for link $(N_2, N_3)$ (because of deterministic service times at link $(N_1, N_2)$ and bulk arrivals at nodes $N_{3+i}$), for $\zeta_{2,3}$ in the range [0, 0.8] the deviations from the simulation results turn out to be less than 10% showing the superiority of our new approach to the analytic method presented by Wolfinger, (1986).

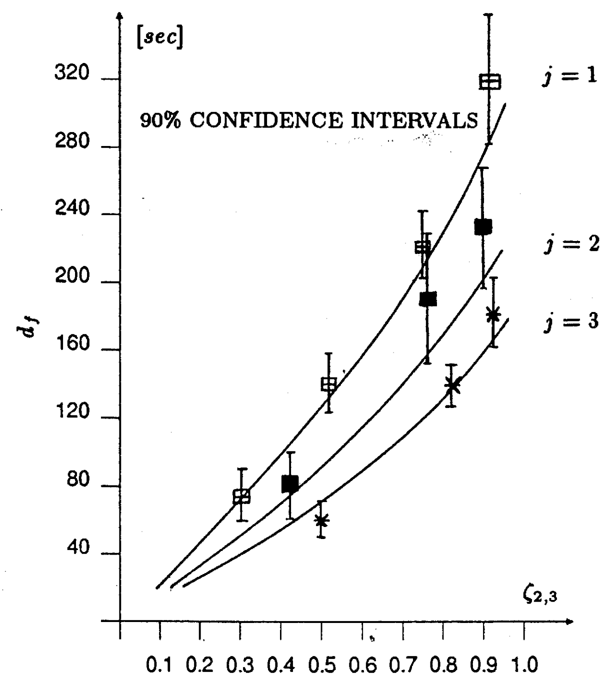



Fig. 4. Analytical and simulation results for series I.

Unlike series I, where the number of incoming links at node $N_2$ was considered, in series II the capacities of links $(N_{3+i}, N_2)$ (or equivalently, the service rates for these links) are kept variable, thus introducing different levels of fluctuations in the arrival of messages at node $N_2$. Furthermore, file arrival rates for sessions $S_i$ are varied in order to consider utilization factors $\zeta_{2,3}$ of link $(N_2, N_3)$ in the range [0.2, 0.9]. Figure 5 illustrates the accuracy of the analytical model for series II of validation experiments assuming two most realistic values for $k$ ($k = 1$ and $k = 2$, i.e., node $N_2$ connected to up to 4 nodes) and network–wide constant message length of 16 *kbit*. The case $k = 1$ reveals close agreement between analytical and simulation results. This is true even for configurations where messages of additional file transfers are served with a rate of 10 [$msg/sec$]; this is remarkable because the resulting very large arrival rate at node $N_2$ could indeed also describe the situation where additional files are directly generated in node $N_2$. For the case $k = 2$, the results are not too different and only for configurations where messages of additional file transfers are transmitted towards node $N_2$ via two high speed links (or generated locally) deviations become more significant.

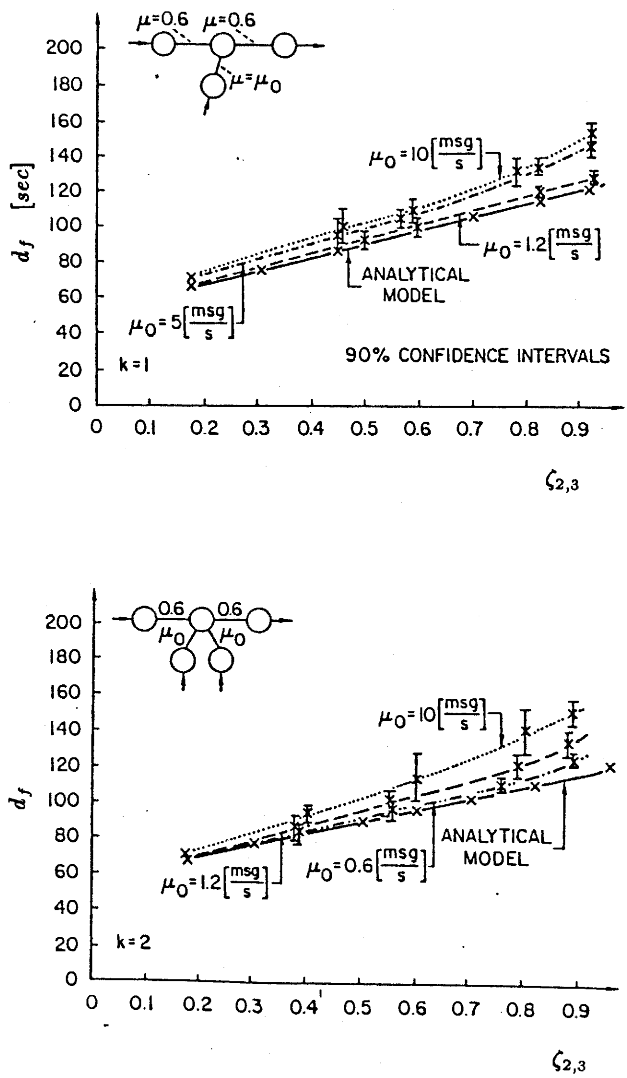



Fig. 5. Analytical and simulation results for series II.

Further validation experiments, considering e.g. configurations with a larger number of hops, have been carried out. They indicate that the accuracy requirements of less than 10% deviations with respect to simulation results are not only met by the analytical models in nearly any configuration considered, but that deviations are most often well below 5%.

## 4. Extended Model for Delay Analysis in Mixed Traffic Networks

After having done a few steps towards a validation of our (basic) FT model, we will now show how the basic model may be extended to cover heterogenous traffic situations. For this purpose computer networks will be considered which offer two application–oriented services to end– users: FT and interactive (dialog) service. The transfer service shall be supposed as in the basic model. The dialog service will be supposed to imply a flow of (dialog) messages, called *interactive traffic*, between a terminal user (at node $N_i$) and a timesharing–computer (node $N_j$). The generalized model which also takes into account dialog service will be called *mixed traffic* model.

The assumptions we made for the basic model concerning configuration and FT sessions are still valid for the mixed traffic model. The interactive traffic is reflected only by the stream of dialog messages arriving at each link. We assume that the arrival stream of dialog messages to each link $L_i$ (modeled by queue $Q_i$) is Poisson with rate $\lambda_{i,\text{dia}}^{\text{ext}}$. In order to cover realistic traffic characteristics in computer networks, messages created by interactive traffic are supposed to have non–preemptive priority over messages generated by FT sessions.

Let us write $\lambda_{i,\text{ft}}^{\text{ext}}$ for the arrival rate of external messages (generated by transfer sessions other than a tagged session $S$) and $\lambda_i^{\text{ext}} = \lambda_{i,\text{ft}}^{\text{ext}} + \lambda_{i,\text{dia}}^{\text{ext}}$ for the total arrival rate of external messages at $Q_i$. The average service time at $Q_i$ is then $T_i^{\text{ext}} = (\lambda_{i,\text{ft}}^{\text{ext}}/\lambda_i^{\text{ext}})T_{i,\text{ft}} + (\lambda_{i,\text{dia}}^{\text{ext}}/\lambda_i^{\text{ext}})T_{i,\text{dia}}$, where $T_{i,\text{ft}}$ and $T_{i,\text{dia}}$ denote the mean transfer time for transfer and dialog messages, respectively. Obviously, the service strategy for each $Q_i$ remains load conserving for the mixed traffic case, so that the mean unfinished work $U_i(t)$ at time t is the same as for FCFS.

Given a startload of $v_0$ time units we then have (cf. special case III of section 2.5)

$$U_i(t) = W(\lambda_i^{\text{ext}}, T_i^{\text{ext}}, v_0; t) \tag{19}$$

provided that only external (FT or dialog) messages have arrived at $Q_i$ before time $t$. A tagged transfer message $m$ generated by session $S$ will then experience the following mean virtual delay at queue i when arriving at time $t$

$$V_i(t) = T_{i,\text{ft}} + U_i(t) + E_i(t) \tag{20}$$

where $E_i(t)$ is the mean amount of work brought into $Q_i$ by external dialog messages, which arrive later than $t$ but are served before $m$. An expression for $E_i(t)$ in terms of known performance characteristics can be derived easily as follows

$$U_i(t) + E_i(T) = U_i(t) + U_i(t)\lambda_{i,\text{dia}}^{\text{ext}} T_{i,\text{dia}} + U_i(t)(\lambda_{i,\text{dia}}^{\text{ext}} T_{i,\text{dia}})^2 + \ldots =$$
$$= U_i(t) / (1 - \zeta_{i,\text{dia}}^{\text{ext}}) \tag{21}$$

where $\zeta_{i,\text{dia}}^{\text{ext}} = \lambda_{i,\text{dia}}^{\text{ext}} T_{i,\text{dia}}$ is the utilization of $Q_i$ by external dialog messages. Let further $\zeta_i^{\text{ext}}$ be the total utilization of $Q_i$ by external messages and $T_{i,\text{ft}}^{(2)}$ and $T_{i,\text{dia}}^{(2)}$ the second moments of the service times for FT and dialog messages, respectively. Then, if the average first message delay $d_i(f_1)$ at $Q_i$ for the first file $f_1$ created by $S$ is computed by (cf. Kleinrock, 1976):

$$d_i(f_1) = \frac{\lambda_{i,\text{ft}}^{\text{ext}} T_{i,\text{ft}}^{(2)} + \lambda_{i,\text{dia}}^{\text{ext}} T_{i,\text{dia}}^{(2)}}{(1 - \zeta_i^{\text{ext}})(1 - \zeta_{i,\text{dia}}^{\text{ext}})} \tag{22}$$

and the mean virtual delay $V_i(t)$ is computed by (20) instead of (3), algorithm FTA evaluates the mean FT delay even for the mixed traffic case.

## 5. Areas of Model Applicability

In order to appreciate the potential use of the models introduced, let us shortly discuss some of their main application areas. Basic possibilities of using our models for file delay calculations include the following:

i) The file delay calculations can be used to support the design of new routing algorithms of ***shortest path*** type (e.g. in the case where delay minimization is representing the objective function).

ii) In the configuring and tuning of communication systems one will be able to investigate whether a specific network configuration will allow one to satisfy (for a given load) application – oriented performance requirements, such as limited file delay. The reader should note that limiting file delays becomes increasingly important in today's computer networks, where access to file servers is very common.

iii) Interactions of different kinds of traffic (bulk data transfer and interactive traffic typically of higher priority) can be studied based on the models presented. The mutual impact of this traffic types can be determined from the performance point of view.

iv) Dimensioning of communication networks can be supported, in particular choice of line speed for point–to–point links, e.g. between switching nodes, bridges, gateways, etc. (Note that choice of high speed links on paths still including one or more ***slow hops*** will not be very useful for file transfer according to our results as the latter will still remain bottlenecks).

In general, the following properties of a computer/communication network have a strong impact on the applicability of the models presented:

- limitations in buffer sizes available in intermediate (switching) nodes,
- the way in which performance–relevant communication functions (such as routing, error and flow control) are realized,
- the type of switching technique used (circuit–, packet– or message switching).

Our model assumptions (cf. sections 2 and 4) imply that some restrictions with respect to the above–mentioned properties of a network have to be fulfilled. In particular, the following network characteristics are required as a prerequisite for our FT models to be realistic:

**Buffers**: Buffer in intermediate nodes has to be large enough to exclude significant buffer bottlenecks and, in particular, any loss of messages due to buffer overflow.

**Routing**: Fixed routing for sessions between communicating application processes exists (which does not necessarily imply fixed routing between each pair of endsystems).

**Error control**: Control information exchanged to achieve error control can be neglected.

**Flow control**: The influences of hop–level and of host–host flow control, cf. (Gerla and Kleinrock, 1980), should not have significant impact on performance; however, some types of virtual circuit (process–process) flow control can be taken into account (cf. section 5.1) though it is evident that flow control based on sliding window techniques, cf. (Schwartz, 1987), requires a different modeling approach. However, a variant of our analytical model (not considered in this paper) covers the modeling of sliding window techniques, too.

**Switching technique**: Packet switching via unlayered virtual circuits is modeled in a realistic way; if connection set–up and clearing may be neglected, our models can be applied in a trivial way for circuit switching too (the communication network degenerates to a single hop in this case).

### 5.1. Acceptable Types of Process–Process Flow Control

Among other things, there is a frequently used type of flow control between application processes that can be taken into account by the models introduced above. Consider a flow control scheme according to a window mechanism (cf. Schwartz, 1987), which works as follows:

- a constant credit of w data units is specified for communicating processes;
- if the credit expires the sending process has to wait for an acknowledgement;
- the receiving process generates an acknowledgement after having received exactly w data units.

This mechanism can be used for flow control as well as for setting checkpoints during transfer.

Evidently, the special case of $w = 1$ for the above scheme implies a send–and–wait protocol on Session Layer. In this case, the typical pipeline effect disappears in the FT.

Let now $\bar{\nu}$ be the mean number of messages generated by the transfer of one file and let (without loss of generality) $\bar{\nu} = kw$ for some integer $k$. Then, the only case of interest is $k \geq 2$, which can be solved as follows:

i) Calculate the mean delay $T_{\text{ack}}$ of an acknowledgement, traveling from receiver process $R$ to sending process $S$, and seeing the network in equilibrium.

ii) Apply the iterative algorithm to a network with the following modified load pattern generated by $S$: *subbulks* of w messages, each is generated with a mean interarrival time between messages of $1/\lambda_m$; $k$ such subbulks are generated with a mean interarrival time between subbulks of $1/(2T_{\text{ack}})$; files (each of $k$ subbulks on the average) are generated with a mean interarrival time of $1/\lambda_f$.

iii) The transfer delay of a total file can now be determined by adding $kT_{\text{ack}}$ to the sum of all $k$ subbulk delays.

Obviously, the modified calculation algorithm directly results from the algorithm given in section 2.4 by just adding a further iteration on the subbulks generated within each FT.

### 5.2. Modeling of FT in Interconnected Computer Networks

**Case I**: Coupling on Application Layer
Interconnected computer networks are typically coupled by so–called *gateway computers* which have to solve the mutual mapping between the computer network architectures involved. A first type of coupling for FT can be achieved on Application Layer (cf. Bauerfeld, 1986; Wolfinger *et al.*, 1986). In this case, files are e.g. completely reassembled within a gateway (computer) by storing the total file temporarily on the gateway's disc. Such an intermediate buffering of files on disc implies that the file delay $T_f(C)$ for an interconnected network $C$ via single networks $\{CN_1, CN_2, ..., CN_m\}$ can be calculated as follows

$$T_f(C) = \sum_{i=1,\dots,m} T_f(CN_i)$$

where $T_f(CN_i)$ denotes the file delay within computer network $CN_i$, which can be determined by looking at network $CN_i$ in isolation.

**Case II**: Coupling on Network Layer
A second type of coupling for FT can be achieved on Network Layer (cf. Bauerfeld, 1986; Wolfinger *et al.*, 1986). In this case, from the performance point of view, the gateway can be considered as an additional intermediate switching node transfering messages (*on the fly*) roughly in the same way as intermediate nodes of the networks

interconnected. For delay calculation we thus just have to look at the total path through all the networks involved in the transfer.

**Case III**: LAN–WAN interconnection
Until now we have restricted ourselves to communication networks based on point–to–point links, connecting a set of intermediate nodes to form an irregularly meshed topology. However, it can be seen that in some cases slightly loaded local–area networks (in particular those with ring or bus topology) can be described quite realistically by means of the models presented in this paper. This claim is valid, because important types of local–area networks (LANs) also satisfy the basic assumptions, which we had to make with respect to data transfer on point–to–point links in wide–area networks (WANs), i.e.

- at most one data transfer request is served at any instant of time,
- FCFS–service discipline for all nodes having access to the common transmission medium,
- service time is completely determined by the message length (in particular, service time is proportional to length).

Thus, we are finally able to apply the models (of sections 2 and 4) to interconnected LAN–WAN configurations, comprising point–to–point links as well as LANs based e.g. on bus or ring topology. Of course, an important restriction is that no reassembly of files takes place in intermediate nodes but only in endsystems.

**Example**: Modeling of LAN–WAN interconnections
To illustrate the modeling of configurations which include several LANs let us give a short example. We look at a configuration including 11 nodes, 6 point–to–point duplex links and 2 LANs. The topology of this configuration is depicted Figure 6.

Let the following five sessions $\boldsymbol{S}_1, ..., \boldsymbol{S}_5$ exist:
$\boldsymbol{S}_1$ on path $(N_1 \to N_4 \to N_5 \to N_7 \to N_9)$ as tagged session
$\boldsymbol{S}_2$ on path $(N_2 \to N_4 \to N_6)$
$\boldsymbol{S}_3$ on path $(N_{11} \to N_8 \to N_5 \to N_4)$
$\boldsymbol{S}_4$ on path $(N_6 \to N_5)$
$\boldsymbol{S}_5$ on path $(N_5 \to N_8 \to N_{10})$.

We now have to take into account that

- in node $N_i$ all those sessions generate interfering data transfer traffic for which $N_i$ is part of their path;
- in a local–area network $\text{LAN}_j$ all those sessions generate interfering data traffic for which $\text{LAN}_j$ is part of their path.

Therefore, the queueing model given in Figure 7 allows us to calculate the file delay for the tagged session $\boldsymbol{S}_1$.

The reader will realize that one of the basic assumptions we make in modeling LANs, according to our approach, is that a LAN can be modeled in a sufficiently realistic way by an M/G/1–queueing system. Important publications on network performance evaluation, such as (Schwartz, 1987) and (Bertsekas and Gallager, 1988), justify this M/G/1 assumption at least for two of the presently dominant LAN architectures, namely as an approximation for Ethernet and Token Ring.

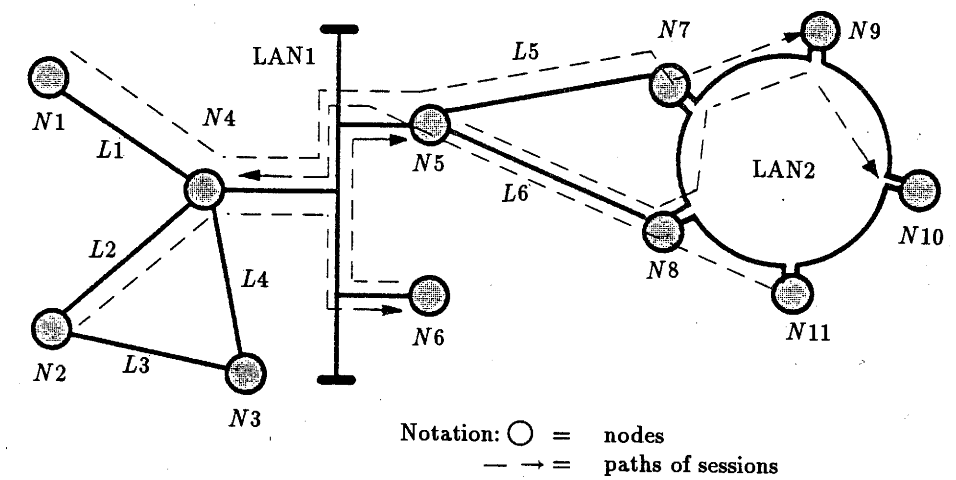



Fig. 6. A sample configuration of an interconnected network.

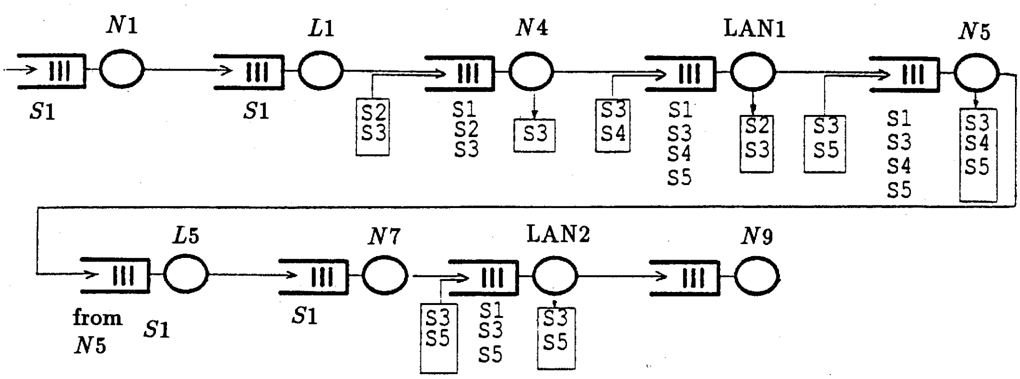
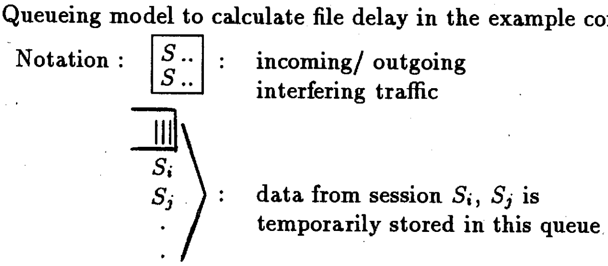



Fig. 7. Queueing model to calculate file delay in the example configuration:



## 6. Summary

For the important classes of application–oriented services in packet–switched networks, which yield to the transfer of bulk data (such as FT, remote job entry, electronic mail etc.), we have introduced a set of approximative analytical queueing models. The basic models to determine file delays have been specialized, in particular, for the cases of deterministic and exponential service times. Extensions of the basic model allowed us to describe mixed traffic situations, where e.g. interactive traffic (of higher priority) overlays a given FT traffic.

Model validation by simulation has indicated the high precision of results for a large variety of configurations and load situations. A discussion on possible areas of model application has been included in the paper as well as some hints on the limits of the models. Of course, in high–speed networks our approach will have to be applied in such a way that the delay caused by lines (e.g. fiber–optic connections) is neglected, whereas the (switching) nodes should be modeled to represent the service–stations in the tandem queue. Most of the examples in the paper illustrate the *dual* situation (i.e. lines representing the communication system bottlenecks, as is still quite often the case in wide–area networks with links operating in the order of 10 *kbit/s* or in networks with satellite links).

Comparisons of analytical results to measurements obtained from existing (packet–switched and interconnected) networks are planned for the future in order to complete model validation by using real world data.

We hope that the models introduced here may represent a small step with respect to a more realistic load modeling in computer network performance analyses.

**Acknowledgement.** We are indebted to Mr. Jan Kummer who supported our validation experiments and who pointed out an arithmetical error occurring in an earlier draft of this paper.

## References

**Bauerfeld W. L.** (1986): *Interconnection issues between local area and wide area networks in DFN.*– In: P. Kuehn (ed.): Proc. 8–th Int. Conf. on *Computer Communication.* North–Holland Publ. Comp., pp.717–722.

**Bertsekas D. and Gallager R.G.** (1988): *Data Networks.*– Prentice–Hall.

**Benes V. E.** (1957): *On queues with Poisson arrivals.*– Ann. Math. Statist., v.28, pp.670–677.

**Cohen J. W.** (1982): *The single server queue.*– North–Holland Publ. Comp.

**Evequoz J. W. and Tropper C.** (1987): *On reassembly delay in packet–switching networks.*– Proc. IEEE INFOCOM '87, 6–th Ann. Conf., pp.971–980.

**Gerla M. and Kleinrock L.** (1980): *Flow control: A comparative survey.*– IEEE Trans. Comm., v.COM–28, No.4, pp.553–574.

**Jobmann M.** (1985): *Modellbildung und analyse von Rechensystemen mit Hilfe des Programmiersystems MAOS.*– Proc. 3-rd GI Conf. *Messung, Modellierung und Bewertung von Rechensystemen* (in German).

**Kim C.** (1988): *Virtual waiting time analysis of an erlangian single server queueing system.*– State University of New York at Buffalo, Technical Report, 88–19.

**Kleinrock L.** (1976): *Queueing Systems.* Vol. II: *Computer Applications.*– John Wiley & Sons.

**Lampson B. W., Paul M. and Siegert H. J. (eds.)** (1981): *Distributed Systems. Architecture and Implementation.*– Springer–Verlag.

**Rubin I.** (1974): *Communication networks: message path delays.*– IEEE Trans. Information Theory, v.IT–20, No.6.

**Rubin I.** (1975): *Message path delays in packet switching communication networks.*– IEEE Trans. Comm., v.COM–23, No.2.

**Spirn J. R., Chien J. and Hawe W.** (1984): *Bursty traffic local network modeling.*– IEEE J. Selected Areas in Communications, v.SAC–2, No.1, pp.250–257.

**Schwartz M.** (1987): *Telecommunication Networks: Protocols, Modeling and Analysis.*– Addison–Wesley Publ. Comp.

**Sauer Ch. and MacNair E.** (1983): *Simulation of Computer Communication Systems.*– Prentice–Hall.

**Tanenbaum A. S.** (1989): *Computer Networks.* (2nd edition).– Prentice–Hall.

**Wolfinger B., Johannsen W. and Schulze J.** (1986): *Simulation of gateway computers using different modeling tools.*– In: S. Schoemaker (Ed.): Computer Networks and Simulation III.– North–Holland Publ. Comp., pp.123–152.

**Wolfinger B.** (1986): *Delay approximations for File transfers in computer networks.*– Proc. IEEE INFOCOM'86, 5-th Ann. Conf., pp.609–615.

Received May 25, 1992
Revised August 31, 1992

## Appendix A

Adopting the notations and abbreviations of section 2.5 we give here a sketch of the proofs for formulas (16) and (17): Starting from (15) the Laplace transform $L_f(x)$ of density $f(t)$ is

$$L_f(x) = pe^{-Dx} + q\frac{\mu}{\mu + x}, \tag{A.1}$$

so that for $L_n(x) := (L_f(x))^n$ follows

$$L_n(x) = p^n e^{-nDX} + \sum_{k=1}^{n-1} \binom{n}{k} p^{n-k} e^{-D(n-k)x} q^k \frac{\mu^k}{(\mu + x)^k} + q^n \frac{\mu^n}{(\mu + x)^n}$$

Inversion of $L_n(x)$ yields the density

$$f_n(t) = p^n \delta_{nD}(t) + \sum_{k=1}^{n} \binom{n}{k} p^{n-k} q^k \eta_{D(n-k)}(t), \tag{A.2}$$

where $\delta_x(t)$ is the Dirac function at value $x$ and

$$\eta_{D(n,k)}(t) := \psi_{n,k} \mu^k \frac{(t - D(n-k))^{k-1}}{(k-1)!} e^{-\mu(t-D(n-k))}$$

with $\psi_{n,k}$ defined as in section 2.5.

For the computation of $S_n := \int_0^{t-v_0} \frac{t-u}{t} f_n(u) du$ we need the following formula which follows from integration by parts.
For any positive integer $k$ and any real numbers $a, b, \xi$ $(a < b)$ we have

$$\frac{\xi^k}{(k-1)!} \times \int_a^b x^{k-1} e^{-\xi x} dx = \sum_{r=0}^{k-1} (a^r e^{-\xi a} - b^r e^{-\xi b}) \frac{\xi^r}{r!} \tag{A.3}$$

Using (A.3) after a few simple manipulations we arrive at

$$S_n = \psi_{n,k} p^n \left(1 - \frac{nD}{t}\right) + \sum_{k=1}^{n} \psi_{n,k} C(n,k)$$

Plugging this expression for $S_n$ into (2) yields the claimed formula for $\pi(v_0, t)$. We shortly consider two special cases.

**Case 1**. If $q = 0$ (i.e. deterministic service times), (16) simplifies directly to (11).

**Case 2**. If $p = 0$ (i.e. exponential service times), (16) writes

$$\begin{aligned} \pi(v_0, t) &= e^{-\lambda t} \left\{ 1 + \sum_{n=1}^{\infty} \frac{(\lambda t)^n}{n!} \left[ \left(1 - \frac{n}{\mu t}\right) \left(1 - \sum_{j=0}^{n-1} \frac{(\mu \tau)^j}{j!} e^{-\mu \tau}\right) \right.\right. \\ &\left.\left. + \frac{\mu^{n-1} \tau^n}{t(n-1)!} e^{-\mu \tau} \right] \right\} \end{aligned} \tag{A.4}$$

where $\tau = t - v_0$.

The computation of the virtual delay at time $t$ involves the evaluation of $I := \int_0^t \pi(v_0, u) du$ and with it the evaluation of the integrals $W_i$ $(i = 1, ..., c)$. Applying (A.3) this can be done in a straightforward manner, although the computation is rather tedious and lengthy. A rigorous derivation of (17), (18) is left to the reader.

## Appendix B

In the following section we prove the expressions for $\pi(v_0, t)$ and for $\int_0^t \pi(v_0, x)dx$ for the case of exponentially distributed service times, cf. equations (13), (14) in section 2.5 (special case II).
In Appendix A, a first expression for $\pi(v_0,t)$--which covers the service time distributions $D$ (deterministic) and $M$ (exponential)- has been given, cf. (A.4). Equation (A.4) can be transformed into

$$\begin{aligned}\pi(v_0,t) &= e^{-\lambda t} + e^{-\lambda t}e^{-\mu\tau}\sum_{n=1}^{\infty}\frac{(\lambda t)^n}{n!}\left(1-\frac{n}{\mu t}\right)\sum_{j=n}^{\infty}\frac{(\mu\tau)^j}{j!} \\ &+ e^{-\lambda t}e^{-\mu\tau}\sum_{n=1}^{\infty}\frac{(\lambda t)^n}{n!}\frac{(\mu\tau)^n}{(n-1)!}\frac{1}{\mu t}\end{aligned} \tag{B.1}$$

if we assume exponentially distributed service times. (Note that: $\tau = t - v_0$ as in Appendix A). Shifting the index by 1 in the second summation and summarizing the first two terms yields

$$\pi(v_0,t) = e^{-\lambda t}e^{-\mu\tau}\left(\sum_{n=0}^{\infty}\frac{(\lambda t)^n}{n!}\left[\sum_{j=n}^{\infty}\frac{(\mu\tau)^j}{j!} - \frac{\lambda}{\mu}\sum_{j=n+2}^{\infty}\frac{(\mu\tau)^j}{j!}\right]\right) \tag{B.2}$$

We want to consider (B.2) as the final result for $\pi(v_0,t)$, though it is evident that other representations of this result would be possible, too. Therefore, it seems appropriate to study shortly the problem of numerical evaluation for equation (B.2). For this purpose let us set

$$a_n := \frac{(\mu\tau)^n}{n!}, \qquad b_n := \frac{(\lambda t)^n}{n!}$$

$$A_n := \sum_{j=n}^{\infty} a_j, \qquad X_n := A_n - \frac{\lambda}{\mu}A_{n+2} \quad \text{and} \quad S_n := b_n X_n$$

With these abbreviations equation (B.2) can be rewritten as follows

$$\begin{aligned}\pi(v_0,t) &= e^{-\lambda t}e^{-\mu\tau}\sum_{n=0}^{\infty}S_n = e^{-\lambda t}e^{-\mu\tau}\sum_{n=0}^{\infty}b_n\left(A_n - \frac{\lambda}{\mu}A_{n+2}\right) \\ &= e^{-\lambda t}e^{-\mu\tau}\sum_{n=0}^{\infty}b_n\left[a_n + a_{n+1} + \left(1-\frac{\lambda}{\mu}\right)A_{n+2}\right] \\ &= e^{-\lambda t}e^{-\mu\tau}\sum_{n=0}^{\infty}b_n\left[X_{n-1} - a_{n-1} + \frac{(\lambda\tau)}{n+1}a_n\right]\end{aligned} \tag{B.3}$$

So the factors $S_n$ required to evaluate the infinite sums can be calculated in a rather simple and straightforward way by using in the $n$–th step the values of $a_{n-1}$, $b_{n-1}$ and $X_{n-1}$ as well as constants $(\lambda t)$, $(\mu\tau)$ and $(\lambda\tau)$ and setting

$$b_n = \frac{(\lambda t)}{n} b_{n-1}, \qquad a_n = \frac{(\mu\tau)}{n} a_{n-1}$$

$$X_n = X_{n-1} - a_{n-1} + \frac{(\lambda\tau)}{n+1} a_n, \qquad S_n = b_n X_n$$

Next, let us now determine $V(\lambda, T, v_0; t)$ by means of equation (3). Equation (1) shows that it is satisfactory to evaluate the integral

$$\int_0^t \pi(v_0, x) dx.$$

For the special case of exponential service times, equation (B.2) directly yields

$$\int_0^t \pi(v_0, x) dx = \int_{v_0}^t \pi(v_0, x) dx$$

$$= e^{\mu v_0} \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} \sum_{j=n}^{\infty} \frac{\mu^j}{j!} \sum_{k=0}^{j} \binom{j}{k} (-v_0)^k \int_{v_0}^t (e^{-\alpha x} x^{n+j-k}) dx -$$

$$-e^{\mu v_0} \frac{\lambda}{\mu} \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} \sum_{j=n+2}^{\infty} \frac{\mu^j}{j!} \sum_{k=0}^{j} \binom{j}{k} (-v_0)^k \int_{v_0}^t (e^{-\alpha x} x^{n+j-k}) dx \tag{B.4}$$

where $\tau$ is defined as introduced above and $\alpha := \lambda + \mu$. Evaluation of both integrals in (B.4) implies:

$$\int_0^t \pi(v_0, x) dx =$$

$$= e^{\mu v_0} \frac{1}{\alpha} \sum_{n=0}^{\infty} \left(\frac{\lambda}{\alpha}\right)^n \sum_{j=n}^{\infty} (-\mu v_0)^j \sum_{k=0}^{j} \binom{n+k}{k} \frac{1}{(j-k)!} \frac{1}{(-\alpha v_0)^k} \sum_{i=0}^{n+k} \frac{\alpha^i}{i!} z(i)$$

$$-e^{\mu v_0} \frac{1}{\alpha} \frac{\lambda}{\mu} \sum_{n=0}^{\infty} \left(\frac{\lambda}{\alpha}\right)^n \sum_{j=n+2}^{\infty} (-\mu v_0)^j$$

$$\times \sum_{k=0}^{j} \binom{n+k}{k} \frac{1}{(j-k)!} \frac{1}{(-\alpha v_0)^k} \sum_{i=0}^{n+k} \frac{\alpha^i}{i!} z(i) \tag{B.5}$$

with the definition $z(i) := e^{-\alpha v_0} v_0^i - e^{-\alpha t} t^i$.

After some algebraic manipulations, equation (B.5) can be reduced to

$$\int_0^t \pi(v_0,x)dx =$$

$$-\frac{1}{\alpha}e^{-\lambda v_0}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\alpha}\right)^n h(n,v_0;n) - \frac{1}{\alpha}e^{\mu v_0}e^{-\alpha t}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\alpha}\right)^n h(n,t;n)$$

$$-\frac{\lambda}{\alpha\mu}e^{-\lambda v_0}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\alpha}\right)^n h(n+2,v_0;n)$$

$$+\frac{\lambda}{\alpha\mu}e^{\mu v_0}e^{-\alpha t}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\alpha}\right)^n h(n+2,t;n) \qquad \text{(B.6)}$$

where the auxiliary function $h(m,y;n)$ is defined as follows

$$h(m,y;n) := \sum_{j=m}^{\infty}(-\mu v_0)^j \sum_{k=0}^{j}\binom{n+k}{k}\frac{1}{(j-k)!}\frac{1}{(-\alpha v_0)^k}\sum_{i=0}^{n+k}\frac{(\alpha y)^i}{i!}$$

To simplify numerical evaluation (B.6) can be transformed into the following form

$$\int_0^t \pi(v_0,x)dx =$$

$$= \frac{1}{\alpha}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\alpha}\right)^n g(n;n) + \frac{1}{\alpha}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\alpha}\right)^n g(n+1;n)$$

$$+ \left(1-\frac{\lambda}{\mu}\right)\frac{1}{\alpha}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\alpha}\right)^n \sum_{j=n+2}^{\infty} g(j;n) \qquad \text{(B.7)}$$

with the auxiliary function

$$g(j;n) = g(j;n,\lambda,\mu,t,v_0) := (-\mu v_0)^j \sum_{k=0}^{j}\binom{n+k}{k}\frac{1}{(j-k)!}\frac{1}{(-\alpha v_0)^k}e^{\mu v_0}$$

$$\times \sum_{i=0}^{n+k}\left(e^{-\alpha v_0}\frac{(\alpha v_0)^i}{i!} - e^{-\alpha t}\frac{(\alpha t)^i}{i!}\right).$$

It can be verified that as expected in the two special cases $\lambda = 0$ and $\lambda \to \infty$ the final result (B.7) degenerates to

$$\lim_{\lambda\to 0}\int_0^t \pi(v_0,x)dx = t - v_0$$

and

$$\lim_{\lambda\to\infty}\int_0^t \pi(v_0,x)dx = 0$$