Nov 76

Modelling and Analysis of Store-And-Forward Data Switching Centres with Finite Buffer Memory and Acknowledgement Signalling

Werner Bux
University of Stuttgart, Stuttgart, Federal Republic of Germany

ABSTRACT

In this paper a queueing model for a data switching centre within a store-and-forward switching network is developed, in order to provide a tool for the dimensioning of such networks. The model includes the finite buffer memory and the control processor of the switching centre, as well as the data channels (transmission time plus propagation delay). Moreover, acknowledgement signalling is considered because transmitted packets are stored in buffer until an acknowledgement is received. Unacknowledged packets are retransmitted after a constant time-out.

For exponentially distributed packet lengths a closed form solution for the equilibrium distribution of queue sizes is derived. This allows to evaluate performance values of the system (buffer overflow probability, flow times etc.). For constant packet lengths a simple approach is given for estimating the most important performance values. The approximation is checked by simulation. Moreover an algorithm is proposed to analyze iteratively whole networks with the aid of the developed model.

1. INTRODUCTION

The dimensioning of network devices (buffer memories, control processors, data channels) for a given traffic load is an important task in the development of store-and-forward switching data networks.

At best, the characteristic values of these devices should be balanced in such a way that a prescribed performance is achieved with a minimum amount of costs. Dimensioning is possible only, if the most important performance values are known, namely, buffer overflow probability, buffer load, mean time of packets in buffer, data throughput.

Section 2 of the paper contains a brief description of the structure and operating mode of store-and-forward switching networks. In Section 3 a basic model for a switching centre and its environment is developed in detail. In Section 4 an analysis of the data flow within the presented model is performed and an algorithm is proposed how to apply the model for analyzing whole networks. Using a number of diagrams some typical numerical results are discussed in Section 5.

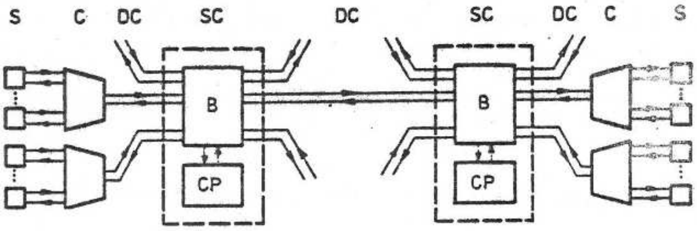
2. STRUCTURE AND OPERATING MODE OF STORE-AND-FORWARD SWITCHING NETWORKS

2.1 NETWORK PROPERTIES

In store-and-forward switching networks blocks of data are stored completely in any intermediate switching centre and then sent forward to the next node. The main components of such a network are shown in Fig.1.

For security reasons a copy of any transmitted block of data is held within the buffer memory until the succeeding node has positively acknowledged its receipt. We will use the term "packet" for the block of data handled by the network although most results are valid for message switching networks, too.

The data channels between the switching centres or between switching centres and data concentrators are assumed to operate in a full duplex manner, so that each node can simultaneously send and receive packets on all data channels to and from other nodes.



S DATA SUBSCRIBER
C DATA CONCENTRATOR
DC DATA CHANNELS
SC DATA SWITCHING CENTRE
B BUFFER MEMORY
CP CONTROL PROCESSOR

Fig.1. Components of a store-and-forward switching network

2.2 OPERATING MODE OF SWITCHING CENTRES

The data flow through a switching centre is shown in Fig.2 and is described as follows:

Storing Packets

Incoming packets are stored in the buffer memory ①. Usually the buffer memory is partitioned into a number of fixed length buffers, each of which is used in order to store a single packet.

When the first character of a packet is received, a complete buffer is occupied even if the packet is shorter. This is a simple and widely used method for buffer allocation in store-and-forward switching systems [1], [2], [3].

If the buffer memory is full, incoming packets are rejected ②. When the end of an incoming packet is detected a pointer is placed on the queue in front of the control processor ③. All incoming packets are checked for errors by an error detecting code. If a packet is received in error it is discarded ④.

Processing Packets

After a possible waiting time a correctly received packet is served by the control processor ⑤.

The control processor determines the next node the packet will traverse on the path to its destination. This is done with the aid of a routing procedure utilizing the destination address given in the packet header. Then a pointer is put on the queue in front of the output channel connected to that next node ⑥.

Transmitting Packets

When the outgoing channel has become free the packet is transmitted ⑦ but a copy is still held in the buffer memory until a positive acknowledgement is received. If the packet is not acknowledged within a certain time-out τ, it is retransmitted ⑧.

Acknowledgement Handling

If the packet is correctly received by the neighbour node then a positive acknowledgement is constructed ⑨ and sent

Beitrag des Instituts für
Nachrichtenvermittlung und Datenverarbeitung der
Universität Stuttgart zum 8th International Teletraffic
Congress Melbourne vom 10.-17. November 1976

back to the preceding node (10) . Neighbour nodes in this context may be other switching centres and data concentrators, as well. When the switching centre receives an acknowledgement it deletes its copy of the corresponding packet (11) .

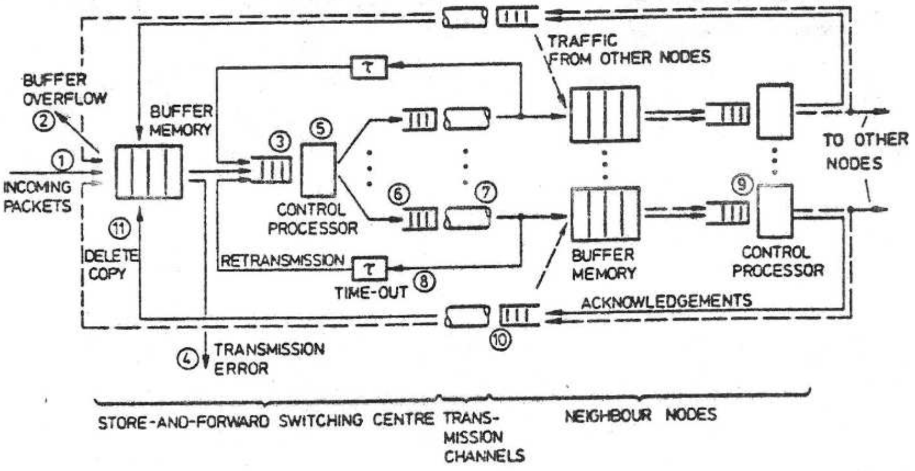


Fig.2. Data flow in a store-and-forward switching centre

3. MODELLING

3.1 BASIC MODEL OF A SWITCHING CENTRE

A queueing model to describe the data flow in a store-and-forward switching centre is shown in Fig.3.

Packet arrivals from all incoming data channels are combined in one arriving process with mean arrival rate λ (1) . (Therefore, the dashed lines in Fig.3 do not end at the buffer memory like in Fig.2.) Each packet buffer is modelled by one server of an n-server loss stage. Such a server is busy during the input time of a packet (mean input time h_1), but it remains occupied, even when the complete packet is stored, unless the packet was received in error. The latter happens with probability p_0 (4) . Should all buffers be occupied arriving packets are rejected (2) .

Correctly received packets are served by the control processor (mean service time h_2) (3) , (5) in the order of arrival.

After leaving the control processor the traffic is divided into R streams according to fixed probabilities $p_1, \ldots, p_R$ which represent the routing decisions to the R neighbour nodes (6) . The packets are queued in front of the outgoing data channel until transmission to the neighbour node takes place according to a first-come-first-served discipline (mean transmission time over the r-th channel h_{3r}, $r \in \{1,\ldots,R\}$) (7) .

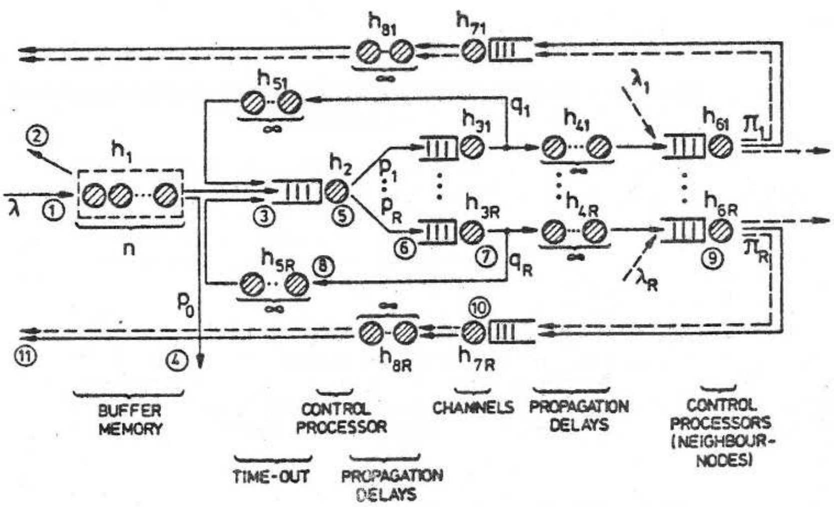


Fig.3. Basic queueing model of a store-and-forward switching centre

After some propagation delay h_{4r} modelled by infinite server stations the packets are stored in the buffer memory of the r-th neighbour node.

Buffer overflow in the neighbour nodes and transmission errors are considered to occur with fixed probabilities $q_1, \ldots, q_R$. How these probabilities could be determined iteratively is shown in Section 4.4. Unacknowledged packets are retransmitted after a constant time-out $\tau = h_{5r}$ (8) .

In the neighbour node the acknowledgement for a successfully received packet is constructed (mean service time h_{6R}) (9) and placed on the queue of the back channel (10) . Of course, the considered acknowledgements have to compete there with traffic from other sources. The impact of this traffic on the situation in the considered node is taken into account by introducing external arrival rates $\lambda_1, \ldots, \lambda_R$ to the respective stations of the model. The rate of external traffic which flows from neighbour node r back to the considered switching centre is defined as $\pi_r \cdot \lambda_r$. After transmission of the acknowledgement (mean transmission time h_{7r}, propagation delay h_{8r}) the occupied server in the first stage of the model is freed (11) .

3.2 MODIFICATIONS OF THE MODEL

The described basic mechanism of packet handling in store-and-forward switching centres can be realized in many different ways. These specific realizations will, of course, lead to different models of the systems. Depending on particular realizations it may be necessary to modify, extend, or specify the presented basic model.

For some of the following modifications models are already developed [4] , [5] , [6] , for others investigations are currently performed : various buffer allocation schemes, modified acknowledgement signalling (negative acknowledgements), special flow-control and routing strategies, priorities.

4. ANALYSIS

4.1 ASSUMPTIONS

We shall investigate our model under the following assumptions :

1) Poisson input processes
2) exponentially distributed service times of control processors
3) exponentially distributed packet lengths or constant packet lengths, respectively
4) general Erlangian distributed time-outs and propagation delays.

These assumptions are justified by the following arguments:

1) The total input process to a switching node normally results from a large number of independent traffic sources; hence it can be described with sufficient accuracy by a Poisson process.

2) Normally, packet processing (routing) is done with a relatively low priority. Hence, the service times of packets vary by virtue of a random number of short interruptions. Inaccuracies in describing the service time within the processor by an exponential distribution will be of little impact on global performance values (e.g. buffer overflow probability), as long as the processor is fast compared to the data channels.

3) As long as the prescribed maximum packet length is not too short compared with the mean length of subscriber messages, real packet lengths will vary over a wide range [7] . In this case, the exponential assumption will be suitable.

If we have a relatively short maximum packet length then the exponential distribution is only a rough approximation of real packet lengths, so that the results obtained under this assumption will usually overestimate waiting times, flow times, and buffer overflow. Then it is more accurate to assume constant packet lengths. With these two "extreme" assumptions the performance of most real systems could be estimated.

4) With the general Erlangian distribution the constant time-outs and propagation delays can be approximated with any desired accuracy.

4.2 EXPONENTIALLY DISTRIBUTED PACKET LENGTHS

For a certain class of queueing networks it is well-known that a product form solution for the equilibrium state probabilities holds [8] , [9] .

Such networks consist of an arbitrary number of queueing stations. Requests travel through the network according to transition probabilities. For different classes of requests, different transition probabilities may be prescribed. Changes of class membership are allowed when leaving a station.

Service times at first-come-first-served single server queues have to be exponentially distributed, whereas service times at pure delay stations ("infinite server" stations) may be general Erlangian distributed (rational Laplace transform of service time distribution).

In the case of an open network, external arrival processes have to be Poissonian; the instantaneous mean arrival rate may in a particular way be dependent on the actual number of requests within the network. (For a detailed description of these networks the reader is referred to reference [9] .)

Although the conditions for the arrival and service processes are met by our assumptions, this solution seems to be not *a priori* applicable to our problem because of two facts: 1) the buffer overflow and 2) the occupation of a buffer when a pointer belonging to the packet, stored within the buffer, is placed on the processor queue, or when the buffer contains a copy of an already transmitted but unacknowledged packet.

In the following section we will show, however, that our model can be mapped into an equivalent system without this blocked state of buffers but a state-dependent arrival rate of packets.

4.2.1 OUTLINE OF SOLUTION

For the sake of simplicity, we will derive the solution for a simplified version of our general model (cf. Fig.4). The solution for the general model can be found analogously; it is given in Section 4.2.2 . The considered system consists of 10 stations numbered as depicted in the figure. Obviously, the simplified system corresponds to the model of a switching centre in Fig.3 with two neighbour nodes where propagation delays are neglected.

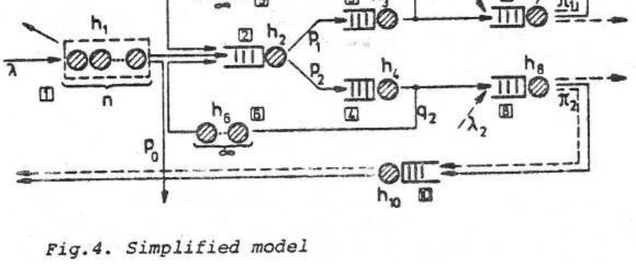


Fig.4. Simplified model

Request Classes

In order to know the number of requests which are stored in a buffer of the considered switching node, we have to distinguish between two classes of requests within our model.

All requests which enter the queueing network via station 1 (i.e. which occupy a buffer in the considered switching node) belong to request class 1; requests which enter the network via stations 7 or 8 (i.e. which do not momentarily occupy a buffer in the considered node) belong to request class 2. According to this distinction we define the following random variables:

X_1 random number of requests in station 1 corresponding to *momentarily incoming* packets (class 1 requests)

X_i random number of requests in station i ($i \varepsilon \{2,\dots,10\}$) which entered the queueing network via station 1 (class 1 requests)

Y_i random number of requests in station i ($i \varepsilon \{7,\dots,10\}$) which entered the queueing network via stations 7 or 8 (class 2 requests).

Transition Matrix

The requests travel through our queueing network according to transition probabilities. In the general case, requests of a certain class r which complete their service at station i will next go to station j and change their class membership to s with probability $p_{(ir),(js)}$.

No class changes are necessary in our basic model, therefore all $p_{(ir),(js)}$ with $r \neq s$ are set to zero.

The values for the other transition probabilities must be chosen according to the probabilities p_0, p_1, p_2, q_1, q_2, π_1, π_2 in Fig.4. (For example, the transition probability $p_{(21),(31)}$ is equal to p_1.)

The transition matrix $P = p_{(ir),(js)}$ in our model has the following important property:
The Markov chain defined by P is decomposable into two ergodic subchains. In more detail, the sets of states in these subchains are:

$$\begin{aligned} E_1 &= \{(i1) \mid i \,\varepsilon \{1,\dots,10\}\} \\ E_2 &= \{(i2) \mid i \,\varepsilon \{7,\dots,10\}\} \end{aligned} \tag{1}$$

Equivalent Network

As one can see our queueing network can be described equivalently by deleting the blocked state of servers in station 1 (cf. the definition of X_1) *but* introducing a state-dependent mean arrival rate λ^* of class 1 requests defined as follows:

$$\lambda^* = \begin{cases} \lambda \text{ if } \sum_{i=1}^{10} x_i < n \\ 0 \text{ if } \sum_{i=1}^{10} x_i \geq n \end{cases} \tag{2}$$

This corresponds exactly to the real input process of the finite buffer memory. Hence, the mean arrival rate λ^* is a function only of the number of requests in subchain E_1. This fact is a necessary condition for the validity of the solution, presented in the following.

Class 2 requests from external sources entering the network via stations 7 or 8 are assumed to arrive with constant rates λ_1 or λ_2, respectively.

Average Number of Visits to the Stations

In order to get the joint equilibrium distribution of queue sizes we have to determine the average number of visits to the 10 stations of our queueing network for class 1 requests (e_{i1}) and class 2 requests (e_{i2}). These values are given by the following two sets of equations corresponding to subchains E_1, E_2.

Subchain E_1:

$$\sum_{i=1}^{10} e_{i1}\, p_{(i1),(j1)} + u_{j1} = e_{j1} \quad (j \varepsilon \{1,\ldots,10\})$$

$$\text{where} \quad u_{j1} = \begin{cases} 1 & \text{if } j=1 \\ 0 & \text{if } j \neq 1 \end{cases} \tag{3}$$

Subchain E_2:

$$\sum_{i=7}^{10} e_{i2}\, p_{(i2),(j2)} + u_{j2} = e_{j2} \quad (j \varepsilon \{7,\ldots,10\}) \tag{4}$$

$$\text{where} \quad u_{j2} = \begin{cases} \lambda_1/(\lambda_1+\lambda_2) & \text{if } j=7 \\ \lambda_2/(\lambda_1+\lambda_2) & \text{if } j=8 \\ 0 & \text{if } j \notin \{7,8\} \end{cases}$$

(u_{jk} is the probability that a class k request enters the network via station j; therefore the above given values for u_{j1}, u_{j2} are obvious.)

We don't want to give the explicit expressions for e_{i1}, e_{i2} in this context, because they appear implicitly in the solution (6) of the general case.

Once the values e_{i1}, e_{i2} are known, then the joint equilibrium distribution of queue sizes can be found in a straightforward manner [9].

The state probabilities become:

$$\begin{aligned} p(S_o) &= p(x_1,\ldots,x_{10},y_7,\ldots,y_{10}) \\ &= P(X_1=x_1,\ldots,X_{10}=x_{10},Y_7=y_7,\ldots,Y_{10}=y_{10}) \\ &= C_o\, \lambda^{x_1+\ldots+x_{10}} (\lambda_1+\lambda_2)^{y_7+\ldots+y_{10}} \frac{(e_{11}h_1)^{x_1}}{x_1!} \prod_{i=2}^{4} (e_{i1}h_i)^{x_i} \cdot \\ &\quad \cdot \prod_{i=5}^{6} \frac{(e_{i1}h_i)^{x_i}}{x_i!} \prod_{i=7}^{10} \binom{x_i+y_i}{x_i} (e_{i1}h_i)^{x_i} (e_{i2}h_i)^{y_i} \end{aligned} \tag{5}$$

$$\sum_{S_o \varepsilon \Phi_o} p(S_o) = 1$$

$$\Phi_o = \left\{ (x_1,\ldots,x_{10},y_7,\ldots,y_{10}) \mid x_i, y_i \,\varepsilon\, N_o \wedge \sum_{i=1}^{10} x_i \leqq n \right\}$$

4.2.2 SOLUTION FOR THE GENERAL MODEL

The steady state probabilities for our general model from Fig.3 are found in a way, similar to that explained in the preceding section. Therefore, we can present the final result immediately in Eq. (6). (Note that the subscripts of variables x (class 1 requests) and y (class 2 requests) correspond to the subscripts given in Fig.3.)

$$\begin{aligned} p(S_1) &= p(x_1,x_2,x_{31},\ldots,x_{3R},x_{41},\ldots,x_{4R},x_{51},\ldots,x_{5R},x_{61},y_{61},\ldots \\ &\qquad \ldots,x_{6R},y_{6R},x_{71},y_{71},\ldots,x_{7R},y_{7R},x_{81},y_{81},\ldots,x_{8R},y_{8R}) \\ &= C_1 \frac{a_1^{x_1}}{x_1!} a_2^{x_2} \prod_{r=1}^{R} \left\{ a_{3r}^{x_{3r}} \frac{a_{4r}^{x_{4r}}}{x_{4r}!} \frac{a_{5r}^{x_{5r}}}{x_{5r}!} \binom{x_{6r}+y_{6r}}{x_{6r}} a_{6r}^{x_{6r}} b_{6r}^{y_{6r}} \right\} \cdot \\ &\quad \cdot \prod_{r=1}^{R} \left\{ \binom{x_{7r}+y_{7r}}{x_{7r}} a_{7r}^{x_{7r}} b_{7r}^{y_{7r}} \frac{a_{8r}^{x_{8r}}}{x_{8r}!} \frac{b_{8r}^{y_{8r}}}{y_{8r}!} \right\} \end{aligned}$$

$$\sum_{S_1 \varepsilon \Phi_1} p(S_1) = 1 \tag{6}$$

$$\Phi_1 = \left\{ (x_1,\ldots,y_{8R}) \mid x_1,\ldots,y_{8R} \,\varepsilon\, N_o \wedge x_1+x_2+\sum_{j=3}^{8} \sum_{r=1}^{R} x_{jr} \leqq n \right\}$$

$$\left.\begin{array}{l} a_1=\lambda h_1 \quad a_2=\beta\lambda h_2 \quad a_{3r}=\beta p_r \lambda h_{3r} \quad a_{5r}=\beta p_r q_r \lambda h_{5r} \\ a_{jr}=\beta p_r (1-q_r) \lambda h_{jr} \quad \text{for} \quad j \,\varepsilon\, \{4,6,7,8\} \\ b_{6r}=\lambda_r h_{6r} \quad b_{7r}=\pi_r \lambda_r h_{7r} \quad b_{8r}=\pi_r \lambda_r h_{8r} \end{array}\right\} \begin{array}{l} \text{for} \\ r \varepsilon \{1,\ldots,R\} \end{array}$$

$$\text{where} \quad \beta = (1-p_o)/(1-\sum_{r=1}^{R} p_r q_r)$$

We are mainly interested in performance values related to class 1 requests; hence it is useful to simplify the solution (6) by evaluating the marginal distributions:

$$\begin{aligned} &P(X_1=x_1,X_2=x_2,X_{31}=x_{31},\ldots,X_{3R}=x_{3R},\ldots,X_{81}=x_{81},\ldots,X_{8R}=x_{8R}) \\ &= p(x_1,x_2,x_{31},\ldots,x_{3R},\ldots,x_{81},\ldots,x_{8R}) = p(S_2) \end{aligned}$$

This is done, by summing the $p(S_1)$ over all possible values of $y_{5r}, y_{6r}, y_{7r}, y_{8r}$. It follows from Equation (6):

$$p(S_2) = C_2 \frac{a_1^{x_1}}{x_1!} a_2^{x_2} \prod_{r=1}^{R} \left\{ a_{3r}^{x_{3r}} \frac{a_{4r}^{x_{4r}}}{x_{4r}!} \frac{a_{5r}^{x_{5r}}}{x_{5r}!} \left[\frac{a_{6r}}{1-b_{6r}}\right]^{x_{6r}} \left[\frac{a_{7r}}{1-b_{7r}}\right]^{x_{7r}} \frac{a_{8r}^{x_{8r}}}{x_{8r}!} \right\}$$

$$\sum_{S_2 \varepsilon \Phi_2} p(S_2) = 1 \tag{7}$$

$$\Phi_2 = \left\{ (x_1,\ldots,x_{8R}) \mid x_1,\ldots,x_{8R} \,\varepsilon\, N_o \wedge x_1+x_2+\sum_{j=3}^{8} \sum_{r=1}^{R} x_{jr} \leqq n \right\}$$

4.2.3 PERFORMANCE VALUES

With the knowledge of the state probabilities, now the performance values can be determined:

Probability of Buffer Overflow

$$B = P(X_1+X_2+\sum_{j=3}^{8} \sum_{r=1}^{R} X_{jr} = n) = \sum_{S_2 \varepsilon \Psi(n)} p(S_2) \tag{8}$$

$$\text{where} \quad \Psi(\xi) = \left\{ S_2 \mid S_2 \varepsilon \Phi_2 \wedge x_1+x_2+\sum_{j=3}^{8} \sum_{r=1}^{R} x_{jr} = \xi \right\}$$

Load of Buffer Memory

$$Y_b = \sum_{x=0}^{n} [x \cdot P(X_1+X_2+\sum_{j=3}^{8} \sum_{r=1}^{R} X_{jr} = x)] = \sum_{x=0}^{n} [x \cdot \sum_{S_2 \varepsilon \Psi(x)} p(S_2)] \tag{9}$$

Mean Time of Packets in Buffer

$$t_b = \frac{Y_b}{\lambda(1-B)} \tag{10}$$

Packet Throughput Rate

$$\lambda_s = \lambda(1-B) \tag{11}$$

4.2.4 EXTENSION

We have assumed that the mean input time is equal for all packets. If this is not true (e.g. because of different speeds of incoming data channels) the model can be extended by introducing c additional request classes each of which having a certain distinct mean input time $h_{11},\ldots,h_{1c}$ and mean arrival rate $\lambda_{11},\ldots,\lambda_{1c}$.

In this case, the term $a_1^{x_1}/x_1!$ in Equation (6) is split up into a product of c factors

$$\prod_{\nu=1}^{c} \frac{(\lambda_{1\nu}\, h_{1\nu})^{x_{1\nu}}}{x_{1\nu}!}$$

where $x_{1\nu}$ denotes the number of instantaneously incoming packets having the same mean input time $h_{1\nu}$ (i.e. which belong to the same "input class" ν).

4.2.5 NUMERICAL EVALUATION

Because of the very large number of states in models for practical systems the numerical evaluation of the presented solution is only possible by applying special effective algorithms [10] , [11] .

The cited algorithms are valid for closed queueing networks but can be used here as well to determine performance values related to class 1 requests [6] .

4.3 CONSTANT PACKET LENGTHS

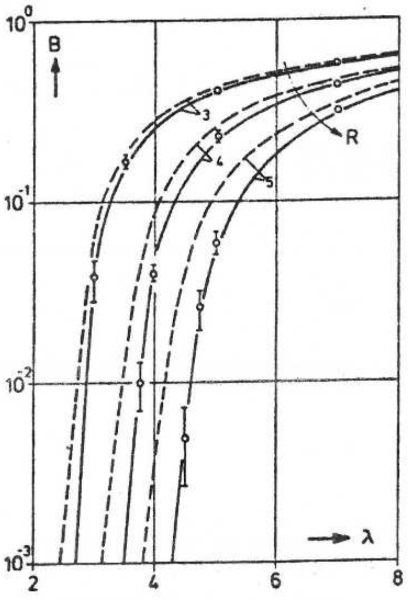
The assumption of constant packet lengths leads to tremendous theoretical difficulties in the investigation of such a complex queueing network. Approximate results could be obtained at least for the special case of an infinite buffer ($n \to \infty$) applying one of the following methods: diffusion approximation [12] , parametric analysis [13] , decomposition methods [14] .

To determine the value in which we are most interested, namely buffer overflow, a simple heuristic approach yields good results at least in the range of practical interest.

This approach is based on the fact that the service time distribution of the data channels influences the mean time, a packet is stored in buffer, and hence, the buffer overflow probability. This observation leads to the following approximation:

In a first step, we estimate the mean arrival rate Λ to a data channel by assuming exponentially distributed service times and applying our solution. Then we determine the mean service time h_f of an M/M/1 queue which yields the same mean flow time as an M/D/1 queue with the *prescribed* mean service time of the channel, when both queues Poisson traffic with mean arrival rate Λ is offered. With this "fictitious" mean service time h_f , determined for each channel separately, we evaluate the buffer overflow probability again with the aid of our solution and take this value as an approximation for the case of constant packet lengths.

An example for the accuracy of this quite simple approximation is given in Fig.5; simulation results in this diagram are shown with 95% confidence intervals.



Parameters:

$n=50$

$p_0=0.001$

$p_1=\ldots=p_R=1/R$

$q_1=\ldots=q_R=0.01$

$\lambda_1=\ldots=\lambda_R=0$

$h_1=1$

$h_2=0.02$

$h_{31}=\ldots=h_{3R}=1$

$h_{41}=\ldots=h_{4R}=0$

$h_{51}=\ldots=h_{5R}=2$

$h_{61}=\ldots=h_{6R}=0$

$h_{71}=\ldots=h_{7R}=0.2$

$h_{81}=\ldots=h_{8R}=0$ ·

--- exponentially distributed packet lengths (exact solution)

—— constant packet lengths (approximation)

ǫ constant packet lengths (simulation)

Fig.5. Buffer overflow probability B versus mean arrival rate λ for exponentially distributed or constant packet lengths, respectively. Parameter: number of outgoing data channels R

Of course, the approximation could be refined, if necessary, with a more accurate description of the input process for estimating the flow times through the queues with constant service times.

Numerous simulations showed that performance values with constant packet lengths are always superior to those with exponentially distributed packet lengths [6] , [15] . This evidently plausible result guarantees that the dimensioning of store-and-forward switching networks assuming exponentially distributed packet lengths yields values "located in the safe region".

4.4 ANALYSIS OF NETWORKS

As mentioned above the retransmission probabilities $q_1, \ldots, q_R$ do not only depend on the probability p_{Or} that a packet will have at least one error when transmitted over the r-th channel but also on the overflow probability B_r of neighbour node r, namely,

$$q_r = p_{Or} + B_r - p_{Or}B_r \qquad (12)$$

A whole network (or parts of it) can be analyzed by applying the following algorithm:

(i) set initial values for the retransmission probabilities q_r in any involved node

(ii) determine overflow probabilities B_r for any involved node

(iii) correct the retransmission probabilities according to equation (12) and the mean arrival rates, if necessary

(iv) repeat steps (ii) and (iii) until equation (12) is fulfilled for any involved node with the desired accuracy

(v) evaluate performance values with the determined retransmission probabilities for any involved node separately.

5. NUMERICAL RESULTS

Figs. 6,7,8,9,10,11 show typical numerical performance values obtained for exponentially distributed packet lengths. Those system parameters, which are common for all these diagrams, are summarized in Table 1, whereas specific parameters are given in the figures.

Figs. 6,7 show the relationship between mean arrival rate λ, buffer memory size n, or retransmission probability q, respectively, and the buffer overflow probability B.

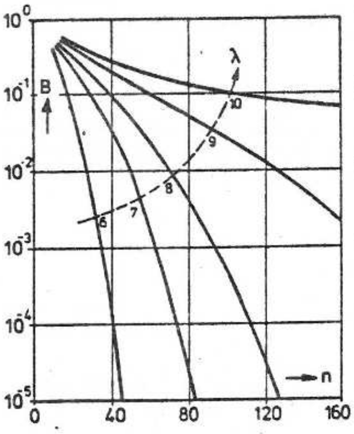


Fig.6. Buffer overflow probability versus buffer memory size n; parameter: mean arrival rate λ (q=0, h_2=0.02, h_p=0)

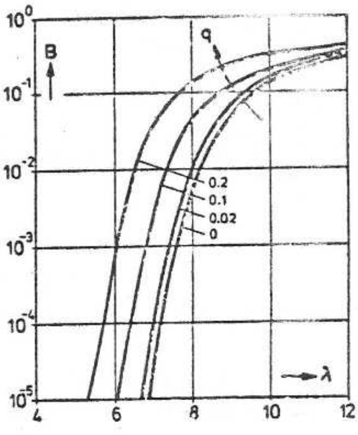


Fig.7. Buffer overflow probability B versus mean arrival rate λ; parameter: retransmission probability q (n=80, h_2=0.02, h_p=0)

R=10	p_0=0.001	p_1=...=p_{10}=0.1	q_1=...=q_{10}=q
		π_1=...=π_{10}=0.1	λ_1=...=λ_{10}=10
h_1=1	h_{31}=...=$h_{3,10}$=1	h_{41}=...=$h_{4,10}$=h_p	
	h_{51}=...=$h_{5,10}$=2	h_{61}=...=$h_{6,10}$=0.01	
	h_{71}=...=$h_{7,10}$=0.2	h_{81}=...=$h_{8,10}$=h_p	

Table 1. Common parameters for Figures 6,7,8,9,10,11

In Figs. 8,9 the mean time of packets in buffer t_b and the normalized load of the buffer memory Y_b/n are plotted versus the mean arrival rate λ, for several sizes of the buffer memory n.

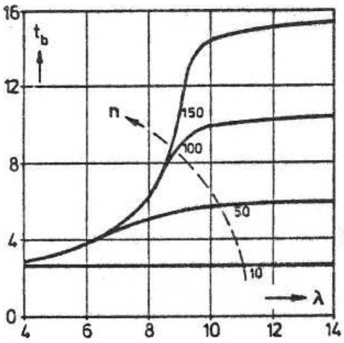


Fig.8. Mean time of packets in buffer t_b versus mean arrival rate λ; parameter: buffer memory size n (q=0, h_2=0.02, h_p=0)

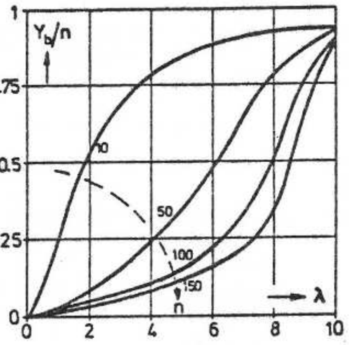


Fig.9. Normalized load of buffer memory Y_b/n versus mean arrival rate λ; parameter: buffer memory size n (q=0, h_2=0.02, h_p=0)

Fig.10 shows how the buffer overflow probability B depends on the mean service time h_2 of the control processor.

The influence of the propagation delay h_p on the buffer memory size n, necessary for achieving a prescribed buffer overflow probability B, is shown in Fig.11.

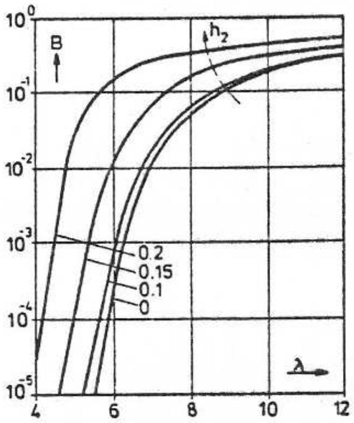


Fig.10. Buffer overflow probability B versus mean arrival rate λ; parameter: mean service time of the control processor h_2 (n=50, q=0.01, h_p=0)

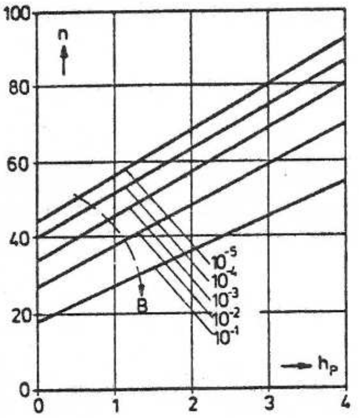


Fig.11. Buffer memory size n versus propagation delay h_p; parameter: buffer overflow probability B (λ=5, q=0, h_2=0.02)

6. CONCLUSION

With the aid of the developed model it is possible to analyze the fundamental traffic phenomena within store-and-forward switching centres. As indicated in Section 3.2 the basic model can be modified according to special realizations. Investigations of such modified models may perhaps be performed in a way similar to that described in Section 4 but they may as well lead to enormous analytical difficulties. Simulation will therefore also be a necessary and powerful tool in investigating the data flow within store-and-forward switching networks.

Performance values, obtained by investigations, like the presented one, should be useful as a guide for dimensioning such data networks.

ACKNOWLEDGEMENT

The author acknowledges with thanks the many helpful suggestions and comments of U. Herzog and P. Kühn. He is also grateful to the Federal Ministry of Research and Technology (BMFT) of the Fed. Rep. of Germany for supporting this work.

REFERENCES

[1] Davies,D.W., Barber,D.L.A.: Communications Networks for Computers. J.Wiley&Sons, London/New York, 1973.

[2] Heart,F.E., Kahn,R.E., Ornstein,S.M., Crowther,W.R., Walden,D.C.: The Interface Message Processor for the ARPA Computer Network. Proc. AFIPS SJCC 36(1970), pp. 551-567.

[3] Beeforth,T.H., Grimsdale,R.L., Halsall,F., Woolons, D.J.: Proposed Organization for Packet-Switched Data-Communication Networks. Proc. IEE 119(1972)12, pp. 1677-1682.

[4] Bux,W., Kühn,P.: Verkehrstheoretische Probleme beim Aufbau und Betrieb von Rechnerverbundsystemen. Monograph. Institute of Switching and Data Technics, University of Stuttgart, 1974.

[5] Bux,W., Kühn,P.: Untersuchungen zur verkehrsgerechten Dimensionierung von Speichervermittlungen. NTG-Fachtagung "Datennetze", Baden-Baden, 18.-20.2.1976 (NTG-Fachberichte Band 55, VDE-Verlag, Berlin, 1976, pp. 83-93).

[6] Bux,W., Korndörfer,V.: Entwurf und Analyse von Verkehrsmodellen für Speichervermittlungen. Monograph. Institute of Switching and Data Technics, University of Stuttgart, 1976.

[7] Kleinrock,L., Naylor,W.E.: On Measured Behavior of the ARPA Network. Proc. AFIPS NCC 43(1974), pp.767-780.

[8] Jackson,J.R.: Jobshop-Like Queueing Systems. Management Science 10(1963)1, pp. 131-142.

[9] Baskett,F., Chandy,K.M., Muntz,R.R., Palacios,F.G.: Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. J.ACM 22(1975)2, pp. 248-260.

[10] Buzen,J.P.: Computational Algorithms for Closed Queueing Networks with Exponential Servers. C.ACM 16(1973)9 pp. 527-531.

[11] Reiser,M., Kobayashi,H.: Horner's Rule for the Evaluation of General Closed Queueing Networks. C.ACM 18(1975)10, pp. 592-593.

[12] Kobayashi,H.: Application of the Diffusion Approximation to Queueing Networks. IBM Journ. of Res. and Dev. 19(1974)2, pp. 316-328.

[13] Chandy,K.M., Herzog,U., Woo,L.: Approximate Analysis of General Queueing Networks. IBM Journ. of Res. and Dev. 19(1975)1, pp. 43-49.

[14] Kühn,P.: Analysis of Complex Queueing Networks by Decomposition. 8th ITC, Melbourne, 1976.

[15] Bux,W., Krämer,W., Wucher,P.: Simulation von Warteschlangennetzen. Monograph. Institute of Switching and Data Technics, University of Stuttgart, 1975.