

This item is the archived peer-reviewed author-version of:

A Hessenberg Markov chain for fast fibre delay line length optimization

Reference:

Lambert Joke, Rogiest W., van Houdt Benny, Fiems D., Blondia Christian, Bruneel H.- *A Hessenberg Markov chain for fast fibre delay line length optimization*

Lecture notes in computer science - ISSN 1611-3349 - 5055(2008), p. 101-113

Handle: <http://hdl.handle.net/10067/705480151162165141>

A Hessenberg Markov chain for fast Fibre Delay Line length optimization

J. Lambert¹, W. Rogiest², B. Van Houdt¹, D. Fiems²,
C. Blondia¹, and H. Bruneel²

¹ PATS Research Group, IBBT & University of Antwerp,
Middelheimlaan 1, B-2020 Antwerp - Belgium,
{joke.lambert,benny.vanhoudt,chris.blondia}@ua.ac.be
² SMACS Research Group, TELIN Department, Ghent University,
Sint-Pietersnieuwstraat 41, B-9000 Gent - Belgium,
{wouter.rogiest,dieter.fiems,herwig.bruneel}@ugent.be

Abstract. In this paper we present an approach to compute the invariant vector of the $N + 1$ state Markov chain P presented in (Rogiest *et al.*, Lecture Notes in Computer Science, NET-COOP 2007 Special Issue, pp. 4465:185-194) to determine the loss rate of an FDL buffer consisting of N lines, by solving a related Hessenberg system (i.e., a Markov chain skip-free in one direction). This system is obtained by inserting additional time instants in the sample paths of P and allows us to compute the loss rate for various FDL lengths by solving a single system. This is shown to be especially effective in reducing the computation time of the heuristic LRA algorithm presented in (Lambert *et al.*, Proc. NAEC 2005, pp. 545-555) to optimize the FDL lengths, where improvements of several orders of magnitude can be realized.

1 Introduction

The rapid growth of the Internet traffic has resulted in an increasing demand for bandwidth as well as support for diverse service demands. As the electronic switches and routers are becoming the bottlenecks of the backbone network, these networks must evolve to new architectures based upon all optical switching. Optical burst switching (OBS) has been proposed [11], [13], [14] as an efficient and flexible switching method and it combines the benefits of optical circuit switching and optical packet switching. Although wavelength conversion reduces the need for OBS network buffering [7], contention can still arise when two or more data bursts arrive at the same output wavelength. In optical buffering, fibre delay lines (FDLs) are used to delay a burst for a specified amount of time, which is determined by the length of the delay line. Several architectures that make use of an FDL buffer have already been identified [3], [4], [8]. The main performance measure of an FDL buffer is its loss rate and analytic and simulation based results concerning the loss rates have been reported in [2], [5], [6], [9], [10], [12].

While the typical approach is to build an analytic model based on the evolution of the scheduling horizon, a more effective modeling approach was developed in [1] by focusing on the waiting time only. This model is applicable for Bernoulli arrival processes, general burst size distributions and arbitrary sets of FDLs. As the waiting time of a burst corresponds to the length of one of the FDLs, the computational complexity in [12] depended solely on the number of FDLs N , which is typically small, whereas with the scheduling-horizon-based analysis the complexity was a function of the length of the longest delay line (in slots).

Eventhough N is typically small, the approach taken in [12] has to be re-performed for each set of FDL lengths, which can be very time-consuming in an FDL length optimization procedure. In this paper we develop a Hessenberg reduction of the model presented in [12], which not only allows us to assess the loss rate more quickly for a single set of FDL lengths, but also enables us to compute the loss rate for various FDL length settings at the same time. This significantly reduces the time required to perform an FDL length optimization.

The remainder of this paper is structured as follows. Section 2 introduces the general notion of FDL buffers. We present the traffic model and the corresponding Markov chain in Section 3. From this we derive the transition probabilities and loss probabilities. Section 4 makes use of the memoryless nature of the arrival process to reduce the Markov chain to a Hessenberg, i.e., skip-free in one direction, Markov chain. In Section 5 we present a comparative study between the different solution methods discussed in this paper.

2 Fibre delay lines

In this paper, we study a single Wavelength Division Multiplexing (WDM) channel and assume contention for it is resolved by means of a Fibre Delay Line (FDL) buffer, which can delay, if necessary, data packets, called optical bursts (OBs), until the channel becomes available again. Unlike conventional electronic buffers, however, it cannot delay bursts for an arbitrary period of time, but it can only realize a discrete set of N delay values. In [9, 2] it is shown that this leads to voids on the outgoing channel. We do not attempt to fill these voids (as this would alter the order of the bursts). We denote the N delay values as $a_1 \leq a_2 \leq \dots \leq a_N$ and define $a_0 = 0$. Traditionally, there are two possibilities for the delay values, either all fibres have the same length, i.e., $a_k = A$ with $k = 1, 2, \dots, N$, or the values are equidistant, i.e., $a_k = kD$ with $k = 1, \dots, N$, where D is termed the buffer granularity. The analysis in this paper, however, is done for arbitrary lengths. We define the set \mathcal{A} as $\mathcal{A} = \{a_0, a_1, \dots, a_N\}$.

Define the scheduling horizon H at time t as the time difference between t' and t , where t' ($t' \geq t$) is the earliest time at which all OBs present at time t will have left the system. When the k -th burst sees a scheduling horizon $H_k > 0$ upon arrival, with $a_i < H_k \leq a_{i+1}$ for some $0 \leq i \leq N$ (and with $a_{N+1} = \infty$), it will have to be delayed by a_{i+1} time units (if $i < N$, otherwise it is dropped), possibly creating a void on the outgoing channel (which occurs if $i < N$ and $a_i < H_k < a_{i+1}$). Bursts that observe a scheduling horizon $H_k = 0$ do not

experience any delay. Since the length of the longest delay line corresponds to the maximum achievable delay a_N , an OB will be dropped (that is, lost) if the burst sees a scheduling horizon larger than a_N upon arrival.

3 Analysis

As the reduction procedure proposed in this paper heavily builds on the approach developed in [12], we start with a (brief) discussion of this approach.

3.1 Traffic model

Both the model developed in [12] and the reduction model assume Bernoulli arrivals, i.e., a new OB arrival occurs in a slot with probability p independently from slot to slot. In reality, arrivals destined for an output port may not form a Bernoulli process and the losses do not occur uniformly over time, typically the majority of the losses may be expected to occur in bursts during so-called overload periods. Our Bernoulli assumption, with p large such that the load is close to or above one, therefore corresponds to assuming that these overload periods have a substantial duration and the arrivals during such a period may be approximated by a Bernoulli process. When we dimension the FDL buffer in this manner, we therefore aim for the best possible structure to minimize losses during overload periods as these contain most of the losses.

Arriving bursts are either accepted upon arrival or dropped. We number the OBs in the order at which they arrive, but only assign an index to the OBs that are accepted. With each accepted OB k , we associate an inter-arrival time T_k , that captures the time between the k th arrival and the next, being the arrival of (i) burst $k + 1$, if this next burst is accepted, or (ii) a burst without number, if this next burst is dropped. For the assumed Bernoulli arrival process, these inter-arrival times are independent and geometrically distributed (with mean $1/p$):

$$P(T_k = n) = t_n = q^{n-1}p \quad n \geq 1, \quad (1)$$

where we denote $q = 1 - p$. With each accepted burst, we also associate a burst size B_k and we assume that consecutive OB lengths are uncorrelated with a common distribution. Therefore, we can define general probabilities:

$$P(B_k = n) = b_n \quad n \geq 1, \quad (2)$$

with $0 \leq b_n \leq 1$ and $\sum_{n=1}^{\infty} b_n = 1$. To simplify the notations, we will write $\bar{b}_n = \sum_{k=1}^n b_k$ and $\bar{t}_n = \sum_{k=1}^n t_k$.

3.2 Markov chain

As mentioned, we will track the system's performance by means of the waiting time of an OB. We associate the waiting time W_k with the k th burst and define it as the time between the acceptance of OB k and the start of its transmission.

As described in [12] the next burst is either accepted or dropped resulting in two possible scenarios for the waiting time transitions. These two scenarios are based on the horizon value as seen by the next arriving OB. Let \bar{H}_k be the value of the scheduling horizon as seen by the burst, following burst k . Note that this is not the value of the horizon as seen by the k -th arrival. This horizon value is given by

$$\bar{H}_k = (W_k + B_k - T_k)^+, \quad (3)$$

where $(x)^+$ denotes $\max(x, 0)$.

Scenario 1 We have a lossless transition in case $\bar{H}_k \leq a_N$ with the result that the waiting time of the accepted burst equals

$$W_{k+1} = \lceil W_k + B_k - T_k \rceil_{\mathcal{A}}, \quad (4)$$

where we adopted the relation $\lceil x \rceil_{\mathcal{A}} = \inf\{y \in \mathcal{A} \mid y \geq x\}$, for $x \leq a_N$.

Scenario 2 The burst will be dropped if $\bar{H}_k > a_N$. In this case the burst following burst k is not assigned an index, and possibly, even more bursts are dropped before burst $k + 1$ is accepted. As a result of this (and of the memoryless nature of the arrival process), the waiting time of burst $k + 1$ no longer relates to W_k and is defined as

$$W_{k+1} = \lceil a_N - \tilde{T}_k \rceil_{\mathcal{A}}, \quad (5)$$

where \tilde{T}_k is the time until the arrival of the next accepted burst, i.e., the reactivation time. In case of geometric inter-arrival times, \tilde{T}_k has a shifted geometric distribution:

$$P(\tilde{T}_k = n) = \tilde{t}_n = q^n p \quad n \geq 0. \quad (6)$$

We are now able to build up the Markov chain, associated with the evolution of the waiting times. Remark that the waiting time can only take on $N + 1$ different values $a_i \in \mathcal{A}$, therefore, the Markov chain consists of $N + 1$ states. This chain is characterized by a transition matrix P with probabilities $p_{n,m}$:

$$p_{n,m} = P(W_{k+1} = a_m \mid W_k = a_n) \quad 0 \leq n, m \leq N. \quad (7)$$

Corresponding to the scenarios discussed above, the probabilities $p_{n,m}$ can be split up into two separate contributions, i.e., the first term corresponds to the event of a lossless transition and the second term corresponds to the event of a transition with loss:

$$\begin{aligned} p_{n,m} &= P(a_n + B_k - T_k \leq a_N, a_m = \lceil a_n + B_k - T_k \rceil_{\mathcal{A}}) \\ &\quad + P(a_n + B_k - T_k > a_N, a_m = \lceil a_N - \tilde{T}_k \rceil_{\mathcal{A}}) \\ &= P(a_{m-1} - a_n < B_k - T_k \leq a_m - a_n) \\ &\quad + P(B_k - T_k > a_N - a_n) P(a_N - a_{m-1} > \tilde{T}_k \geq a_N - a_m), \end{aligned} \quad (8)$$

where $a_{-1} = -\infty$. In order to reduce the computation time it is useful to introduce

$$U_n = P(B_k - T_k \leq n) = \sum_{i=1}^n b_i(1 - q^{i-n-1}) + q^{-n-1}B(q), \quad (9)$$

where the sum over n disappears if $n \leq 0$ and $B(z)$ is the probability generating function of B_k evaluated in $z = q$, i.e., $B(z) = E[z^{B_k}] = \sum_{n=1}^{\infty} b_n z^n$. Adopting these notations, (8) can be expressed as

$$p_{n,m} = U_{a_m - a_n} - U_{a_{m-1} - a_n} + (1 - U_{a_N - a_n})(\bar{t}_{a_N - a_{m-1}} - \bar{t}_{a_N - a_m}).$$

The stationary probability vector $\pi = (\pi_0, \dots, \pi_N)$ satisfies $\pi = \pi P$ and $\pi e = 1$, where e is column vector with all entries equal to one.

3.3 Loss probability

Up to now, we considered only accepted OBs and even left the dropped bursts unnumbered. As a consequence the loss probability can be computed from the expected number of losses generated after each accepted packet. For geometric inter-arrival times the expected number of such losses is given by

$$E_{\text{loss}} = \sum_{k=1}^{\infty} \sum_{i=0}^N \pi_i p b_k (a_i + k - a_N - 1, 0)^+. \quad (10)$$

Namely, define the periods during which the system can accept new burst as available periods, and the periods during which the system drops arriving bursts as unavailable periods. Hence, $(a_i + k - a_N - 1, 0)^+$ in equation (10) denotes the length of the unavailable period, following an accepted burst, which had to wait a_i slots and which had a size equal to k . Finally, the loss probability is determined as follows

$$p_{\text{loss}} = \frac{E_{\text{loss}}}{1 + E_{\text{loss}}}. \quad (11)$$

4 Skip-free in one direction Markov chain

By exploiting the memoryless nature of the arrival process, we will reduce the Markov chain characterized by P , to a Markov chain that is skip-free to the left, i.e., the transition probability matrix \bar{P} of the new chain has the following structure:

$$\bar{P} = \begin{bmatrix} \bar{p}_{0,0} & \bar{p}_{0,1} & \bar{p}_{0,2} & \cdots & \bar{p}_{0,N} \\ \bar{p}_{1,0} & \bar{p}_{1,1} & \bar{p}_{1,2} & \cdots & \bar{p}_{1,N} \\ 0 & \bar{p}_{2,1} & \bar{p}_{2,2} & \cdots & \bar{p}_{2,M} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{p}_{N,N-1} & \bar{p}_{N,N} \end{bmatrix}. \quad (12)$$

We shall refer to the transition matrix P , defined in Section 3.2, as the dense matrix and to the transition matrix \bar{P} (equation (12)) as the Hessenberg matrix. Besides the computational advantage of the Hessenberg form, which allows us to compute the invariant vector of \bar{P} in $O(n^2)$ time, an even more important property of this reduction is that a_N , the length of the longest delay line, will only affect two entries in \bar{P} , being $\bar{p}_{N,N-1}$ and $\bar{p}_{N,N}$. This will allow us to express the invariant vectors $\bar{\pi}$ of \bar{P} for different a_N values in terms of the invariant vector of the system with $a_N = a_{N-1} + 1$. This feature will greatly simplify the optimization step of the LRA algorithm (see Section 5.1).

The idea behind the reduction works as follows. Suppose P makes a transition from state n to state m without a loss, where $a_m = \lceil a_n - T_k + B_k \rceil_{\mathcal{A}}$. Then, \bar{P} will first go from state n to state $n - s$, with $a_{n-s} = \lceil a_n - T_k \rceil_{\mathcal{A}}$ in s steps (from state n via $n - 1, n - 2$, etc. to $n - s$). Afterwards, it will jump from state $n - s$ to state m in one step. Thus, we insert s additional time instants for each lossless transition from state n to state m .

The reduction for transitions that do involve losses is somewhat more involving. As with the lossless transitions, the chain characterized by \bar{P} starts by visiting state $n - s_1$ (in s_1 steps) after which it will jump to state N (in one step). From state N , we repeatedly travel down (one step at a time) to some state $n - s_j$ (where $a_{n-s_j} = \lceil a_N - \tilde{T}_{k,j} \rceil_{\mathcal{A}}$, with $\tilde{T}_{k,j}$ having a shifted geometric distribution with parameter p), for $j \geq 2$ and jump back to state N as long as $\tilde{T}_{k,j} < B_{k,j}$, with $B_{k,j}$ distributed as B_k . As soon as $\tilde{T}_{k,j} \geq B_{k,j}$, we go from state $n - s_j$ to state m , with $m = \lceil a_N - \tilde{T}_{k,j} + B_{k,j} \rceil_{\mathcal{A}}$. We can split the original loss transitions in this manner, because conditioned on $\tilde{T}_{k,j} \geq B_{k,j}$, one easily checks that $\tilde{T}_{k,j} - B_{k,j}$ is distributed according to a shifted geometric distribution as required.

For ease of notation we introduce

$$F_n = P(B_k - T_k \leq 0 \ \& \ T_k \leq n) = \sum_{i=1}^n t_i \bar{b}_i. \quad (13)$$

Due to the upper Hessenberg form we have $\bar{p}_{n,m} = 0$ if $m < n - 1$. Furthermore, there is a transition from state n to state $n - 1$ in case the inter-arrival time is larger than $a_n - a_{n-1}$ (for $n = 1, \dots, N$), i.e., $\bar{p}_{n,n-1} = 1 - \bar{t}_{a_n - a_{n-1}}$. For $0 < n < N$, we find, analogous to the derivation of the transition probabilities in Section 3.2, $\bar{p}_{n,n} = F_{a_n - a_{n-1}}$, whereas for $0 < n < m < N$

$$\begin{aligned} \bar{p}_{n,m} &= \sum_{i=1}^{a_n - a_{n-1}} t_i (\bar{b}_{i+a_m - a_n} - \bar{b}_{i+a_{m-1} - a_n}) \\ &= q^{a_n - a_m} (F_{a_m - a_{n-1}} - F_{a_m - a_n}) - q^{a_n - a_{m-1}} (F_{a_{m-1} - a_{n-1}} - F_{a_{m-1} - a_n}), \end{aligned}$$

and for $0 \leq m \leq N - 1$ we find $\bar{p}_{0,m} = U_{a_m} - U_{a_{m-1}}$.

Only in the last column we need to add a contribution that corresponds to the event of a transition with loss. This leads to the probability $\bar{p}_{0,N} = 1 - U_{a_{N-1}}$,

for $0 < n < N$ we get

$$\begin{aligned}\bar{p}_{n,N} &= \sum_{i=1}^{a_n - a_{n-1}} t_i (1 - \bar{b}_{i+a_{N-1}-a_n}) \\ &= \bar{t}_{a_n - a_{n-1}} - q^{a_n - a_{N-1}} (F_{a_{N-1} - a_{n-1}} - F_{a_{N-1} - a_n}),\end{aligned}$$

and finally $\bar{p}_{N,N} = \bar{t}_{a_N - a_{N-1}}$. Remark that the transition probabilities $\bar{p}_{n,m}$ are somewhat easier to compute than the transition probabilities $p_{n,m}$.

The stationary probability vector of \bar{P} will be denoted as $\bar{\pi} = (\bar{\pi}_0, \dots, \bar{\pi}_N)$. To compute the loss probability we need the stationary probability vector π of P , where we will prove the following relation between π and $\bar{\pi}$:

$$\pi_j = \frac{\sum_{i=0}^N \bar{\pi}_i \hat{p}_{i,j}}{\sum_{k=0}^N \sum_{i=0}^N \bar{\pi}_i \hat{p}_{i,k}}, \quad (14)$$

where $\hat{p}_{i,j}$ is defined as

$$\begin{aligned}\hat{p}_{0,m} &= \bar{p}_{0,m} = U_{a_m} - U_{a_{m-1}}, & 0 \leq m < N \\ \hat{p}_{n,m} &= 0, & m < n \\ \hat{p}_{n,n} &= \bar{p}_{n,n} = F_{a_n - a_{n-1}}, & 0 < n < N \\ \hat{p}_{n,m} &= \bar{p}_{n,m} = q^{a_n - a_m} (F_{a_m - a_{n-1}} - F_{a_m - a_n}) \\ &\quad - q^{a_n - a_{m-1}} (F_{a_{m-1} - a_{n-1}} - F_{a_{m-1} - a_n}), & 0 < n < m < N \\ \hat{p}_{0,N} &= U_{a_N} - U_{a_{N-1}} \\ \hat{p}_{n,N} &= q^{a_n - a_N} (F_{a_N - a_{n-1}} - F_{a_N - a_n}) \\ &\quad - q^{a_n - a_{N-1}} (F_{a_{N-1} - a_{n-1}} - F_{a_{N-1} - a_n}), & 0 < n < N \\ \hat{p}_{N,N} &= F_{a_N - a_{N-1}}.\end{aligned} \quad (15)$$

Recall, the idea behind the Hessenberg reduction makes that for each sample path of the dense system P (of Section 3.2), there is a corresponding path for \bar{P} . The probabilities $\hat{p}_{i,j}$ are exactly those contributions to \bar{P} for which the visit to state j also occurs in the corresponding sample path of P .

Let $\check{p}_{i,j}$ be the remaining probability $\bar{p}_{i,j} - \hat{p}_{i,j}$ (hence, $\bar{P} = \hat{P} + \check{P}$), then equation (14) can be proven as follows. Due to the sample path correspondence we have $P = (I - \check{P})^{-1} \hat{P}$, while the steady state vector of \bar{P} satisfies

$$\begin{aligned}\bar{\pi} &= \bar{\pi}(\hat{P} + \check{P}) \\ \Leftrightarrow \bar{\pi}(I - \check{P}) &= \bar{\pi}\hat{P} \\ \Leftrightarrow \bar{\pi} &= \bar{\pi}\hat{P}(I - \check{P})^{-1} \\ \Leftrightarrow \bar{\pi}\hat{P} &= \bar{\pi}\hat{P}(I - \check{P})^{-1}\hat{P} \\ \Leftrightarrow \bar{\pi}\hat{P} &= \bar{\pi}\hat{P}P.\end{aligned}$$

Therefore we can conclude that the steady state vector π can be computed as described in equation (14).

5 Numerical Results

Since in practical cases the number of delay lines is small (typically 5 to 10), the gain of the reduction to the Hessenberg matrix, as discussed in Section 4, will be

limited when considering a single set of FDL lengths. However, the contribution of our new model lies in its applicability in optimization studies, where the gain of the computational complexity will be more explicit. In this section we will make a comparative study for a specific optimization algorithm, namely the Largest Reduction Addition algorithm. This study illustrates that the exploitation of the skip-free to the left structure of the transition matrix places us in a position to deal with larger systems, while drastically reducing the computation times.

5.1 Largest Reduction Addition (LRA) algorithm

In [10] we studied how the loss rate of the classic equidistant FDL buffer can be improved by considering alternate delay line structures, which means that the delay line lengths a_k are not necessarily a multiple of the constant D . Therefore we developed and compared three different heuristic algorithms. Due to performance and computational reasons, the Largest Reduction Addition (LRA) algorithm was superior to the other two algorithms. When applying the LRA algorithm, we assume that the size of the buffer N and the maximum packet length B_{\max} are known. Besides we define $\mathcal{A}_0 = \{a_0\}$ and the addition of the delay value a_{n+1} to the set \mathcal{A}_n is denoted by $\mathcal{A}_{n+1} = \{\mathcal{A}_n, a_{n+1}\}$. In each step of the LRA algorithm an FDL is added. The length of this FDL is chosen as the one whose addition causes the largest reduction in the loss rate. The interval in which we have to look for the optimal a_{n+1} , given the set \mathcal{A}_n , will be given by $I_{n+1} = [a_n + 1, a_n + B_{\max} - T_{\min}]$, where T_{\min} denotes the minimal inter-arrival time. By default the LRA algorithm sets the minimal value of the optimization domain to $a_n + 1$ (various numerical experiments, not reported here, have indicated that the optimum was always found beyond $a_n + 1$). To upper bound the interval one can reason as follows. Consider the FDL set \mathcal{A}_n and assume that the last accepted burst experienced a delay of a_n time units. The highest observable horizon value equals a_n plus B_{\max} , the largest possible burst size, minus the minimal inter-arrival time T_{\min} . Hence, the next burst will always be accepted if the buffer has an additional delay line a_{n+1} with length $a_{n+1} = a_n + B_{\max} - T_{\min}$. Therefore, selecting a_{n+1} beyond this value will only cause more losses as such an a_{n+1} value would result in a higher horizon value without being able to accept more bursts.

The LRA algorithm consists of the following N steps:

- Step 1: an FDL buffer consisting of one FDL is constructed. Its length is given by $a_1 = \operatorname{argmin}_{\omega \in I_1} \left(p_{\text{loss}}^{\{\mathcal{A}_0, \omega\}} \right)$, where $p_{\text{loss}}^{\mathcal{A}}$ denotes the loss probability for the specific set \mathcal{A} of FDLs.
- Step $n = 2$ to N : The set \mathcal{A}_{n-1} is updated to \mathcal{A}_n by adding one delay value. The length of the delay value is chosen as $a_n = \operatorname{argmin}_{\omega \in I_n} \left(p_{\text{loss}}^{\{\mathcal{A}_{n-1}, \omega\}} \right)$.

In [10] the computation of the loss probabilities in each step and for each of the ω -values was based on the horizon approach. Using the approach taken in [12], one significantly reduces the computation time. However, most of the computations have to be redone when altering the value of ω . In the next section, we will show

that for the Hessenberg reduction, we can retrieve the loss probability for all ω values by solving a single Markov chain (in $O(n^2)$ time).

5.2 A fast LRA implementation based on the Hessenberg matrix

The structure of the Hessenberg matrix provides a fast implementation of the LRA algorithm. We denote the transition matrix that characterizes the Markov chain corresponding with the set \mathcal{A}_n as $\bar{P}^{(n)}$ and the corresponding probabilities $\bar{p}_{i,j}$, respectively $\hat{p}_{i,j}$, as $\bar{p}_{i,j}^{(n)}$, respectively as $\hat{p}_{i,j}^{(n)}$. Assume the set \mathcal{A}_{n-1} is known and we have to determine the optimal length for a_n which has to be searched in the interval I_n . We first construct the transition matrix $\bar{P}^{(n,1)}$ corresponding with the delay line length $a_n = a_{n-1} + 1$. It is not hard to verify that

$$\bar{P}^{(n,1)} = \left[\begin{array}{c|cc} \bar{P}_l^{(n-1)} & \hat{P}_{0,n-1}^{(n-1)} & \bar{P}_{0,n}^{(n)} \\ & \hat{P}_{1,n-1}^{(n-1)} & \bar{P}_{1,n}^{(n)} \\ & \vdots & \vdots \\ & \hat{P}_{n-1,n-1}^{(n-1)} & \bar{P}_{n-1,n}^{(n)} \\ \hline 0 & \dots & 0 & \bar{P}_{n,n-1}^{(n)} & \bar{P}_{n,n}^{(n)} \end{array} \right],$$

where $\bar{P}_l^{(n-1)}$ represents the $n \times (n-1)$ matrix found in the upper left corner of $\bar{P}^{(n-1)}$. So we can construct the transition matrix $\bar{P}^{(n,1)}$ by recomputing at most $n+2$ elements. Furthermore, the probabilities $\hat{p}_{i,j}^{(n,1)}$ can be extracted immediately from the transition matrix $\bar{P}^{(n,1)}$ except for $0 \leq i \leq n = j$. Thus, recomputing $2(n+1) + 1$ elements suffices to determine $\bar{\pi}^{(n,1)}$ and its corresponding loss probability $p_{\text{loss}}^{(n,1)} = p_{\text{loss}}^{\{\mathcal{A}_{n-1}, a_{n-1}+1\}}$. For the other possible delay line lengths within the interval I_n , i.e., $a_n = a_{n-1} + x$, with $x > 1$, we can easily determine the steady state vector $\pi^{(n,x)}$ without the need to compute $\bar{P}^{(n,x)}$ or $\bar{\pi}^{(n,x)}$. This can be explained as follows. Since the transition matrix $\bar{P}^{(n,x)}$ differs only in two entries from $\bar{P}^{(n,1)}$, namely in $\bar{p}_{n,n-1}^{(n,x)}$ and $\bar{p}_{n,n}^{(n,x)}$, the steady state vector $\bar{\pi}^{(n,x)}$ can be expressed in terms of $\bar{\pi}^{(n,1)}$:

$$\bar{\pi}_i^{(n,x)} = \frac{\bar{\pi}_i^{(n,1)}}{\sum_{j=0}^n \bar{\pi}_j^{(n,x)}} \quad i = 0, \dots, n-1 \quad (16a)$$

$$\bar{\pi}_n^{(n,x)} = \frac{\theta^{(n,x)}}{\sum_{j=0}^n \bar{\pi}_j^{(n,x)}}, \quad (16b)$$

where

$$\theta^{(n,x)} = \frac{\sum_{k=0}^{n-1} \bar{\pi}_k^{(n,1)} \bar{p}_{k,n}^{(n,1)}}{(1 - \bar{t}_x)}. \quad (17)$$

This can be used to determine the steady state vector $\pi^{(n,x)}$ based on equation (14):

$$\pi_i^{(n,x)} = \frac{\sum_{k=0}^i \bar{\pi}_k^{(n,1)} \hat{p}_{k,i}^{(n,1)}}{\sum_{j=0}^n \pi_j^{(n,x)}} \quad i = 0, \dots, n-1 \quad (18a)$$

$$\pi_n^{(n,x)} = \frac{\sum_{k=0}^{n-1} \bar{\pi}_k^{(n,1)} \hat{p}_{k,n}^{(n,x)} + \theta^{(n,x)} \hat{p}_{n,n}^{(n,x)}}{\sum_{j=0}^n \pi_j^{(n,x)}}. \quad (18b)$$

Remark that we do not have to compute $\bar{\pi}^{(n,x)}$ to get $\pi^{(n,x)}$, but we can compute $\pi^{(n,x)}$ immediately from $\bar{\pi}^{(n,1)}$ and the probabilities $\hat{p}_{i,n}^{(n,x)}$ with $0 \leq i \leq n$. Therefore we have to recompute only $n+1$ elements for each value of x .

At the end of each step of the LRA algorithm, we determine x_{opt} such that $p_{\text{loss}}^{(n,x_{\text{opt}})} = \min_{x \in I_n} (p_{\text{loss}}^{(n,x)})$ and we denote the corresponding transition matrix $\bar{P}^{(n,x_{\text{opt}})}$ as $\bar{P}^{(n)}$ and the corresponding probabilities $\hat{p}_{i,j}^{(n,x_{\text{opt}})}$, respectively $\hat{p}_{i,j}^{(n,x_{\text{opt}})}$, as $\bar{p}_{i,j}^{(n)}$, respectively $\hat{p}_{i,j}^{(n)}$.

Notice, for the dense matrix approach of [12], we can only reuse some of the intermediary results of the calculation of $P^{(n-1)}$ when computing the transition matrices $P^{(n,x)}$.

5.3 Execution times of the fast LRA algorithm

As explained in Section 3.1, we focus on overload periods during which we assume Bernoulli arrivals with p large such that the load is close to or above one. Therefore, when regarding the loss probabilities computed in this section, we should keep in mind that these are the loss rates during the overload periods and not the long-run loss probabilities, which will be several orders of magnitude smaller. Clearly, the results and conclusions drawn in this section remain significant when the overload loss rates are scaled down to the long-run situation.

The study in [10] made use of packet traces collected by the NLANR (National Laboratory for Applied Network Research). More specifically, in this paper we will use an IP packet trace coming from the following link: AIX (a measurement point that sits at the interconnection point of NASA Ames and the MAE-West interconnection of Metropolitan Fiber Systems). The distribution of the packet sizes of the considered trace is depicted in Figure 1. Furthermore we use an FDL buffer with $N = 10$ FDLs. In order to keep the computation time in [10] limited, we had to define the length of a single slot equal to a number of bytes (more specifically, 50 bytes), as a result the packet sizes distribution had to be clustered (such that all packets had a multiple of 50 bytes as their packet length). With the approach taken in [12], clustering is no longer required as the packet length distribution has no impact on the size of the state space of the Markov chain. In other words, we can select the slot length equal to one byte. The main objective of this section exists in demonstrating the substantial

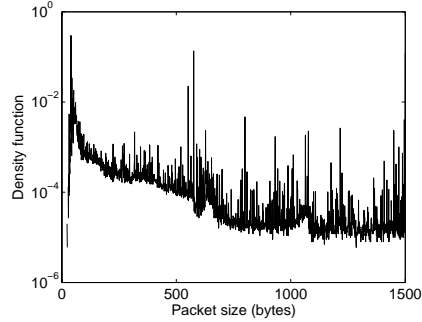


Fig. 1. Trace-based IP packet length distribution as captured on the AIX link

computational gain that is realized by the Hessenberg reduction over the dense matrix solution of [12] when running the LRA algorithm.

Figure 2(a) shows the execution times as a function of the number of FDLs, with a load $\rho = 0.9$. From this figure we can conclude that the Hessenberg solution offers a gain of several orders of magnitude over the dense matrix approach. Similar results are observed in Figure 2(b) that shows the execution times as a function of the load in case there are 10 FDLs. In order to quantify

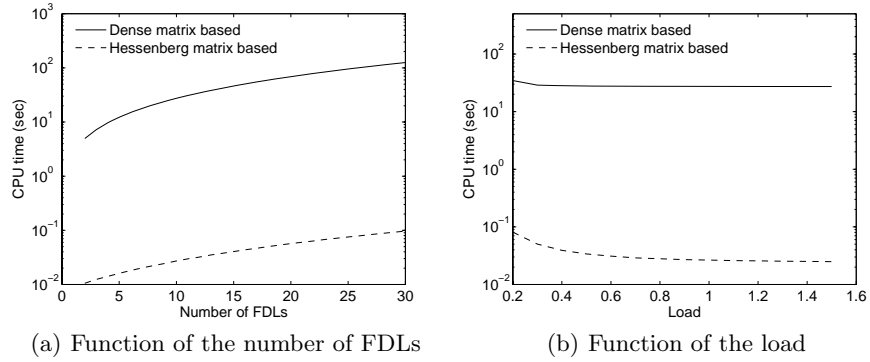


Fig. 2. Execution times for the different strategies that can be used to solve the LRA algorithm

the loss reduction typically realized by the LRA algorithm, Figure 3(a) shows the loss probability as a function of the load for both equidistant delay values, with $D = B_{\max} - 1$, and for the combination of delay values found by the LRA algorithm.

Figure 3(b) shows the loss probabilities for different FDL length settings as they are calculated during the N steps of the LRA algorithm with load $\rho = 0.9$.

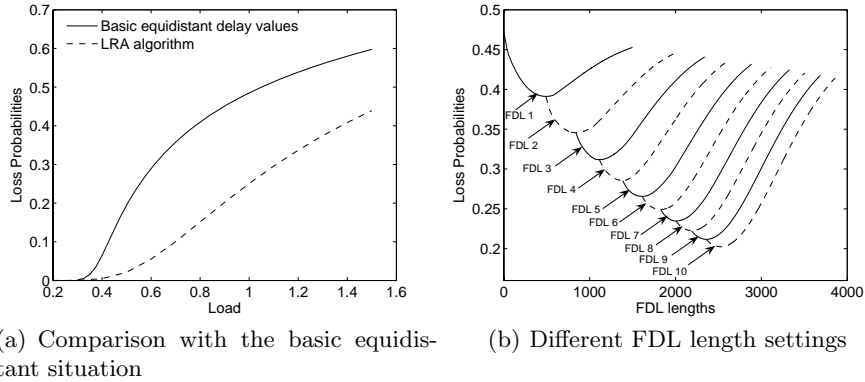


Fig. 3. Loss probabilities determined with the LRA algorithm

Table 1. FDL lengths a_k and differences d_k between subsequent FDL lengths determined with the LRA algorithm

FDL k	$\rho = 0.2$		$\rho = 0.9$	
	a_k	d_k	a_k	d_k
1	1499	1499	491	491
2	2998	1499	841	350
3	4497	1499	1114	273
4	5996	1499	1386	272
5	7495	1499	1612	226
6	8994	1499	1827	215
7	10493	1499	2005	178
8	11992	1499	2187	182
9	13491	1499	2360	173
10	14990	1499	2531	171

The curve of FDL 1 shows the loss rate as influenced by the length of the first line. A minimum is realized at $a_1 = 491$ bytes and serves as a starting point for the second step of the LRA algorithm. During the second step, the optimal length for the second FDL is found in $a_2 = 841$ (FDL 2 in Figure 3(b)). This procedure is repeatedly executed until N FDL lengths are found. Note, the LRA algorithm is a heuristic algorithm to find useful non-equidistant FDL lengths, but it does not necessarily realize a global minimum. The resulting FDL sets and differences between subsequent FDL lengths (i.e., $d_k = a_k - a_{k-1}$) are displayed in Table 1. The results for the scenario with a low load of $\rho = 0.2$ are also included in this table.

The experimental results of this section indicate that for low loads the performance of the LRA algorithm coincides with the equidistant FDL buffer, whereas for higher loads we can realize a substantial reduction in loss when using non-equidistant delay values. This observation was already made in [10] based on the

clustered trace, confirming our intuition that generally a clustered trace should suffice to get a good understanding of the system behavior.

6 Conclusion

This paper presented a novel approach to compute the invariant vector of the $N + 1$ state Markov chain presented in [12], based on a Hessenberg reduction, and resulting in a fine-tuned fibre delay line length optimization scheme. While previous work already allowed to compute the loss rate of a general FDL buffer [12], it was not adapted to the iterative scheme typically associated with optimization and as such inefficient to determine the optimal FDL length. Applying the Hessenberg reduction to the Largest Reduction Addition algorithm [10], we showed how the exploitation of the specific structure of the transition matrix (skip-free to the left) allows to compute the loss rate for an entire range of FDL lengths at the same time. This places us in a position to deal with larger systems, while drastically reducing the computation times. Finally, a comparative study showed how non-equidistant FDL sets are especially useful during overload periods, however, the results and conclusions remain of course significative when the overload loss rates are scaled down to the long-run situation.

References

1. R. C. Almeida, J. U. Pelegri, and H. Waldman. A generic-traffic optical buffer modeling for asynchronous optical switching networks. *Communications Letters, IEEE*, 9(2):175–177, 2005.
2. F. Callegati. Optical buffers for variable length packet switching. *IEEE Communications Letters*, 4:292–294, 2000.
3. Chlamtac et al. CORD: Contention resolution by delay lines. *IEEE Journal on Selected Areas in Communications*, 14:1014–1029, June 1996.
4. Z. Haas. The staggering switch: An electronically controlled optical packet switch. *Journal on Lightwave Technology*, 11:925–936, May/June 1993.
5. D. Hong, F. Poppe, J. Reynier, F. Bacelli, and G. Petit. The impact of burstification on TCP throughput in optical burst switching networks. In *Proc. of the 18th International Teletraffic Congress (ITC)*, Berlin (Germany), Sept. 2003.
6. B. Van Houdt, K. Laevens, J. Lambert, C. Blondia, and H. Bruneel. Channel utilization and loss rate in a single-wavelength fibre delay line (FDL) buffer. In *Proceedings of IEEE Globecom 2004, paper OC05-07*, Dallas USA, November 2004.
7. D. Hunter and I. Andonovic. Approaches to optical Internet packet switching. *IEEE Communications Magazine*, 38(9):116–120, Sept. 2000.
8. D. Hunter, W.D. Cornwell, T.H. Gilfedder, A. Franzen, and I. Andonovic. SLOB: A switch with large optical buffers for packet switching. *IEEE/OSA Journal on Lightwave Technology*, 16:1725–1736, Oct. 1998.
9. K. Laevens and H. Bruneel. Analysis of a single wavelength optical buffer. In *Proceedings of Infocom*, San Francisco, April 2003.
10. J. Lambert, B. Van Houdt, and C. Blondia. Single-wavelength optical buffers: non-equidistant structures and preventive drop mechanisms. In *Proceedings of NAEC 2005*, pages 545–555, Riva del Garda, 2005.

11. J. Qiao and M. Yoo. Optical burst switching: A new paradigm for an optical Internet. *Journal on High-Speed Networks*, 8:69–84, 1999.
12. W. Rogiest, D. Fiems, K. Laevens, and H. Bruneel. Tracing an optical buffer's performance: An effective approach. In *Lecture Notes in Computer Science, NET-COOP 2007 Special Issue*, pages 4465:185–194, 2007.
13. J. Turner. Terabit burst switching. *Journal on High-Speed Networks*, 8:3–16, 1999.
14. Y. Xiong, M. Vandenhoute, and H. Chankaya. Control architecture in optical burst-switched WDM-networks. *IEEE Journal on Selected Areas in Communications*, 18:1838–1851, 2000.