End-to-end delays in polling tree networks

P. Beekhuizen
Philips Research
Eindhoven, The Netherlands
and
EURANDOM
Eindhoven University of Technology
beekhuizen@eurandom.tue.nl

T.J.J. Denteneer
Philips Research
Eindhoven, The Netherlands
dee.denteneer@philips.com

J.A.C. Resing
Eindhoven University of Technology
Department of Mathematics and Computer Science
Eindhoven, The Netherlands
resing@win.tue.nl

ABSTRACT

We consider a tree network of polling stations operating in discrete-time. Packets arrive from external sources to the network according to batch Bernoulli arrival processes. We assume that all nodes have a service discipline that is *HoL-based*. The class of HoL-based service disciplines contains for instance the Bernoulli and limited service disciplines, and hence also the classical exhaustive and 1-limited. We obtain an exact expression for the overall mean end-to-end delay, and an approximation for the mean end-to-end delay of packets per source. The study is motivated by Networks on Chips where multiple processors share a single memory.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Queueing Theory

General Terms

Theory, Performance

Keywords

Polling, Concentrating tree networks, HoL-based service disciplines, Networks on Chips

1. INTRODUCTION

In polling systems multiple queues share a single server, which leads to all kinds of research topics in performance analysis, optimisation, etc. Polling models have many applications, for example in telecommunications, transportation, healthcare, etc., and have been the subject of numerous studies (for surveys, see [19, 20, 22]).

The main application that motivates this study is a Network on Chip (NoC). Networks on Chips are an emerging paradigm for the connection of on-chip modules like processors and memories. Such modules are traditionally connected via single buses, but because these buses cannot be used by multiple modules simultaneously, communication difficulties arise as the number of modules increases. Networks on Chips have been proposed as a solution (see [9]). In NoCs, routers are used to transmit packets to their destination, so that multiple links can be used at the same time and communication becomes more efficient.

In particular, we are motivated by a NoC where multiple masters (e.g., processors) share a single slave (e.g., memory). Packets travel from the processors to the memory over a number of routers. Each router has several queues sharing a single link connecting that router to the next. Because the link can be seen as a server attending multiple queues, the router can be seen as a polling station, and the network of routers thus as a network of polling stations.

Although polling systems have been studied extensively, few attempts have been made to analyse *networks* of polling servers; one of the rare examples is a heavy-traffic study [13]. Recently, the authors showed in [1] that a tree network of polling systems can be reduced to a single node, while preserving some information on the mean end-to-end delay. This reduction will be discussed in more detail in Section 2.1.

The mean end-to-end delay per source is an important measure for the performance of Networks on Chips. For instance, if this delay is large, it means that processors have to wait a long time before data can be written to or read from the memory, which in turn degrades the performance of the processors.

In this paper, we obtain an exact expression for the mean end-to-end delay averaged over all sources and an approximation of the mean end-to-end delay per source. The essential steps in this approximation are on the one hand the assumption that all streams passing through a certain queue at a node have the same mean waiting time in that node, and on the other hand application of the reduction result of [1].

In the approximation, we express the mean end-to-end delay per source in terms of the mean waiting time (per queue) in single-station polling systems. Depending on the service disciplines used, the mean waiting time in these single-station polling systems can either be determined exactly or has to be approximated.

The reduction result is valid for the class of HoL-based ser-

ValueTools 2008, October 21–23, 2008, Athens, GREECE

vice disciplines. This class contains for instance the Bernoulli scheduling and m_i-limited service disciplines. In this paper, we are especially interested in polling stations with the 1-limited service discipline, because that discipline will prove valuable for Networks on Chips.

This paper is organised as follows: The model is introduced in Section 2. In Section 3, we derive an exact expression for the mean end-to-end delay averaged over all sources, and obtain the approximation of the mean end-to-end delay per source. We perform a detailed simulation study on the accuracy of the assumption that all streams passing through a certain queue at a node have the same mean waiting time in that node in Section 4. The availability of single-station results is discussed in more detail in Section 5. For trees with a symmetry property, our approximation assumption becomes exact, as is discussed in Section 6. In Section 7 we combine the mean end-to-end delay approximation of Section 3 with a single-station approximation of Section 5 and apply these to a tree network model of Networks on Chips. We present our conclusions in Section 8.

2. MODEL

We consider a concentrating tree network operating in discrete time. An example is displayed in Figure 1. All packets and time slots have fixed size 1. Packets arrive from external sources at the end of time slots in batches according to batch Bernoulli arrival processes. By this we mean that the number of packets arriving is stochastically identical in each time slot, and independent of what happened in previous time slots. Furthermore, we assume that all arrival processes from the external sources are mutually independent.

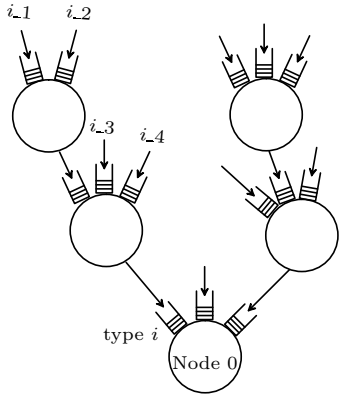


Figure 1: A polling tree network.

A packet arriving at a node at the end of time slot $[t-1, t)$, i.e., at time t, may be served in time slot $[t, t+1)$. In this case it reaches the next node at time $t+1$ where it may be served in time slot $[t+1, t+2)$, and so on.

All nodes in the network are polling nodes without switch-over times. Node 0 is a node with N queues and is the last node of the network (the sink). All packets in the network must eventually pass through it and leave the network after that.

The service disciplines of all nodes are work-conserving. Besides that, they remain unspecified for the moment; we will make an additional assumption on the service disciplines in Section 2.1.

We call a packet that passes through queue i of node 0 a 'type i' packet. There are N_i external sources from which type i packets arrive. We subdivide type i packets into 'type i_j' packets, $j = 1, \ldots, N_i$, such that the type denotes the source from which packets arrive (see Fig. 1). The set of type i packets is thus the union of the sets of type i_j packets.

The size of the batches of type i_j packets arriving each time slot is given by an arbitrary discrete non-negative random variable, denoted by X_{i_j}. We further define $X_i = \sum_{j=1}^{N_i} X_{i\_j}$, and $X = \sum_i X_i$. Because all packets have size 1, we denote the expected batch sizes by ρ_{i_j}, ρ_i, and ρ, respectively. We assume $\rho < 1$ so all nodes are stable.

Every node n in the tree network is itself the last node (the sink) in a smaller tree network consisting of all nodes above n and n itself. We call the latter network the node n subtree.

2.1 Reduction to a single node

In [1], it was shown that, under a relatively mild assumption on the service discipline of node 0, an arbitrary tree network can be reduced to a single-station polling system, called the *reduced* system (see Fig. 2). This reduction leaves the mean delay of type i packets invariant; the mean end-to-end delay of type i packets in the original system is equal to the mean waiting time in queue i of the reduced system. Here, the end-to-end delay of a packet is defined as the sum of its waiting times at the individual nodes.

The reduced system is a system with arrival processes that are given by superpositions of the original arrival processes, i.e., it is a system with arrivals $X_i = \sum_j X_{i\_j}$ to queue i, $i = 1, \ldots, N$. The service discipline of the reduced system is the same as that of node 0 in the original system.

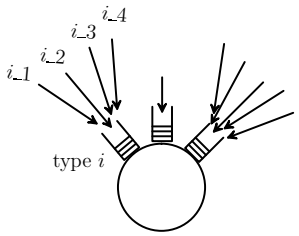


Figure 2: The reduced system.

If we denote the end-to-end delay of type i packets by Z_i, and the waiting time in queue i of the reduced system by W_i', we have:

$$\mathbb{E}[Z_i] = \mathbb{E}[W_i']. \tag{2.1}$$

Equation (2.1) is valid if node 0 uses a so-called Head-of-Line based (*HoL-based*) service discipline. For the precise definition of HoL-based we refer to [1], but it entails that the server decides which packet it is going to serve at time t only based on whether queues are empty or non-empty at times $t, t-1, \ldots, t-M$ for an arbitrary finite M. It may not, for instance, take queue lengths into account. Service disciplines such as longest/shortest queue first are thus not HoL-based.

For the present paper it suffices to say that the class of HoL-based service disciplines includes - but is not limited to - the following two classes of service discplines:

- Bernoulli scheduling, i.e., after serving a packet at queue i, the server serves queue i (if it is non-empty) again with probability p_i, and moves to one of the other non-empty queues with probability $1-p_i$;

- m_i-limited, i.e., the server serves queue i until it has served m_i packets, or the queue becomes empty, whichever happens first, before moving to one of the other non-empty queues.

If the server decides to select one of the other non-empty queues, it may do so according to some fixed order (e.g., a cyclic order) or according to Markovian routing. The exhaustive service policy (i.e., serve queue i until it becomes empty) is a special case of the Bernoulli scheduling, namely $p_i = 1$, and a limiting case of m_i-limited, namely $m_i \to \infty$. The 1-limited service discipline is a special case of both, namely $p_i = 0$ and $m_i = 1$.

The reduction result described here only yields an expression for the mean end-to-end delay of type i packets (called the mean type i end-to-end delay), while we are in particular interested in the mean end-to-end delay of type i_j packets (called the mean type i_j end-to-end delay). Even so, the reduction result will prove vital for the analysis of the mean type i_j end-to-end delay in Section 3. In order to apply the reduction to all possible subtrees, we assume that all nodes use HoL-based service disciplines.

REMARK 2.1. *The latter assumption can be slightly weakened. We apply the reduction result to all node n subtrees, except for nodes n where all queues store packets arriving directly from the exterior. If one or more queues of node n store packets coming from another node, the service discipline of node n has to be HoL-based. If all queues store packets arriving directly from the exterior, the service discipline can be an arbitrary work-conserving one.*

3. ANALYSIS OF THE TREE

In this section we describe how the reduction result can be applied to obtain expressions for the mean end-to-end delay. First, we obtain an exact expression for the mean end-to-end delay of packets of *any* type, called the mean overall end-to-end delay. Second, we approximate the mean type i_j end-to-end delay using the results for the mean overall end-to-end delay.

3.1 Overall end-to-end delay

Recall from Section 2.1, Eq. (2.1), that

$$\mathbb{E}[Z_i] = \mathbb{E}[W_i'],$$

where $\mathbb{E}[Z_i]$ is the mean type i end-to-end delay, and $\mathbb{E}[W_i']$ is the mean waiting time in queue i of the reduced system, which is a polling system with arrivals X_i to queue i, $i = 1, \ldots, N$. Because an arbitrary packet is of type i with probability ρ_i/ρ, it follows that $\mathbb{E}[Z]$, the mean overall end-to-end delay, is given by

$$\mathbb{E}[Z] = \sum_{i=1}^{N} \frac{\rho_i}{\rho} \mathbb{E}[Z_i] = \sum_{i=1}^{N} \frac{\rho_i}{\rho} \mathbb{E}[W_i'] \tag{3.1}$$

The right hand side of (3.1) can be recognised as part of the conservation law for polling systems [2, 4, 7]. This law states that for any work-conserving service discipline,

$$\sum_{i=1}^{N} \frac{\rho_i}{\rho} \mathbb{E}[W_i'] = C, \tag{3.2}$$

where C is a constant. For unit packet sizes, C is given by [2, Eq. (14), divided by ρ]:

$$\begin{aligned} C &= \frac{1}{2\rho(1-\rho)} \left[\sum_i \mathbb{E}[X_i(X_i-1)] + \sum_i \sum_{j \neq i} \rho_i \rho_j \right] \\ &= -\frac{1}{2} + \frac{1}{2\rho(1-\rho)} \sum_i \mathrm{Var}(X_i). \end{aligned} \tag{3.3}$$

By combining this with (3.1) and (3.2), and applying the definition of X_i, we obtain

$$\begin{aligned} \mathbb{E}[Z] &= -\frac{1}{2} + \frac{1}{2\rho(1-\rho)} \sum_i \mathrm{Var}(X_i) \\ &= -\frac{1}{2} + \frac{1}{2\rho(1-\rho)} \sum_i \sum_j \mathrm{Var}(X_{i\_j}) \end{aligned} \tag{3.4}$$

as the mean overall end-to-end delay.

REMARK 3.1. *Equation (3.4) gives the mean overall end-to-end delay, regardless of the precise HoL-based service discipline. The work of Morrison [12] and Shalmon [16] entails that Equation (3.4) holds without the assumption of HoL-based service disciplines; any work-conserving service discipline suffices. The assumption of HoL-based service disciplines will, however, become crucial in the next subsection. Shalmon [16] also gives an expression for the mean overall end-to-end delay in a concentrating tree network with Poisson arrivals, which is valid in discrete as well as continuous time (Eq. (3.4) with $\mathrm{Var}(X_{i\_j}) = \rho_{i\_j}$).*

3.2 End-to-end delay per type

In this subsection, we derive an approximation of the mean type i_j end-to-end delay. Our main result is that we express the type i_j end-to-end delay in the *network* in the mean waiting time per queue of *single-station* polling systems. Although the latter is not always known exactly, we assume that it can somehow be determined, either through exact analysis or approximation. We will come back to this issue in Section 5.

The first observation is that the type i_j end-to-end delay consists of the sum of the waiting times of type i_j packets at all nodes along their path from the source to node 0. In other words, if we approximate the mean waiting time of type i_j packets at an arbitrary node, an approximation of the mean type i_j end-to-end delay automatically follows by summing the mean waiting time approximations at the individual nodes.

A second observation is the following: Consider Figure 3 and suppose for a moment that we want to determine the mean waiting time of type i_j packets in node i. Everything that happens outside the node i subtree (marked by the dashes) has no influence on the mean waiting time in node i, so it suffices to consider only the node i subtree. Node i, however, is itself the sink of the node i subtree. In order to approximate the mean waiting time of type i_j packets in an arbitrary node, it hence suffices to determine the mean waiting time in the *last* node of an arbitrary network.

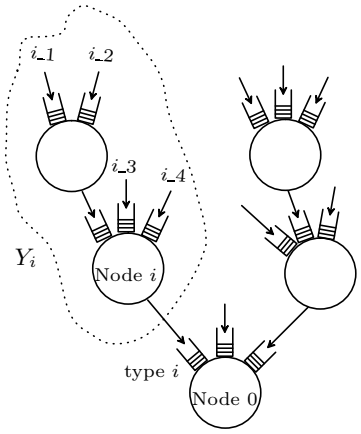


Figure 3: The example network.

In the sequel, we approximate the mean waiting time of type i_j packets in node 0, which leads to an approximation of the mean type i_j end-to-end delay as described by the two observations above. We denote the mean waiting time of type i_j packets in node 0 by $\mathbb{E}[W_{i\_j}^{(0)}]$.

It is not immediately clear, however, how $\mathbb{E}[W_{i\_j}^{(0)}]$ can be determined: First, it is unclear which of the type i packets in node 0 are actually type i_j packets. The type i_j packets are intermingled with packets of type i_j_1, i_j_2, etc. Packets are stored in node 0 in an intricate unknown order that is determined by the service disciplines of the nodes upstream. Second, $\mathbb{E}[W_{i\_j}^{(0)}]$ represents the mean waiting time in a polling model where the arrivals are given by the output of the node upstream.

The first difficulty is circumvented by the following approximation:

Approximation 3.2. *At every node, the mean waiting time of type i_j packets in that node is equated to the mean waiting of* all *packets passing through the* same queue *in that node.*

The accuracy of this approximation will be studied numerically in Section 4.

Applying Approximation 3.2 to node 0 entails that we approximate the mean waiting time at node 0 of type i_j packets by the mean waiting time of type i packets, i.e.,

$$\mathbb{E}[W_{i\_j}^{(0)}] \approx \mathbb{E}[W_i^{(0)}]. \tag{3.5}$$

The quantity $\mathbb{E}[W_i^{(0)}]$, however, still represents the mean waiting time in a polling model where arrivals are given by the output of the node upstream.

We can now circumvent the second difficulty with the reduction result of Section 2.1: The mean waiting time of type i packets at node 0, $\mathbb{E}[W_i^{(0)}]$, is equal to the mean type i end-to-end delay in the entire tree, $\mathbb{E}[Z_i]$, minus the mean type i end-to-end delay in the node i subtree, denoted by $\mathbb{E}[Y_i]$. Using the reduction result (Equation (2.1)) we thus obtain

$$\mathbb{E}[W_i^{(0)}] = \mathbb{E}[W_i'] - \mathbb{E}[Y_i].$$

Because all packets in the node i subtree are type i packets, $\mathbb{E}[Y_i]$ is the mean *overall* end-to-end delay in the node i subtree. It follows from the analysis of Section 3.1 (i.e., Equation (3.4) applied to the node i subtree) that

$$\mathbb{E}[Y_i] = -\frac{1}{2} + \frac{1}{2\rho_i(1-\rho_i)} \sum_j \mathrm{Var}(X_{i\_j}). \tag{3.6}$$

In summary,

$$\mathbb{E}[W_{i\_j}^{(0)}] \approx \mathbb{E}[W_i^{(0)}] = \mathbb{E}[W_i'] - \mathbb{E}[Y_i] \tag{3.7}$$

where $\mathbb{E}[Y_i]$ is given by (3.6), and $\mathbb{E}[W_i']$ is the mean waiting time in queue i of the reduced system. The two key steps in the derivation of (3.7) are Approximation 3.2 and application of the reduction result.

Remark 3.3. *If type i_j packets arrive to node 0 directly, there is of course no suitable subtree. In this case, we can replace $\mathbb{E}[Y_i]$ by 0, so that $\mathbb{E}[W_{i\_j}^{(0)}] \approx \mathbb{E}[W_i^{(0)}] = \mathbb{E}[W_i']$.*

4. ACCURACY OF APPROXIMATION 3.2

In this section, we analyse the accuracy of Approximation 3.2 by means of a simulation study over a large parameter space.

We consider the smallest non-trivial polling tree network, which consists of two nodes, node 0 and 1, both with two queues (see Fig. 4). Queue 1 of node 0 stores packets arriving from node 1 while queue 2 of node 0 stores packets arriving from the exterior directly. There are three different types of packets, namely type 1_1, type 1_2, and type 2_1. All arrivals occur according to ordinary (non-batch) Bernoulli arrival processes, i.e., each time slot an arrival of type i_j takes place with probability ρ_{i_j}. We introduce a unit sum weight vector $\nu = (\nu_{1\_1}, \nu_{1\_2}, \nu_{2\_1})$ such that $\rho_{i\_j} = \nu_{i\_j}\rho$ for a single load parameter ρ. We assume each node uses the 1-limited service discipline.

Without loss of generality, we assume $\nu_{1\_1} \leq \nu_{1\_2}$. We cover all possible cases of ν_{i_j} with a stepsize of 0.05 between consecutive values of ν_{i_j}. This leads to a total of 90 possible cases (see Table 1). For each case, we run simulations for ρ from 0.01 to 0.99.

We analyse the error made in the approximation of the mean waiting time at node 0 (Eq. (3.5)), i.e., we analyse the value of

$$\varepsilon_j = \frac{\mathbb{E}[W_1^{(0)}]}{\mathbb{E}[W_{1\_j}^{(0)}]} - 1, \qquad j = 1, 2,$$

where both $\mathbb{E}[W_1^{(0)}]$ and $\mathbb{E}[W_{1\_j}^{(0)}]$ are determined by simulation.

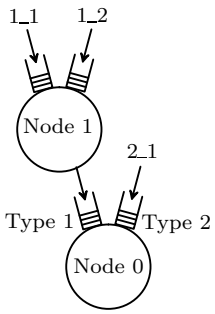
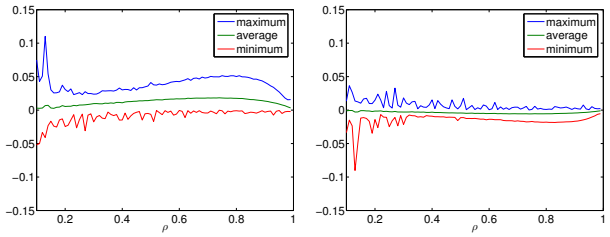


Figure 4: The network of Section 4.

Table 1: The 90 cases considered.

Case	ν_{1_1}	ν_{1_2}	ν_{2_1}	Case	ν_{1_1}	ν_{1_2}	ν_{2_1}
1	0.05	0.05	0.90	35	0.15	0.15	0.70
2	0.05	0.10	0.85	36	0.15	0.20	0.65
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	0.05	0.90	0.05	48	0.15	0.80	0.05
19	0.10	0.10	0.80	49	0.20	0.20	0.60
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
34	0.10	0.85	0.05	89	0.45	0.45	0.10
				90	0.45	0.50	0.05

Figure 5 displays the average and extreme values of ε_j over all cases. It clearly shows that the average error is within a few percent for all loads above 0.1. For loads close to 0, ε_j is the ratio of two numbers close to zero, which leads to some irrelevant variability in the graph. The results for $\rho < 0.1$ have therefore been omitted from the graph.



$j = 1$ $j = 2$

Figure 5: Average and extreme values of ε_j over all cases.

Apart from average and extreme values of ε_j, it is interesting to see which cases typically induce large errors. Table 2 shows the five cases that most frequently have large errors; clearly, cases with large errors are typically quite asymmetric. Additional simulations have furthermore shown that the error is typically largest if such an asymmetric case is combined with a load of around 0.7, 0.8.

There are, however, even more asymmetric cases, which are not in Table 2. Apparently, the error is again smaller for very asymmetric cases. To study this effect in more detail, we perform the following experiment: We fix $\rho = 0.8$ and $\nu_{2\_1} = 0.1$ (these settings generally lead to larger errors, so that the effect of asymmetry is clearly visible). We vary ν_{1_1} and ν_{1_2} subject to the constraints that $\nu_{1\_1} + \nu_{1\_2} = 0.9$ and $\nu_{1\_1} \leq \nu_{1\_2}$.

Figure 6 shows the values of ε_j, $j = 1, 2$, in this experiment. On the horizontal axis we have $\nu_{1\_2} - \nu_{1\_1}$, which is a measure for how asymmetric node 1 is: The left side corresponds to a symmetric system ($\nu_{1\_1} = \nu_{1\_2}$), and the right side corresponds to a completely asymmetric system ($\nu_{1\_2} - \nu_{1\_1} = 0.9$, i.e., $\nu_{1\_1} = 0$ and $\nu_{1\_2} = 0.9$).

Table 2: Cases that frequently have larger errors.

ν_{1_1}	ν_{1_2}	ν_{2_1}
0.15	0.80	0.05
0.15	0.75	0.10
0.20	0.70	0.10
0.20	0.75	0.05
0.25	0.70	0.05

Type 1_1

ν_{1_1}	ν_{1_2}	ν_{2_1}
0.25	0.65	0.10
0.25	0.70	0.05
0.30	0.60	0.10
0.30	0.65	0.05
0.35	0.60	0.05

Type 1_2

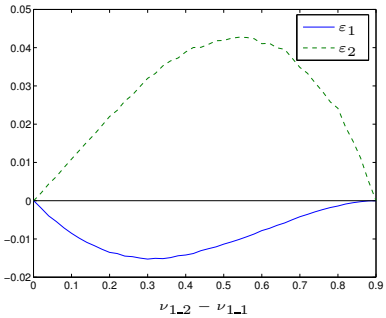


Figure 6: The influence of asymmetry.

Clearly, the absolute values of the errors increase if node 1 becomes more asymmetric, but only up to a certain point. After this point, the absolute values of the errors decrease again.

5. SINGLE STATION RESULTS

In Section 3, we expressed the mean type i_j end-to-end delay in terms of the mean waiting time per queue in the reduced system. In this section, we discuss the availability of single-station results for the reduced system. Recall that the reduced system has N queues where each time slot a batch of size X_i arrives to queue i. The service discipline of the reduced system is the same as that of node 0.

There is a remarkable distinction between service disciplines that so far have defied exact analysis of the mean waiting time in queue i, $\mathbb{E}[W_i']$ (except for special cases like symmetric and 2-queue stations), such as 1-limited, and service disciplines for which various methods exist to obtain $\mathbb{E}[W_i']$ exactly, such as exhaustive and gated service. With gated service, each time the server visits a queue an imaginary gate is placed behind the last packet in the queue. When all packets in front of that gate have been served, the server starts serving the next queue, again with an imaginary gate behind the last packet, and so on.

In [14], it is shown that service disciplines satisfying a well-known 'branching property' can be exactly analysed. This branching property states the following:

PROPERTY 5.1. *If the server arrives to queue i and finds k_i packets there, then during the course of the server's visit, all of these k_i packets are effectively replaced in an i.i.d. manner by an N-dimensional random population.*

For instance, with exhaustive service, all type i packets will have been removed (i.e., replaced by 0 packets) once the server moves to the next queue, whereas for $j \neq i$, as many type j packets will be added to queue j as there are arriving during one type i busy period (i.e., every type i packet is replaced by the packets arriving to the other queues during a type i busy period). Likewise, with gated service, for all j, including $j = i$, every type i packet will be replaced by the packets arriving during a type i service. In contrast, with the 1-limited service discipline *one* type i packet is replaced by packets arriving during the type i service, but all other type i packets are left unchanged (i.e., replaced by one type i packet). Hence, the 1-limited service discipline does not satisfy the branching property.

For service disciplines satisfying the branching property, it is shown in [14] that the number of packets in different

queues, embedded at time points where the server visits queue 1, constitutes a multi-type branching process (MTBP) with immigration. Furthermore, it is mentioned that the class of MTBPs is one of the exceptional classes of multi-dimensional Markov chains for which the equilibrium distribution can be determined.

This at least partially explains why methods exist to obtain mean queue lengths (and thus mean waiting times) for exhaustive and gated service disciplines. Nevertheless, even for these service disciplines, $\mathbb{E}[W_i']$ is, apart from special cases such as symmetric systems, not given explicitly but in terms of a matrix inverse, infinite product, or a solution to a set of equations.

We are in particular interested in HoL-based service disciplines for which $\mathbb{E}[W_i']$ can be obtained exactly. Gated service, however, is not HoL-based. With the gated service discipline an imaginary gate is placed behind packets in the queue, so the decision to serve a particular queue at time t may depend on the queue lengths before t, which is not allowed for HoL-based service disciplines.

Exhaustive service, on the other hand, is HoL-based. The mean queue lengths in a polling station with exhaustive service can be found in [15] and [18], where it is given in terms of a solution to a system of equations. Although the expressions given there are still implicit, $\mathbb{E}[W_i']$ can be determined numerically from them.

In the remainder of this section, we give explicit results for symmetric stations, and we give an approximate result for the 1-limited service discipline. The latter will be particularly important in Section 7.

5.1 Symmetric stations

In this subsection, we obtain exact results for symmetric stations. Throughout this subsection, we assume all arrival processes are stochastically identical: $X_j \stackrel{d}{=} X_1$ for all j.

We first introduce the concept of *symmetric service disciplines*: We define a service discipline to be symmetric if it satisfies the following three properties: First, if the server is at a queue it serves a number of packets there according to a fixed rule such as 1-limited, exhaustive, or Bernoulli service. This rule is the same for all queues. Second, Markovian routing is used, which means that after service of queue j, the server moves to queue $k \neq j$ with probability p_{jk}. Third, the routing matrix $P = (p_{jk})$ is circulant, i.e.,

$$P = \begin{pmatrix} 0 & p_2 & p_3 & \cdots & p_N \\ p_N & 0 & p_2 & \cdots & p_{N-1} \\ p_{N-1} & p_N & 0 & \cdots & p_{N-2} \\ \vdots & \vdots & & \ddots & \vdots \\ p_2 & p_3 & p_4 & \cdots & 0 \end{pmatrix}, \qquad (5.1)$$

or can be written in circulant form after a permutation of the queues. Note that cyclic routing has a circulant P-matrix with $p_2 = 1$.

Suppose that the reduced system uses a service discipline with a circulant P-matrix. Then all rows of P are identical, apart from being shifted. Because, in addition, $X_j \stackrel{d}{=} X_1$ for all j, there is no difference between the various queues, i.e., the mean waiting times per queue are invariant under permutation of the queues. Similarly, if the P-matrix is not circulant but can be written in circulant form, it can also be argued that the mean waiting times per queue are invariant under permutation of the queues.

Because the mean waiting times per queue are invariant under permutation, we have $\mathbb{E}[W_i'] = \mathbb{E}[W_1']$ for all i. It thus follows from the conservation law (3.2) that $\mathbb{E}[W_i']$ is exactly given by

$$\mathbb{E}[W_i'] = C = -\frac{1}{2} + \frac{1}{2\rho(1-\rho)} N \mathrm{Var}(X_1),$$

for all i, regardless of the precise service discipline, as long as it is symmetric.

5.2 1-limited

We give the approximation proposed by Boxma and Meister [6] for 1-limited polling systems. Although their analysis is aimed at continuous-time systems, it can easily be established that the key steps in their derivation are valid for discrete-time systems as well.

The Boxma-Meister approximation states that

$$\mathbb{E}[W_i'] \approx C \frac{1 - \rho + \rho_i}{1 - \rho + \frac{1}{\rho} \sum_j \rho_j^2},$$

where C is again the constant of the conservation law (3.3). The Boxma-Meister approximation has the two following properties: (i) It is exact for symmetric systems, and (ii) its numerical accuracy degrades for heavily loaded and very asymmetric systems.

In particular, if the 1-limited service discipline is used, all queues receive a positive fraction of the capacity of the server, even if the load is larger than 1. As a result, some queues remain stable even though others become unstable (see e.g., [8, 11]). The Boxma-Meister approximation does not deal with this well; if ρ tends to 1, C tends to infinity so the approximations of all queues become unbounded.

As a refinement to their approximation, Boxma and Meister suggest in [5] that for heavily loaded, very asymmetric systems, a group of heavily loaded queues can be replaced by a suitable switch-over time. Although this generally increases the accuracy of the approximation for such cases, we will see in Section 7.2 that the approximation presented here is accurate for loads up to 0.7, even in quite asymmetric systems.

Many other approximations have been suggested (for instance, Blanc [3], Groenendijk and Levy [10], Srinivasan [17], and Van Vuuren and Winands [21]), that might perform better in some cases. However, they generally lack an accessible closed-form expression like that of the Boxma-Meister approximation. Furthermore, transferring their derivations to discrete-time models often involves subtleties. In this paper, we therefore restrict our attention to the Boxma-Meister approximation.

6. SYMMETRIC TREES

Approximation 3.2 states that all packets passing through the same queue at a node are assumed to have the same mean waiting time at that node. In this section, we introduce a class of trees for which Approximation 3.2 is in fact not an approximation but an exact statement.

We say that a polling tree is symmetric if it satisfies all of the following five properties:

1. All external arrival processes are stochastically identical.

2. All external arrivals occur at the same level. Here, the level of a node is defined as the distance to the sink.

3. All nodes within a particular level have the same number of queues.

4. All nodes within a particular level use the same service discipline.

5. All nodes use a symmetric service discipline.

Now consider an arbitrary polling tree network, let node i be the node directly above queue i of node 0, and suppose that the node i subtree is symmetric. An example of such a tree is displayed in Figure 7.

Given that the entire node i subtree is symmetric, it follows that there is no distinction between the type i_j packets, $j = 1, \ldots, N_i$. In particular, the mean waiting time of type i_j packets at node 0 is invariant under permutation of j:

$$\mathbb{E}[W_{i\_j}^{(0)}] = \mathbb{E}[W_i^{(0)}].$$

Approximation 3.2 is thus an exact statement rather than an approximation for node 0. Because all subtrees within the node i subtree are again symmetric, Approximation 3.2 is exact for all nodes in the node i subtree.

Moreover, the mean end-to-end delay of type i_j packets is invariant under permutation of j. We thus obtain:

$$\mathbb{E}[Z_{i\_j}] = \mathbb{E}[Z_i] = \mathbb{E}[W_i'],$$

where Z_{i_j} is the end-to-end delay of type i_j packets, and W_i' is the waiting time in queue i of the reduced system. If, furthermore, exact results are available for the reduced system (for instance if it is a symmetric node or if it uses exhaustive service), the mean type i_j end-to-end delay can be obtained exactly.

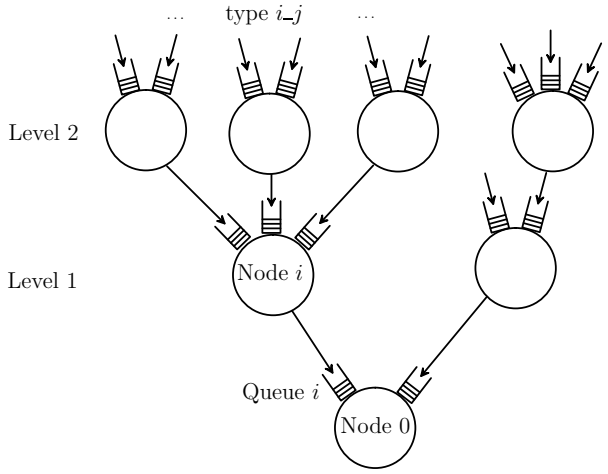


Figure 7: A tree with a symmetric node i subtree

Note that there is no condition on nodes outside the node i subtree; all conditions apply to the node i subtree, and all other nodes (including node 0) are arbitrary.

7. NETWORKS ON CHIPS

In this section, we study a network model based on a Network on Chip. In Section 7.1, we describe the network in more detail and combine the approximations of Section 3 and Section 5 to approximate the mean type i_j end-to-end delay. In Section 7.2 we analyse the accuracy of the combination of these approximations numerically.

7.1 Description

We study a model of a Network on Chip with multiple routers (nodes), as depicted in Figure 8. All traffic has the same destination, motivated by Networks on Chips with a single memory.

The routers in this network are organised in a mesh topology; all routers have four queues and are placed on a lattice (2×2 in this case) with connections in four directions (up, down, right, left) if possible. The routing mechanism of this network is XY-routing, which means that packets first traverse the X-direction, as far as they have to go, and then move in the Y-direction to their destination. This entails there is in fact a link between node 3 and node 1, but it is never used because all traffic is headed to node 0. It is thus the particular routing strategy that ensures the mesh topology is a tree network corresponding to the setting of this paper.

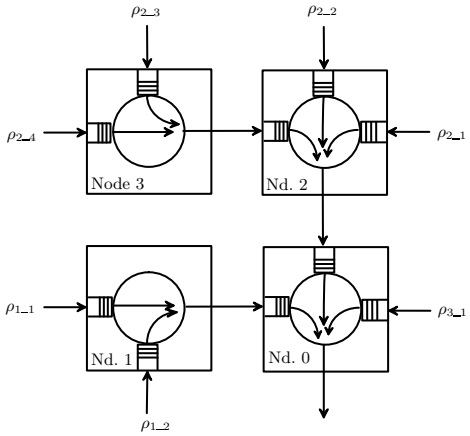


Figure 8: A NoC with 4 routers and 7 input streams

Routers in NoCs often employ wormhole routing, which has two implications: First, once the first packet of a batch starts transmission at a certain node, the entire batch has to complete transmission before another batch may start transmission. The second implication of wormhole routing is that if a batch consisting of multiple packets is being transmitted, one packet of the batch is transmitted to the next router each time slot. These packets might start transmission at the next router immediately (for instance if that router is empty). Multiple packets of a single batch might thus be spread out over several nodes. As a result, a batch of size K that never has to wait completes transmission over L nodes in $L+K-1$ time slots, rather than LK in networks without wormhole routing.

Inside routers batches are served according to round-robin scheduling, which means a batch is served from queue 1, then queue 2, etc. We assume batches have fixed size K, so that wormhole routing is mimicked by the cyclic K-limited service discipline.

We furthermore assume a batch of size K arrives each time slot with probability λ_{i_j}, so

$$X_{i\_j} = \begin{cases} 0 & \text{w.p. } 1 - \lambda_{i\_j}, \\ K & \text{w.p. } \lambda_{i\_j}, \end{cases}$$

and

$$\mathrm{Var}(X_{i\_j}) = K^2 \lambda_{i\_j}(1 - \lambda_{i\_j}).$$

Because $\rho_{i\_j} = \mathbb{E}[X_{i\_j}]$, it follows that $\lambda_{i\_j} = \rho_{i\_j}/K$.

We define the waiting time (and end-to-end delay) of a batch to be the waiting time (end-to-end delay) of the first packet in that batch (the header). In the remainder, we use an additional subscript h to indicate headers.

We now approximate the mean waiting time of a type i_j header in node 0, denoted by $\mathbb{E}[W^{(0)}_{i\_j,h}]$. As in Section 3, we do so by reducing the network to a single node, called the reduced system.

The reduced system, like node 0, uses K-limited service. Because the batch sizes are fixed and equal to K, the mean waiting time of a header in the reduced system is equal to the mean waiting time of a packet in a 1-limited system with deterministic service times equal to K and ordinary (non-batch) Bernoulli arrival processes with parameter λ_{i_j}.

By applying the Boxma-Meister approximation to the latter system, we obtain

$$\mathbb{E}[W'_{i,h}] \approx \frac{1-\rho+\rho_i}{1-\rho+\frac{1}{\rho}\sum_j \rho_j^2} C_K, \tag{7.1}$$

as an approximation of the mean waiting time of headers in queue i of the reduced system. Here, C_K is the constant of the conservation law in a system with non-batch Bernoulli arrivals with parameter λ_{i_j} and fixed service times equal to K. From [2, Eq. (14), divided by ρ], we get after some rewriting:

$$C_K = \frac{\rho}{2(1-\rho)} \left(K - \sum_i \sum_j \left(\frac{\rho_{i\_j}}{\rho} \right)^2 \right). \tag{7.2}$$

Suppose now that $i = 3$ (the case $i = 1, 2$ is slightly different and will be dealt with later). Type 3_1 packets arrive to node 0 directly from the exterior, so the mean waiting time of a header is equal to that of an arbitrary packet minus $(K-1)/2$. The mean waiting time of a header in node 0 (and hence its mean end-to-end delay) is thus given by

$$\mathbb{E}[W^{(0)}_{3\_1,h}] = \mathbb{E}[W^{(0)}_{3\_1}] - \frac{K-1}{2} = \mathbb{E}[W'_3] - \frac{K-1}{2} = \mathbb{E}[W'_{3,h}],$$

where $\mathbb{E}[W'_{3,h}]$ is given by (7.1).

Suppose now that $i = 1, 2$. Due to the wormhole routing, the header always arrives at node 0 one time slot earlier than the second packet, and it always leaves one time slot earlier. The mean waiting time of a header is thus equal to the mean waiting time of an arbitrary packet, $\mathbb{E}[W^{(0)}_{i\_j,h}] = \mathbb{E}[W^{(0)}_{i\_j}]$. We obtain (cf. Eq. (3.7))

$$\mathbb{E}[W^{(0)}_{i\_j,h}] = \mathbb{E}[W^{(0)}_{i\_j}] \approx \mathbb{E}[W^{(0)}_i] = \mathbb{E}[W'_i] - \mathbb{E}[Y_i],$$

where $\mathbb{E}[Y_i]$ is given by (3.6). The quantity $\mathbb{E}[W'_i]$ is the mean waiting time of an *arbitrary* packet in the reduced system, which is equal to the mean waiting time of a header plus $(K-1)/2$, so

$$\mathbb{E}[W'_i] = \mathbb{E}[W'_{i,h}] + \frac{K-1}{2}$$

with $\mathbb{E}[W'_{i,h}]$ as in (7.1).

As in Section 3, the mean waiting times in the other nodes can be obtained similarly, resulting in an approximation of the mean end-to-end delay of type i_j batches.

7.2 Numerical results

In this subsection we study the performance of the mean end-to-end delay approximation for the following two cases: Balanced load division and homogeneous load division. We again assume there is a unit sum weight vector ν describing the division of the total load ρ over the various input streams, i.e., $\rho_{i\_j} = \nu_{i\_j}\rho$.

Case I: Balanced load division

By balanced load division we mean that the loads are divided in such a way that at each node all queues receive the same load. That is, we assume $\nu_{3\_1} = 1/3$, $\nu_{1\_1} = \nu_{1\_2} = 1/6$, $\nu_{2\_1} = \nu_{2\_2} = 1/9$, and $\nu_{2\_3} = \nu_{2\_4} = 1/18$. Because the corresponding arrival rates are equal, the mean type 1_1 end-to-end delay is equal to the mean type 1_2 end-to-end delay. Figure 9 shows mean type 1_i end-to-end delay, $i = 1, 2$, and Figure 10 the mean type 3_1 end-to-end delay. The approximations of these types are the most and least accurate, respectively, of all approximations.

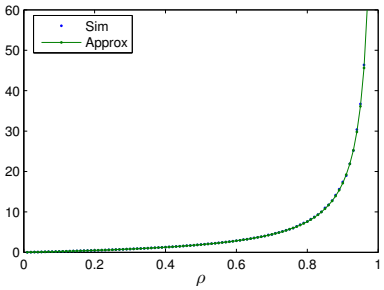


Figure 9: The mean type 1_1 and 1_2 end-to-end delay.

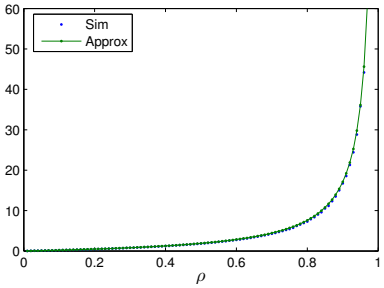


Figure 10: The mean type 3_1 end-to-end delay.

It is clear that the approximation of the mean end-to-end delay is very accurate in this case. This is not very surprising, as all nodes are almost symmetric. For instance, if we apply the reduction result, we obtain a polling system with three queues, each with load $\rho/3$. One of these queues has an arrival process that is the superposition of four arrival processes (namely $\sum_j X_{2\_j}$), one arrival process is a superposition of two ($\sum_j X_{1\_j}$), and one is not a superposition (or a superposition of one). In other words, the loads to all queues are identical, but the arrival processes are superpositions of different Bernoulli arrival processes.

Other than this difference, the system is symmetric, in which case the Boxma-Meister approximation is exact. It is indeed unlikely that such a small asymmetry leads to large errors. Furthermore, we already saw in Section 4 that Approximation 3.2 is very accurate if the individual nodes are nearly symmetric.

Case II: Homogeneous load division

With the homogeneous load division, all input streams receive a fraction 1/7 of the total load, i.e., $\nu_{i\_j} = 1/7$. Again, we show the most accurate approximation (Fig. 11, type 2_1 and 2_2), and the least accurate approximation (Fig. 12, type 3_1).

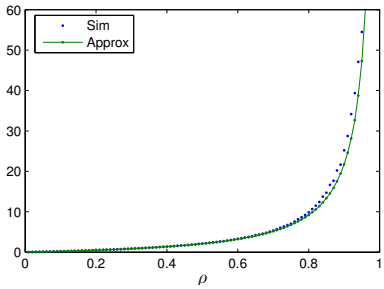


Figure 11: The mean type 2_1 and 2_2 end-to-end delay.

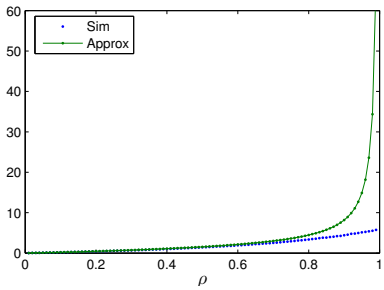


Figure 12: The mean type 3_1 end-to-end delay.

We see that up to a load of roughly 0.7, the approximations are very accurate. Beyond this load, the approximation is only accurate for the input stream with the highest load. This can be explained by the fact that node 0 is rather asymmetric. After all, one queue receives a fraction of 4/7 of the total load, while the other queues get fractions 2/7 and 1/7 respectively. In Section 5 we already mentioned that for asymmetric systems, the Boxma-Meister approximation tends to infinity if ρ tends to 1, even though some queues are still stable. We see this phenomenon too in Figure 12: The mean end-to-end delay approximation is unbounded, whereas the simulated mean delay is still bounded if the load is 1. Other single-station approximations than that of Boxma and Meister might lead to more accurate results here.

8. CONCLUSION

Under the assumption that all nodes use HoL-based service disciplines, we have derived an exact expression for the mean overall end-to-end delay and an approximation for the mean type i_j end-to-end delay. The key steps in this approximation were: Equating the mean waiting time of type i_j packets in a node to that of all packets passing through the same queue at that node (Approx. 3.2), and application of the reduction result of [1]. These two steps combined result in an expression for the mean type i_j end-to-end delay in terms of the mean waiting time per queue in single-station polling systems.

For the 1-limited service discipline, Approximation 3.2 is very accurate over the entire parameter space of the smallest non-trivial tree. It is especially accurate for nearly symmetric systems and extremely asymmetric systems, and somewhat less accurate for moderately asymmetric systems.

In the special case that the subtree directly above queue i is symmetric, Approximation 3.2 becomes an exact statement rather than an approximation. If, in addition, exact results are available for the reduced system, the mean end-to-end delay per source can be determined exactly.

We applied the approximation for the mean end-to-end delay per type to a model based on a Network on Chip using the Boxma-Meister approximation [6] to obtain the necessary single-station results. Although the Boxma-Meister approximation is less accurate for asymmetric systems, we could still accurately approximate the mean type i_j end-to-end delay up to moderately high loads (around 0.7) in an asymmetric case study.

Other single-station approximations than that of Boxma and Meister are known, but they are often less accessible and transferring such approximations to discrete-time models usually involves subtleties. Depending on the precise characteristics of the tree (e.g., nearly symmetric, very asymmetric, etc.) these other approximations may lead to more accurate results. If the mean end-to-end delay approximation is applied to specific trees, one has to choose which single-station approximation to use based on the characteristics of the tree in order to obtain the most accurate results.

9. REFERENCES

[1] P. Beekhuizen, T. Denteneer, and J. Resing. Reduction of a polling network to a single node. *Queueing Systems*, 58(4):303–319, April 2008.

[2] C. Bisdikian. A note on the conservation law for queues with batch arrivals. *IEEE Transactions on Communications*, 41(6):832–835, June 1993.

[3] J. Blanc. An algorithmic solution of polling models with limited service disciplines. *IEEE Transactions on Communications*, 40(7):1152–1155, July 1992.

[4] O. Boxma and W. Groenendijk. Waiting times in discrete-time cyclic-service systems. *IEEE Transactions on Communications*, 36(2):164–170, February 1988.

[5] O. Boxma and B. Meister. Waiting-time approximations for cyclic-service systems with switch-over times. *ACM SIGMETRICS Performance Evaluation Review*, 14(1):254–262, May 1986.

[6] O. Boxma and B. Meister. Waiting-time approximations in multi-queue systems with cyclic service. *Performance Evaluation*, 7:59–70, 1987.

[7] H. Bruneel and B. Kim. *Discrete Time Models for Communication Systems including ATM*. Kluwer Academic Publishers, Norwell, MA, 1993.

[8] R. Chang and S. Lam. A novel approach to queue stability analysis of polling models. *Performance Evaluation*, 40:27–46, 2000.

[9] W. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *Proc. of the 38th Design Automation Conference*, pages 684–689, 2001.

[10] W. Groenendijk and H. Levy. Performance analysis of transaction driven computer systems via queueing

analysis of polling models. *IEEE Transactions on Computers*, 41(4):455–466, April 1992.

[11] O. Ibe and X. Cheng. Stability conditions for multiqueue systems with cyclic service. *IEEE Transactions on Automatic Control*, 33(1):102–103, January 1988.

[12] J. Morrison. A combinatorial lemma and its application to concentrating trees of discrete-time queues. *The Bell Systems Technical Journal*, 57(5):1645–1652, May-June 1978.

[13] M. Reiman and L. Wein. Heavy traffic analysis of polling systems in tandem. *Operations Research*, 47(4):524–534, July 1999.

[14] J. Resing. Polling systems and multitype branching processes. *Queueing Systems*, 13:409–426, 1993.

[15] I. Rubin and L. de Moraes. Message delay analysis for polling and token multiple-access schemes for local communication networks. *IEEE Journal on Selected Areas in Communications*, 1:935–947, 1983.

[16] M. Shalmon. Exact delay analysis of packet-switching concentrating networks. *IEEE Transactions on Communications*, 35(12):1265–1271, December 1987.

[17] M. Srinivasan. An approximation for mean waiting times in cyclic server systems with nonexhaustive service. *Performance Evaluation*, 9:17–33, 1988.

[18] H. Takagi. *Analysis of Polling Systems*. MIT Press, Cambridge, Massachusetts, 1986.

[19] H. Takagi. Queueing analysis of polling models: an update. In H. Takagi, editor, *Stochastic Analysis of Computer and Communication Systems*, pages 267–318. North-Holland, 1990.

[20] H. Takagi. Queueing analysis of polling models: progress in 1990-1994. In J. Dshalalow, editor, *Frontiers in Queueing: Models and Applications in Science and Engineering*, pages 119–146. CRC Press, Boca Raton, 1997.

[21] M. van Vuuren and E. Winands. Iterative approximation of k -limited polling systems. *Queueing Systems*, 55(3):161–178, March 2007.

[22] V. Vishnevskii and O. Semenova. Mathematical methods to study the polling systems. *Automation and Remote Control*, 67(2):173–220, February 2006.