Telecommunication Systems 6(1996)203–227

A mean-value analysis of slotted-ring network models

Andrew J. Coyle[a], Boudewijn R. Haverkort[b], William Henderson[a] and Charles E.M. Pearce[a]

[a]*Department of Applied Mathematics, The University of Adelaide, 5005 Adelaide, South Australia*

[b]*Department of Computer Science, Distributed Systems, Rheinisch-Westfälische Technische Hochschule (RWTH-Aachen), D-52056 Aachen, Germany*

Received January 1994; in final form March 1996

In this paper, we analyse Stochastic Petri Net (SPN) models of slotted-ring networks. We show that a simple SPN model of a slotted-ring network, which exhibits a product-form solution, yields similar results to a more detailed SPN model that has to be analysed by numerical means. Furthermore, we demonstrate a Mean-Value Analysis (MVA) approach to calculate efficiently the results for the simple model. This MVA approach allows for the movement of groups of tokens (customers) rather than just individual customers, as traditional MVA schemes for queueing network models do. Also, the MVA allows for non-disjoint place invariants, whereas previous MVA schemes addressed disjoint place invariants only. From the MVAs, it can be concluded that slotted-rings have very attractive performance characteristics, even under overload conditions (there is no "thrashing"). Also, we found that the choice of the slot size is a key factor in calibrating slotted-ring systems for optimal performance. Having a fast and reasonably accurate means available to evaluate the performance of slotted-ring systems, such as our proposed MVA, eases this calibration task. The proposed MVA for the product-form SPN models should therefore be regarded as a "quick engineering" tool.

1. Introduction

Slotted-ring networks have been proposed as interesting candidates for local and wide-area networks. Especially when large distances need to be covered, or when high transmission speeds are involved, such networks are known to behave in an attractive way, both from the user point of view (in terms of throughput and delay characteristics) and from a system efficiency point of view (not much of the available bandwidth is wasted) [28,29]. With the advent of B-ISDN and ATM, slotted-ring networks become of special interest, for example, as interconnection structure within ATM switches [3,4,20,25].

Over the last few years, considerable interest has been shown in using Stochastic Petri Nets (SPNs) for the modelling and analysis of a wide variety of computer and communication systems. One of the reasons for this is their flexibility and the availability of software tools to support the construction and solution of the SPN models. However, one of the limitations of the use of SPNs has been the size of the models, most notably, the number of states and transitions in the underlying Markov chains, which increases exponentially with the size of a network. Various procedures have been designed to cope with this problem. Symmetries have been exploited by Sanders and Meyer [26] to allow for the application of lumping theorems. Fixed-point iteration techniques have been employed by Ciardo and Trivedi [6] and a decomposition/aggregation approach has been studied by Henderson and Lucic [18]. Approximations based on reduced-load methods have been used by Coyle et al. [8] and state-space truncation techniques were employed successfully by Haverkort [14].

In this paper, two SPN models of slotted-ring networks are presented. The first is the more realistic one, but the amount of CPU time and computer memory required to solve this model is prohibitive in most cases. The second, more simplified SPN model, has a product-form solution for which a recursive MVA scheme is presented that calculates the performance measures of interest. The model is fairly abstract in comparison with some others that have been used for the analysis of slotted-ring networks (see section 2). However, the results are quite accurate and with the presented MVA technique, larger slotted-ring configurations can be analysed than has been possible hitherto. The MVA approach can therefore be regarded as a "quick engineering" approach towards the analysis of slotted-ring systems; see also [10], in which Van Dijk points out the usefulness of product-form results for bounding or quick engineering purposes.

Due to the inherent complexity of slotted rings, most slotted-ring models known from the literature can be analysed only by simulation (see, for example, [22,23,28]). We are aware of only a few performance analysis studies of slotted-ring systems in which analytical or numerical solution techniques are employed. Most notably, Ajmone Marsan et al. [1,2] and De Goei [15] employ SPN-based techniques to study various variants of slotted-rings, and Zafirovic and Niemegeers [28,29] derive approximate closed-form analytical expressions.

The paper is organised as follows. In section 2, we introduce slotted-ring systems and discuss typical system parameters and scenarios in which these systems are used. In section 3, two SPN models of slotted-ring systems are introduced. The first can be analysed only via its underlying Markov chain, which is shown to grow unwieldy. The second, more abstract, model is seen to approximate the first fairly accurately. It is then shown how results for this second model can be obtained very quickly using an MVA approach, which is derived in section 4. Some implementation aspects of the MVA scheme are considered there as well. In section 5, we present numerical examples and compare the MVA approach with a numerical solution approach. Section 6 provides a summary.

2. Slotted-ring systems

We discuss the general operation of slotted-ring systems and some typical scenarios of usage.

GENERAL OPERATION

In slotted-ring systems there are M stations, numbered $1,\ldots,M$, connected to a ring-shaped medium. On the medium k_0 slots circulate, each representing an equal part of the medium capacity. Slots consist of a small header followed by an information or data field. From the header, a station can decide whether a slot is free or in use. Whenever a slot passes a station, the station checks, from the address field in the header of the slot, whether this is the packet's destination. If so, it copies the contents of the slot; if not, it lets the slot pass. If a station wishes to transmit something itself, it waits for a free slot, takes it, and fills it with the address of the addressee and (part of) the message. As slots are of fixed length, a message may have to be split over a number of slots. Higher-layer protocols are assumed to take care of this splitting and the associated reassembly.

There are a number of ways of freeing slots. With destination release, a station that received a slot frees it. The receiving station can either use the slot again immediately, so-called immediate slot reuse, or pass it to its next downstream neighbour. With source release, on the other hand, the sending station on seeing its message return after one ring rotation frees the slot. Although destination release with immediate reuse is the more efficient, it is generally not recommended as it can create large relative unfairness between stations. Another way to enforce fairness is to allow station i to have a maximum of only k_i slots in use at any time.

As examples, the following slotted-ring systems can be mentioned. The CFR is a 100 Mbps slotted-ring system operating with source release. The used slot size is 256 bits [19,27,28]. The CFR-Variant is similar to the CFR except that it allows a station to take as many slots as desired at any time. Another variant of the CFR is Orwell, which is a destination-release slotted ring with a slot size of 128 bits operating with transmission speeds up to 565 Mbps [12,28]. Orwell was intended to be used as the transmission facility within a packet switch. Similar usage of slotted-ring networks have been proposed for internal usage in ATM switches. As ATM becomes more important, for WAN as well as for LAN technology, slotted-ring networks might become increasingly important in the near future as well [20].

TYPICAL SLOTTED-RING SCENARIOS

The presented scenarios are based on system descriptions found in the literature (see, for instance, [23,28]). The scenarios are summarised in table 1.

Scenario 1: A small symmetric system. Consider a slotted-ring system which is used to connect $M = 20$ stations on a medium with length 204.8 m. The roundtrip-delay

Table 1

Summary of the four scenarios.

Scenario	1	2	3	4
M	20	20	50	40
Length (m)	204.8	307.2	204.8	20k
Length (bits)	1024	1536	1024	16384
τ (μs)	10.24	15.36	10.24	100
r (Mbps)	100	100	100	163.84
k_0	4	6	4	16
Slots (bits)	256	256	256	1024
λ_1	10240	10240	4096	40960
$\lambda_{i\neq 1}$	10240	5389.47	4096	1050.26
sr/dr	sr	sr	sr	dr
μ_i	97656.25	65104.17	97656.25	20000
k_1	1	4	1	3
$k_{i\neq 1}$	1	1	1	1

$\tau = 10.24$ μs, assuming a propagation speed $c = 2 \times 10^7$ m/s. With a transmission rate $r = 100$ Mbps, the medium comprises 1024 bits or 4 slots of 256 bits each ($k_0 = 4$).

Assuming a 50% loading, the overall requested transmission capacity equals 50 Mbps. Assuming equally-loaded stations, the request rate of slots per station, $\lambda_i = 50$ Mbps/(20 × 256 bits per slot) = 10240.

Given that the system operates along the lines of source release, the time a slot is occupied always equals 1 roundtrip-delay, that is, $\mu_i = 1/\tau = 97656.25$. We assume furthermore that $k_i = 1$.

Scenario 2: A small asymmetric system. Consider a slotted-ring system which is used to connect $M = 20$ on a medium with length 307.2 m. Furthermore, $\tau = 15.36$ μs, $r = 100$ Mbps. Consequently, the medium comprises 1536 bits or 6 slots of 256 bits each ($k_0 = 6$).

Assuming a 50% loading, the overall requested transmission capacity equals 50 Mbps. However, we now assume that station 1 generates as much traffic as stations 2–20 together. Typically, such a station might be a file-server. We have $\lambda_1 =$ 25 Mbps/(256 bits per slot) = 102400 slots per second and $\lambda_j = \lambda_1/19 = 5389.47$ slots per second ($j = 2,\ldots,20$). To allow station 1 easier access to the ring, we set $k_1 = 4$ and $k_j = 1, j = 2,\ldots,20$. Finally, given that the system operates with source release, we have $\mu_i = 65104.17$.

Scenario 3: A large symmetric system. Scenario 3 is similar to scenario 1 except that now $M = 50$ stations are connected. The other system parameters are the same. Since we deal with more stations, the workload per station is smaller. We have $\lambda_i = 50$ Mbps/(50 × 256 bits per slot) = 4096. Further, we have $\mu_i = 97656.25$ and $k_i = 1$.

Scenario 4: A large asymmetric system. We now address a slotted-ring system of 200 km in length, connecting $M = 40$ stations, that is, $\tau = 100$ μs. With transmission rate $r = 163.84$ Mbps, the medium comprises 16384 bits or 16 slots of 1024 bits each, yielding $k_0 = 16$.

Assuming a 50% loading, the overall requested transmission capacity equals 81.92 Mbps. We assume that station 1 generates 50% of the load, stations 2–40 taking care of the other 50%. Consequently, we have $\lambda_1 = 40.96$ Mbps/(1024 bits per slot) = 40960 slots per second. Accordingly, $\lambda_j = \lambda_1/39 = 1050.26$. The slot restriction for station 1 is 3 ($k_1 = 3$). For the other stations it equals 1 ($k_j = 1$).

If we assume a uniform destination pattern of messages and destination release, we find that the average slot occupancy time is half a ring rotation time, or $\mu_i = 20000$.

3. SPN models of a slotted-ring system

We first discuss an SPN model of a slotted-ring system that does *not* exhibit a product-form solution. Then we discuss an SPN model that *does* have a product-form solution. Then we compare the two models for their computational complexity and accuracy.

AN SPN MODEL *WITHOUT* PRODUCT-FORM SOLUTION

Consider the Petri net model of the slotted-ring network shown in fig. 1 for the case when the number M of stations is equal to 4. Each station tries to use one of the slots every once in a while. There are k_0 available slots on the network, represented by tokens in place P_0. Each station is represented by only three places and transitions. For station i ($i = 1,\ldots,M$), we have places $P_{i,1}$, $P_{i,2}$ and $P_{i,3}$ and transitions $t_{i,1}$, $t_{i,2}$ and $t_{i,3}$. The initial numbers of tokens in places $P_{i,1}$, $P_{i,2}$ and $P_{i,3}$ are k_i, 0 and 0, respectively. The vector of the initial numbers of tokens is $\boldsymbol{k} = (k_0, k_1,\ldots,k_M)$, addressing only the nonzero initial markings, that is, the places P_0 and $P_{i,1}$. The firing times of transitions $t_{i,1}$ and $t_{i,3}$ (depicted as thick bars) are exponentially distributed, with transition rates λ_i and μ_i, respectively, and transition $t_{i,2}$ (depicted as a thin bar) is an immediate transition. We define $\rho_i = \lambda_i/\mu_i$. This SPN model has the input and output bags $I(t_{i,1}) = \{P_{i,1}\}$, $O(t_{i,1}) = \{P_{i,2}\}$, $I(t_{i,2}) = \{P_0, P_{i,2}\}$, $O(t_{i,2}) = \{P_{i,3}\}$, $I(t_{i,3}) = \{P_{i,3}\}$ and $O(t_{i,3}) = \{P_0, P_{i,1}\}$.

The model may be interpreted as follows. There are k_0 slots available on the slotted ring to be used by the stations. The maximum number of slots that station i is

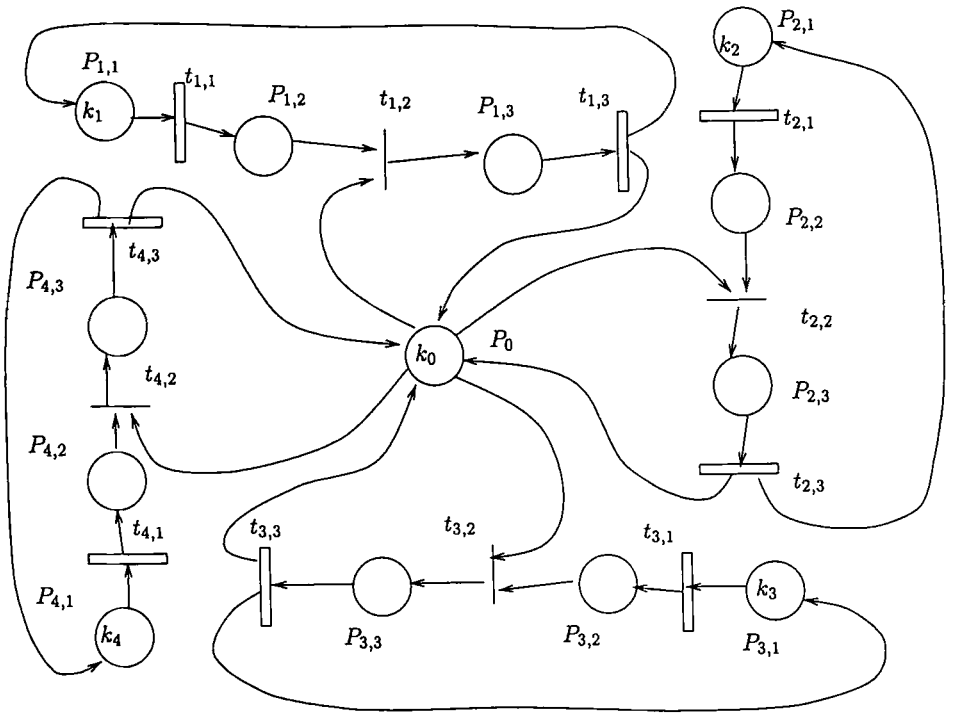

Fig. 1. An SPN model of a slotted-ring system with 4 stations.

either using or requesting at any one time is k_i. If there are tokens in place $P_{i,1}$, then transition $t_{i,1}$ is enabled. The firing of transition $t_{i,1}$ models a request by station i to use a slot. If transition $t_{i,1}$ fires, it takes a token from place $P_{i,1}$ and puts one into place $P_{i,2}$. A token in place $P_{i,2}$ represents a request by station i to use one of the ring's slots. Since transition $t_{i,2}$ is immediate, whenever there are any slots available on the ring, that is, if there are any tokens in place P_0, station i will take a slot immediately, putting a token in place $P_{i,3}$ to signify that the station is using a slot. If there are no slots available on the slotted ring, that is, place P_0 is empty, then the station will wait for a slot to become available, at which point it will immediately take that slot. If more than one station is waiting for a slot when it becomes available, then each station has an equal probability of getting that resource. The number of tokens in place $P_{i,3}$ represents the number of slots that station i is using. If there is at least one token in place $P_{i,3}$, then transition $t_{i,3}$ is enabled and so may fire, signifying that the station has finished using one of the slots it is using. Once station i has finished using a slot and $t_{i,3}$ fires, a token is put back into k_0, signifying that the slot is available, and a token is put back into $P_{i,1}$, signifying that this station has finished using the ring and may make another request.

The operation sketched can be used to model systems with source as well as with destination release by adjusting the time it takes to fire transitions $t_{i,3}$. In all

cases, immediate reuse of the slot by the releasing station is possible. The fact that the firing time of transitions $t_{i,3}$ is exponentially distributed is an approximation, since with source release the slot-usage time is exactly one roundtrip delay. For destination release, the slot-usage time is dependent on the traffic pattern between sources and destination. In this case, at least some randomness occurs, which makes this approximation less severe. In the case of the simpler model to be discussed in section 3, this approximation becomes exact due to insensitivity properties of the product-form solution.

Generally, $\sum_{i=1}^{M} k_i > k_0$, so that there is competition amongst the stations to take as many slots as needed. Whenever $\sum_{i=1}^{M} k_i \leq k_0$, the stations do not influence each other; they can operate and be analysed totally independently.

AN SPN MODEL *WITH* PRODUCT-FORM SOLUTION

Let us now consider a more abstract SPN model of the slotted-ring system (see fig. 2 for the case $M = 4$). This model resembles the previous one; however, it is assumed that a request for a slot that cannot be fulfilled immediately is discarded and does not queue. This implies that each station is now represented by only two places and transitions. For station i we have places $P_{i,1}$ and $P_{i,2}$ and transitions $t_{i,1}$ and $t_{i,2}$,

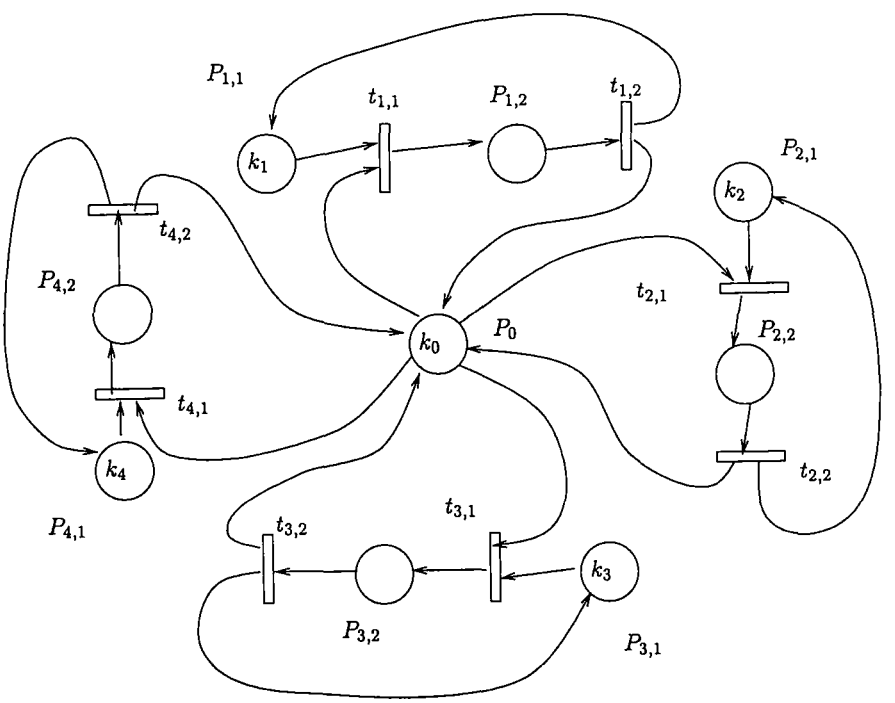

Fig. 2. An SPN model of a non-queueing slotted-ring system with 4 stations.

both with exponentially distributed transition firing times with rates λ_i and μ_i, respectively. The available slots are still represented by tokens in place P_0. The values ρ_i and $\boldsymbol{k} = (k_0,\ldots,k_M)$ are defined as before. This SPN is now specified by the input and output bags $I(t_{i,1}) = \{P_0, P_{i,1}\}$, $O(t_{i,1}) = \{P_{i,2}\}$, $I(t_{i,2}) = \{P_{i,2}\}$ and $O(t_{i,2}) = \{P_0, P_{i,1}\}$.

This model can be interpreted as before except that there are no places for tokens to queue between requesting a slot and using it. This gives a model that has a product-form solution and so results for this network can be found fairly efficiently using, for example, a mean-value analysis.

It also turns out that this model is insensitive to the service-time distributions, so a slotted-ring network with fixed service-time distributions will be modelled as accurately as one with exponential service-time distributions, only the mean service times, that is, the mean slot-usage times, being of importance. Thus, the simplified model is more accurate as far as slot-usage times are concerned for both the source and the destination release strategies.

The state of the SPN can be defined as $\boldsymbol{n} = (\boldsymbol{n}_0, \boldsymbol{n}_1,\ldots,\boldsymbol{n}_M)$, where $\boldsymbol{n}_0 = (n_0)$ and $\boldsymbol{n}_i = (n_{i,1}, n_{i,2})$, with n_0 the number of tokens in place P_0 and $n_{i,j}$ the number in place $P_{i,j}$ $(i = 1,\ldots,M; j = 1, 2)$. We then have the product-form solution

$$\pi(\boldsymbol{n}) = G(\boldsymbol{k})^{-1}\prod_{i=1}^{M}\frac{\rho_i^{n_{i,2}}}{n_{i,2}!} \quad \text{with} \quad G(\boldsymbol{k}) = \sum_{\boldsymbol{n}\in S(\boldsymbol{k})}\prod_{i=1}^{M}\frac{\rho_i^{n_{i,2}}}{n_{i,2}!} \tag{1}$$

for the steady-state probability distribution [11, 13, 16, 17].

This can be verified easily by showing that this product form satisfies the partial-balance equations

$$\begin{cases} \text{flux into } \boldsymbol{n} \text{ by firing } t_{i,1} = \text{flux out of } \boldsymbol{n} \text{ by firing } t_{i,2}, \\ \text{flux into } \boldsymbol{n} \text{ by firing } t_{i,2} = \text{flux out of } \boldsymbol{n} \text{ by firing } t_{i,1}, \end{cases}$$

which are equivalent to

$$\begin{cases} \pi(n_0+1, n_{1,1}, n_{1,2},\ldots, n_{i,1}+1, n_{i,2}-1,\ldots, n_{M,2})\lambda_i = \pi(n_0, n_{1,1},\ldots, n_{M,2})n_{i,2}\mu_i, \\ \pi(n_0-1, n_{1,1}, n_{1,2},\ldots, n_{i,1}-1, n_{i,2}+1,\ldots, n_{M,2})(n_{i,2}+1)\mu_i = \pi(n_0, n_{1,1},\ldots, n_{M,2})\lambda_i. \end{cases}$$

Note that it is because the partial-balance equations are satisfied that the model described here is insensitive to the service-time distribution functions.

COMPARISON OF THE TWO MODELS

Measures of interest. There are a number of interesting measures that can be derived for the above two models. Care should be taken to compare measures that, in some way, are equivalent to one another. As both models are not completely the same, some measures might be useful for one model, but less so for the other model. The following measures do not cause any difficulty:

- $m_{i,1}(\boldsymbol{k})$: the average number of outstanding messages at station i (that is, in $P_{i,1}$);
- $m_{i,2}(\boldsymbol{k})$: the average number of slots being used by station i (that is, $P_{i,2}$ in the case of the product-form model (PF), and $P_{i,3}$ for the non-PF model (non-PF));
- $\Lambda_{i,1}(\boldsymbol{k}) = \Lambda_{i,2}(\boldsymbol{k})$: the throughput of station i (that is, of transitions $t_{i,1}$ and $t_{i,2}$ in the PF model, and of $t_{i,1}$ and $t_{i,3}$ in the case of the non-PF model);
- σ: the probability that all slots are in use (that is, $\sigma = \Pr\{P_0 \text{ is empty}\}$).

Slightly more complicated is the definition of the "blocking-probability" B_i. For the PF model, B_i is the probability that $t_{i,1}$ is not enabled, that is, the probability that no progress can be made, due to the fact that there are no outstanding requests ($P_{i,1}$ empty), or there are no slots free (P_0 empty). For the non-PF model, we have chosen to define B_i as the probability that $P_{i,1}$ is empty, thus also disabling $t_{i,1}$. Note that P_0 does not play a role in the latter case. Since $\Lambda_i = \lambda_i(1 - B_i)$ (of course, as a function of $\boldsymbol{k}$), B_i provides a different view at the effective throughput that is reached, in comparison to the throughput requested (λ_i).

State space sizes. The first model does not have a product-form solution and numerical solutions for this model must be found such as those provided by the package SPNP [5]. The number of states in the Markov chain, however, is very large even for a moderate number of stations and slots. For example, consider scenario 1, with $M = 20$ and $k_0 = 4$. The number of states in the Markov chain is around 300 million, which is too large for a solution to be found. Suppose that we keep the same ring size but double the average interstation distance. This means that we must set $M = 10$ and keep k_0 equal to 4 (referred to as scenario 1′). The number of states in the underlying Markov chain then reduces to 13616.

For the PF model, the number $|S(\boldsymbol{k})|$ of states also increases very quickly with $\boldsymbol{k}$ and M; however, it remains smaller than for the non-PF model. For example, whenever each $k_i = 1$ ($i \neq 0$), we have

$$|S(\boldsymbol{k})| = \sum_{l=0}^{k_0} \binom{M}{l}.$$

Under the above assumptions, we have only 6196 states in scenario 1, and 386 states in scenario 1′.

Analysis results. Scenario 1′ has been analysed using both the product-form (PF-) and the non-PF model. The results are presented in table 2. For the non-PF model, the SPNP package needed about 20 minutes per evaluation (Sparc IPX), whereas the PF model only took a few seconds per evaluation (also with SPNP; Sparc IPX). For an increasing sequence of arrival rates λ_i, we present the blocking probability B_i, the effective throughput Λ_i, and the probability σ that all slots are in use. As can be observed, the PF model is slightly optimistic over the full range of arrival rates, that is, the blocking probabilities in the PF-model are slightly too small, the throughputs

Table 2

Comparing the PF and the non-PF SPNs for scenario 1′; numerical solution using SPNP in both cases.

Scenario 1′	Non-PF SPN; fig. 1			PF SPN; fig. 2		
λ_i	B_i	Λ_i	σ	B_i	Λ_i	σ
560	0.0058	556.71	0.0000	0.0057	556.80	0.0000
2560	0.0258	2493.92	0.0001	0.0256	2494.49	0.0001
5120	0.0504	4861.77	0.0010	0.0504	4862.16	0.0010
10240	0.1015	9200.46	0.0110	0.0999	9216.01	0.0094
15360	0.1588	12920.69	0.0384	0.1515	13033.38	0.0300
20480	0.2239	15894.50	0.0851	0.2039	16303.42	0.0616

slightly too high, and σ slightly too small. The differences are not very large, however. The difference in blocking probabilities is less than 9% (worst case), in throughput less than 2.5% (worst case), and for σ it is smaller than 25% (worst case). In the more moderate case of $l = 10240$, these percentages are 1.5%, 0.2%, and 1.5%, respectively.

Similar results are displayed for scenario 1″ in table 3. Scenario 1″ has been derived from scenario 1′ by making the workload more asymmetric, that is, by setting $\lambda_1 = 5.5\,\lambda$ and $\lambda_i = \lambda/2$ $(i = 2,\ldots,10)$. Notice that the overall workload for a given λ remains the same. Since station 1 is more heavily loaded, it is given more access opportunity, that is, by setting $k_1 = 2$, whereas all the other $k_i = 1$ $(i = 2,\ldots,10)$. Again notice the enormous difference in state space size: the non-PF model has 27642 states, the PF model only 432.

Table 3

Comparing the PF and the non-PF SPNs for scenario 1″; numerical solution using SPNP in both cases.

Scenario 1″	Non-PF SPN; fig. 1					PF SPN; fig. 2				
λ	B_1	B_i	Λ_1	Λ_i	σ	B_1	B_i	Λ_1	Λ_i	σ
560	0.0015	0.0030	5591.33	279.17	0.0000	0.0016	0.0028	5591.31	279.20	0.0000
2560	0.0265	0.0132	24921.86	1263.16	0.0002	0.0265	0.0131	24920.73	1263.27	0.0002
5120	0.0832	0.0272	46942.15	2490.41	0.0022	0.0830	0.0270	46949.28	2490.80	0.0020
10240	0.2156	0.0642	80318.31	4791.55	0.0191	0.2136	0.0617	80529.81	4804.17	0.0166
15360	0.3369	0.1155	101852.03	6792.40	0.0572	0.3291	0.1052	103042.94	6872.94	0.0462
20480	0.4394	0.1787	114816.83	8410.27	0.1130	0.4221	0.1534	118355.20	8669.39	0.0857

The observed differences between the PF- and the non-PF model can be understood as follows. Let us first address differences with respect to σ. In the PF model, $t_{i,1}$ is only enabled if both $P_{i,1}$ and P_0 are non-empty. Therefore, whenever P_0 is empty,

new requests for slots are effectively not produced. On the other hand, in the non-PF model, $t_{i,1}$ only requires $P_{i,1}$ to be non-empty. Thus, even if P_0 is empty, requests might queue. As a consequence, whenever P_0 becomes non-empty, there might be immediate transitions enabled that directly empty P_0 again. In the PF model, this is not possible, since only at the time when P_0 becomes non-empty again, new arrival epochs taking exponential time will start. Thus, the probability that P_0 is empty will be larger in the non-PF models than in the PF models. The larger the utilization of the system, the larger this difference will grow.

Regarding the throughputs, it can be observed that they are larger in the PF models. This is due to the fact that the extra delay per cycle, that is, the waiting for an empty slot (next to the standard delay introduced by the service and interarrival processes), only takes place when P_0 is empty. Since the latter is true in the PF models with smaller probability, the probability on such an extra delay due to congestion is smaller in these models. Therefore, the time for a customer to cycle once around is smaller, thus making the throughput larger.

Comparing the blocking probabilities is the most difficult. At first instance, one is tempted to assume that B_i is smaller in the case of the non-PF models, simply because it sums the probability of fewer states, that is, $B_i = \Pr\{\#P_0 = 0 \vee \#P_{i,1} = 0\}$ (in the PF case), whereas $B_i = \Pr\{\#P_{i,1} = 0\}$ (in the non-PF case). This is, however, not true. Although in the non-PF models fewer states are taken into account, their sum is larger. This must then be due to the fact that the different model structure divides the probability mass in a different way over the states. A more precise explanation can not be found for this phenomenon.

In conclusion, we can state that, also when using SPNP, a large computational gain is attained when using the PF model, at the cost of only limited loss of accuracy. Furthermore, an even quicker solution is available for the PF model using a mean-value analysis. This will be demonstrated in section 4.

4. Mean-value analysis recursion

In this section, we derive an MVA for the SPN models presented in section 3. A general MVA for product-form *batch-movement queues* has been presented (rather compact) in [9]; the derivation here operates along similar lines; however, it is tailored towards product-form *stochastic Petri nets* and provides more details on the individual computational steps to be taken. We first present the place invariants (for the model presented in section 3) and then derive expressions for the mean place occupancies. We also pay special attention to the actually employed and implemented recursion scheme.

PLACE INVARIANTS

To derive the MVA recursion scheme, we first obtain the S-invariants of the model (see fig. 2). There is one such invariant for each station i:

$$S_i : n_{i,1} + n_{i,2} = k_i, \quad i = 1, \ldots, M.$$

There is also a "global" place invariant:

$$S_0 : n_0 + \sum_{i=1}^{M} n_{i,2} = k_0.$$

Note that S_i and S_0 are not disjoint. The set $S(\boldsymbol{k})$ of all possible states is now given as

$$S(\boldsymbol{k}) = \{\boldsymbol{n} \in \mathbb{N}^{2M+1} | S_0, S_1, \ldots, S_M\}.$$

AVERAGE PLACE OCCUPATION

We now proceed with the derivation of the average number of tokens in places P_0, $P_{i,1}$ and $P_{i,2}$, given $\boldsymbol{k}$, which we denote, respectively, by $m_0(\boldsymbol{k})$, $m_{i,1}(\boldsymbol{k})$ and $m_{i,2}(\boldsymbol{k})$. First we address $m_{i,1}(\boldsymbol{k})$. The expected number of tokens in $P_{i,1}$ is

$$m_{i,1}(\boldsymbol{k}) = \sum_{l=1}^{k_i} l \cdot \sum_{\boldsymbol{n} \in \mathcal{N}_{i,l}(\boldsymbol{k})} \pi(\boldsymbol{n}), \tag{2}$$

where $\mathcal{N}_{i,l}(\boldsymbol{k})$ is the set of states with l tokens in place $P_{i,1}$, that is

$$\mathcal{N}_{i,l}(\boldsymbol{k}) = \{\boldsymbol{n} \in \mathbb{N}^{2M+1} | S_0, S_1, \ldots, S_{i-1}, l + n_{i,2} = k_i, S_{i+1}, \ldots, S_M\}.$$

Substituting (1) into (2), we obtain

$$m_{i,1}(\boldsymbol{k}) = \frac{1}{G(\boldsymbol{k})} \sum_{l=1}^{k_i} l \cdot \sum_{\boldsymbol{n} \in \mathcal{N}_{i,l}(\boldsymbol{k})} \prod_{h=1}^{M} \frac{\rho_h^{n_{h,2}}}{n_{h,2}!}$$

$$= \frac{G(\boldsymbol{k} - \boldsymbol{e}_i)}{G(\boldsymbol{k})} \sum_{l=0}^{k_i - 1} (l+1) \cdot \sum_{\boldsymbol{n} \in \mathcal{N}_{i,l}(\boldsymbol{k} - \boldsymbol{e}_i)} \frac{\prod_{h=1}^{M} \frac{\rho_h^{n_{h,2}}}{n_{h,2}!}}{G(\boldsymbol{k} - \boldsymbol{e}_i)}$$

$$= \frac{G(\boldsymbol{k} - \boldsymbol{e}_i)}{G(\boldsymbol{k})} \left(\sum_{l=0}^{k_i - 1} l \cdot \sum_{\boldsymbol{n} \in \mathcal{N}_{i,l}(\boldsymbol{k} - \boldsymbol{e}_i)} \frac{\prod_{h=1}^{M} \frac{\rho_h^{n_{h,2}}}{n_{h,2}!}}{G(\boldsymbol{k} - \boldsymbol{e}_i)} + \sum_{l=0}^{k_i - 1} \sum_{\boldsymbol{n} \in \mathcal{N}_{i,l}(\boldsymbol{k} - \boldsymbol{e}_i)} \frac{\prod_{h=1}^{M} \frac{\rho_h^{n_{h,2}}}{n_{h,2}!}}{G(\boldsymbol{k} - \boldsymbol{e}_i)} \right).$$

Here, $\boldsymbol{e}_i$ is an $(M+1)$-vector with unity in place i $(0 \le i \le M)$ and zeros elsewhere. By the definition of $G(\boldsymbol{k})$, the second sum in the parentheses equals 1. The first term equals the average numer of tokens in place $P_{i,1}$ given that there are initially $\boldsymbol{k} - \boldsymbol{e}_i$ tokens. Consequently, we have

$$m_{i,1}(\boldsymbol{k}) = \frac{G(\boldsymbol{k}-\boldsymbol{e}_i)}{G(\boldsymbol{k})}[m_{i,1}(\boldsymbol{k}-\boldsymbol{e}_i)+1]. \tag{3}$$

In a similar way, we derive

$$m_0(\boldsymbol{k}) = \frac{G(\boldsymbol{k}-\boldsymbol{e}_0)}{G(\boldsymbol{k})}[m_0(\boldsymbol{k}-\boldsymbol{e}_0)+1]. \tag{4}$$

The derivation of $m_{i,2}(\boldsymbol{k})$ is only slightly more complicated. Let $\mathcal{M}_{i,1}(\boldsymbol{k})$ be the set of states with l tokens in place $P_{i,2}$, that is

$$\mathcal{M}_{i,l}(\boldsymbol{k}) = \left\{\boldsymbol{n} \in \mathbb{N}^{2M+1} \mid n_0 + \sum_{g=1, g\neq i}^{M} n_{g,2} + l = k_0; S_1; \cdots; S_{i-1}; n_{i,1} + l = k_i; S_{i+1}; \cdots; S_M\right\}.$$

Now we have

$$\begin{aligned} m_{i,2}(\boldsymbol{k}) &= G(\boldsymbol{k})^{-1} \sum_{l=1}^{k_i} l \cdot \sum_{\boldsymbol{n}\in\mathcal{M}_{i,l}(\boldsymbol{k})} \prod_{h=1}^{M} \frac{\rho_h^{n_{h,2}}}{n_{h,2}!} \\ &= G(\boldsymbol{k})^{-1} \sum_{l=1}^{k_i} l \cdot \sum_{\boldsymbol{n}\in\mathcal{M}_{i,l}(\boldsymbol{k})} \left(\prod_{h=1, h\neq i}^{M} \frac{\rho_h^{n_{h,2}}}{n_{h,2}!}\right) \frac{\rho_i^l}{l!} \\ &= G(\boldsymbol{k})^{-1} \rho_i \sum_{l=0}^{k_i-1} \sum_{\boldsymbol{n}\in\mathcal{M}_{i,l}(\boldsymbol{k}-\boldsymbol{e}_0-\boldsymbol{e}_i)} \prod_{h=1}^{M} \frac{\rho_h^{n_{h,2}}}{n_{h,2}!} \\ &= \rho_i \frac{G(\boldsymbol{k}-\boldsymbol{e}_i-\boldsymbol{e}_0)}{G(\boldsymbol{k})}. \end{aligned}$$

For convenience, we introduce the ancillary quantities

$$\gamma_0(\boldsymbol{k}) = \frac{G(\boldsymbol{k}-\boldsymbol{e}_0)}{G(\boldsymbol{k})}, \quad \gamma_i(\boldsymbol{k}) = \frac{G(\boldsymbol{k}-\boldsymbol{e}_i)}{G(\boldsymbol{k})} \tag{5}$$

and

$$\begin{cases} w_0(\boldsymbol{k}) = m_0(\boldsymbol{k}-\boldsymbol{e}_0)+1, \\ w_{i,1}(\boldsymbol{k}) = m_{i,1}(\boldsymbol{k}-\boldsymbol{e}_i)+1, \\ w_{i,2}(\boldsymbol{k}) = 1. \end{cases} \tag{6}$$

The γ's enjoy the convenient property that

$$\begin{aligned} \gamma_i(\boldsymbol{k}-\boldsymbol{e}_0)\gamma_0(\boldsymbol{k}) &= \frac{G(\boldsymbol{k}-\boldsymbol{e}_0-\boldsymbol{e}_i)}{G(\boldsymbol{k}-\boldsymbol{e}_0)} \frac{G(\boldsymbol{k}-\boldsymbol{e}_0)}{G(\boldsymbol{k})} \\ &= \frac{G(\boldsymbol{k}-\boldsymbol{e}_i-\boldsymbol{e}_0)}{G(\boldsymbol{k}-\boldsymbol{e}_i)} \frac{G(\boldsymbol{k}-\boldsymbol{e}_i)}{G(\boldsymbol{k})} = \gamma_0(\boldsymbol{k}-\boldsymbol{e}_i)\gamma_i(\boldsymbol{k}). \end{aligned} \tag{7}$$

The quantity $w_{i,j}(\boldsymbol{k})$ can be interpreted as the waiting time of a token in place $P_{i,j}$, including its own firing time, normalised by the corresponding transition rate, that is, λ_i in the case of $w_{i,1}(\boldsymbol{k})$ and $n_{i,2}\mu_i$ in the case of $w_{i,2}(\boldsymbol{k})$. In this notation, the recursive relations for the average number of tokens can be written as

$$\begin{cases} m_0(\boldsymbol{k}) = \gamma_0(\boldsymbol{k}) w_0(\boldsymbol{k}), \\ m_{i,1}(\boldsymbol{k}) = \gamma_i(\boldsymbol{k}) w_{i,1}(\boldsymbol{k}), \\ m_{i,2}(\boldsymbol{k}) = \rho_i \gamma_i(\boldsymbol{k} - \boldsymbol{e}_0)\gamma_0(\boldsymbol{k}). \end{cases} \tag{8}$$

From the invariant S_i, we have $m_{i,1}(\boldsymbol{k}) + m_{i,2}(\boldsymbol{k}) = k_i$. Substituting the expressions derived earlier for $m_{i,1}(\boldsymbol{k})$ and $m_{i,2}(\boldsymbol{k})$ in this S-invariant, we obtain

$$\gamma_i(\boldsymbol{k}) w_{i,1}(\boldsymbol{k}) + \rho_i \gamma_i(\boldsymbol{k} - \boldsymbol{e}_0)\gamma_0(\boldsymbol{k}) = k_i,$$

which can be rewitten as

$$\gamma_i(\boldsymbol{k})[w_{i,1}(\boldsymbol{k}) + \rho_i \gamma_0(\boldsymbol{k} - \boldsymbol{e}_i)] = k_i$$

so that

$$\gamma_i(\boldsymbol{k}) = \frac{k_i}{w_{i,1}(\boldsymbol{k}) + \rho_i \gamma_0(\boldsymbol{k} - \boldsymbol{e}_i)}. \tag{9}$$

Similarly, from the invariant S_0 we know that $m_0(\boldsymbol{k}) + \sum_{i=1}^{M} m_{i,2}(\boldsymbol{k}) = k_0$. This leads to

$$\gamma_0(\boldsymbol{k}) = \frac{k_0}{w_0(\boldsymbol{k}) + \sum_{i=1}^{M} \rho_i \gamma_i(\boldsymbol{k} - \boldsymbol{e}_0)}. \tag{10}$$

RECURSION SCHEME AND INITIAL CONDITIONS

Equations (6), (8), (9) and (10) provide the general step of the MVA recursion scheme. One of the advantages of this scheme over other analytic techniques is that the values calculated at each stage of the recursion are relevant to the system being analysed. For example, assume that a slotted ring has a fixed set of arrival and transmission rates and the number of stations is to be chosen such that certain performance requirements are met. The mean-value analysis need only be performed once, the performance measures needed at each stage being calculated using those from the previous stage. If this problem is solved using some other technique, for example solving the underlying Markov chain using SPNP, a new analysis must be peformed each time we change the number of stations in the network.

It remains to supply initial conditions. For $\boldsymbol{k} = h\boldsymbol{e}_0$, with $h \geq 0$, the process has only the singleton state $h\boldsymbol{e}_0$ and from (1), $G(\boldsymbol{k}) = 1$. Hence, by definition

$$\gamma_0(h_0\boldsymbol{e}_0) = 1 \quad \text{for} \quad h_0 > 0.$$

Similarly, we find

$$\gamma_i(h_1\boldsymbol{e}_1 + \cdots + h_M\boldsymbol{e}_M) = 1 \quad (h_i > 0; h_j \geq 0, j \neq i).$$

Likewise, when only station i generates requests and there are slots available, the normalised waiting times for station i equal 1, that is

$$\begin{cases} w_{i,1}(h\boldsymbol{e}_0 + \boldsymbol{e}_i) = 1 & (h > 0), \\ w_{i,2}(h\boldsymbol{e}_0 + \boldsymbol{e}_i) = 1 & (h > 0). \end{cases}$$

We note that when $\boldsymbol{k}$ is an invalid state, then $\gamma_i(\boldsymbol{k}) = 0$ $(i = 0,\ldots,M)$.

TRANSITION THROUGHPUT

Because of the infinite number of servers, the throughput $\Lambda_{i,2}(\boldsymbol{k})$ of transition $t_{i,2}$ is equal to the average number of customers in $P_{i,2}$ multiplied by the service rate. That is,

$$\Lambda_{i,1}(\boldsymbol{k}) = m_{i,2}(\boldsymbol{k})\mu_i = \lambda_i\gamma_i(\boldsymbol{k} - \boldsymbol{e}_0)\gamma_0(\boldsymbol{k}).$$

Due to the structure of the SPN, we have $\Lambda_{i,1}(\boldsymbol{k}) = \Lambda_{i,2}(\boldsymbol{k})$.

IMPLEMENTATION CONSIDERATIONS

For ease of computation, we can perform the mean-value analysis using only the γ's. The relevant equations are

$$\gamma_0(\boldsymbol{k}) = \frac{k_0}{k_0 - \gamma_0(\boldsymbol{k} - \boldsymbol{e}_0)\sum_{l=1}^{M}\rho_l\gamma_l(\boldsymbol{k} - 2\boldsymbol{e}_0) + \sum_{l=1}^{M}\rho_l\gamma_l(\boldsymbol{k} - \boldsymbol{e}_0)}$$

and (11)

$$\gamma_l(\boldsymbol{k}) = \frac{k_l}{k_l + \rho_l\gamma_0(\boldsymbol{k} - \boldsymbol{e}_i)[1 - \gamma_l(\boldsymbol{k} - \boldsymbol{e}_0 - \boldsymbol{e}_l)]}, \quad l = 1,\ldots,M.$$

The initial conditions are

$$\begin{cases} \gamma_l(\boldsymbol{k}) = 0 & \text{if} \quad \boldsymbol{k} < \boldsymbol{0} & (l = 0,\ldots,M), \\ \gamma_0(\boldsymbol{k}) = 0 & \text{for} \quad k_0 = 0;\ k_i \geq 0 & (i = 1,\ldots,M), \\ \gamma_0(\boldsymbol{k}) = 1 & \text{for} \quad k_0 > 0;\ k_i = 0 & (i = 1,\ldots,M), \\ \gamma_l(\boldsymbol{k}) = 0 & \text{for} \quad k_l = 0;\ k_i \geq 0 & (i,l = 1,\ldots,M), \\ \gamma_l(\boldsymbol{k}) = 1 & \text{for} \quad k_0 = 0;\ k_l > 0;\ k_i \geq 0 & (i,l = 1,\ldots,M). \end{cases} \tag{12}$$

The blocking probability B_i for transition $t_{i,1}$ is given by

$$B_i = 1 - \gamma_0(\boldsymbol{k})\gamma_i(\boldsymbol{k} - \boldsymbol{e}_0).$$

The program used to obtain the results in this paper has been written in C and uses a straightforward procedural recursion scheme. Savings can be made when the MVA problem possesses some symmetry. Equation (11) calculates in turn $\gamma_1(\boldsymbol{k} - 2\boldsymbol{e}_0)$, $\gamma_2(\boldsymbol{k} - 2\boldsymbol{e}_0)$,... . In some cases, these will be the same and so only one needs to be calculated, thereby saving some computation time.

5. Numerical examples

In this section, we apply the MVA approach to the modelling of the various slotted-ring configurations discussed in section 2. First, we compare our solution technique with a numerical solution approach performed with SPNP [5]. We then focus on the performance of the slotted-ring systems themselves. Finally, we comment on the validity and exactness of the model.

EVALUATION OF THE MVA APPROACH

Scenario 1 can be analysed numerically using SPNP as well as with our MVA algorithm. For the purpose of comparison, we propose two variants of this scenario. In scenario 1A, the ring is lengthened to exactly 5 slots, that is, $\tau = 12.8\ \mu s$ and $\mu_i = 78125$. In this scenario, we also give more access rights to station 1, that is, we set $k_1 = 4$, whereas every other $k_j = 1$. Scenario 1B resembles scenario 1A; however, we set $k_1 = 4$ and $k_j = 2$. Table 4 summarises the three used scenarios.

Table 4

Summary of scenarios 1, 1A and 1B.

Measure	Scenario 1	Scenario 1A	Scenario 1B
τ	10.24	12.8	12.8
k_0	4	5	5
μ_i	97656.25	78125	78125
k_1	1	4	4
$k_{i\neq 1}$	1	1	2
All other parameters as in scenario 1			

We present the results of these analyses in table 5. The results for the SPNP-based and the MVA-based solution techniques are exactly the same, for the cases 1, 1A and 1B, as expected. We comment on column 1B* below. Without entering into details now, we observe that the total carried traffic (row $\sum_i \Lambda_i$) is smaller than the 50 Mbps actually requested by the application. This is due to competition for the slots. The overall bandwidth obtained for station 2 to 20 in scenario 1B is increased in comparison with 1A because $k_j = 2$.

Table 5

Some performance measures for scenarios 1, 1A, 1B and 1B*.

Measure	Scenario 1	Scenario 1A	Scenario 1B	Scenario 1B*
$m_{1,1}(k)$	0.9114	3.8759	3.8791	3.9017
$m_{j,1}(k)$	0.9114	0.8888	1.8796	1.9020
$m_{1,2}(k)$	0.0886	0.1214	0.1209	0.0983
$m_{j,2}(k)$	0.0886	0.1112	0.1204	0.0984
$\Lambda_{1,1}(k)$ (slots/sec)	8.6553×10^3	9.6910×10^3	9.445×10^3	7.6790×10^3
$\Lambda_{j,1}(k)$ (slots/sec)	8.6553×10^3	8.6880×10^3	9.406×10^3	7.6597×10^3
$\Sigma_i \Lambda$ (Mbps)	42.262	42.666	45.937	37.4056
$B_1(k)$	0.1548	0.0536	0.0776	0.2501
$B_j(k)$	0.1548	0.1516	0.0820	0.2520

Table 6

The runtime (in seconds) and the sizes of the underlying Markov chains, for the MVA, the SPNP, and the SPNP* solution approach (measured on a SUN SPARC 4/65).

Solver	Measure	Scenario 1	Scenario 1A	Scenario 1B	Scenario 1B*
MVA	time	< 0.1	0.2	0.3	–
SPNP	time	137.9	383.9	788.2	–
	states	6196	23071	48740	–
	transitions	46400	211478	401098	–
SPNP*	time	0.5	0.7	–	0.8
	states	16	35	–	45
	transitions	52	128	–	172

Table 6 compares the solution times for the MVA and the SPNP approach (row MVA, and the two rows for SPNP). We measure only the solution part in the SPNP case, not the compilation and construction part. We also indicate the number of states and transitions in the underlying Markov chain. It appears that the MVA method is the most favourable.

However, as already mentioned in section 4, the MVA approach takes symmetries in the model into account, which the SPNP approach does not. When we take these symmetries into account in the SPNP models, by amalgamating 3 to 20 and keeping 1 and 2 separate, as representatives of the high and low load stations, we obtain the solution times of table 6 (rows for SPNP*). The SPNP approach, now denoted SPNP*, is now almost as fast as the MVA. It should be noted that the adapted SPN yields

exact results only for scenarios 1 and 1A. For scenario 1B* it gives the approximate results given in column 1B* in table 5. This is because in scenario 1B*, the request rate for slots for the amalgamated stations is nonlinear in the number of token in place $P_{3,1}$ which represents these stations. However, linearity was assumed in the SPNP model. In the case $k_j = 1$, that is, scenarios 1 and 1A, the above linearity does hold.

As a final remark here, it should be noted that the results derived in section 3 (tables 2 and 3) have been derived using the above symmetries. By the fact that in the employed scenarios there $k_i = 1$, for all i, this symmetry exploitation is exact.

PERFORMANCE RESULTS FOR SLOTTED-RING SYSTEMS

First we analyse scenarios 2–4. Then we perform a sensitivity analysis of the blocking probabilities and throughputs when we vary the offered load. We also study the influence of the slot size on the blocking probabilities and throughput.

Scenarios 2 to 4. Table 7 shows the results for scenarios 2 to 4 obtained with the MVA approach. The computation for scenario 4 took a very long time to complete. It can be observed that in all cases the asymmetry has influence on the bandwidth division between stations. Further, in all cases, the requested bandwidth is not reached (50 Mbps for scenarios 2 and 3, and 81.92 Mbps for scenario 4).

Table 7

Some performance measures for scenarios 2, 3 and 4.

Measure	Scenario 2	Scenario 3	Scenario 4
$m_{1,2}(\boldsymbol{k})$	1.1451	0.0367	1.6022
$m_{j,2}(\boldsymbol{k})$	0.0735	0.0367	0.0500
$\Lambda_{1,1}(\boldsymbol{k})$ (slots/sec)	94.49×10^3	3.581×10^3	32.04×10^3
$\Lambda_{j,1}(\boldsymbol{k})$ (slots/sec)	4.787×10^3	3.581×10^3	1.000×10^3
$\sum_i \Lambda_i$ (Mbps)	45.274	43.708	69.379
$B_1(\boldsymbol{k})$	0.0773	0.1249	0.2177
$B_j(\boldsymbol{k})$	0.1120	0.1249	0.0500

Changing the offered load. We now evaluate scenario 2 under a varying load. In the basic scenario, we put a total load of 50 Mbps on a 100 Mbps slotted-ring network. The overall throughput turned out to be about 45 Mbps (see table 7).

Figure 3 shows the blocking probabilities B_1 and B_j ($j = 2,\ldots,20$) with varying imposed load. Note that the x-axis depicts the load on station 1, or equivalently, the sum of the loads on the other stations. Consequently, to the right of 50 Mbps we overload the system. The blocking probabilities with therefore increase to 1. Notice

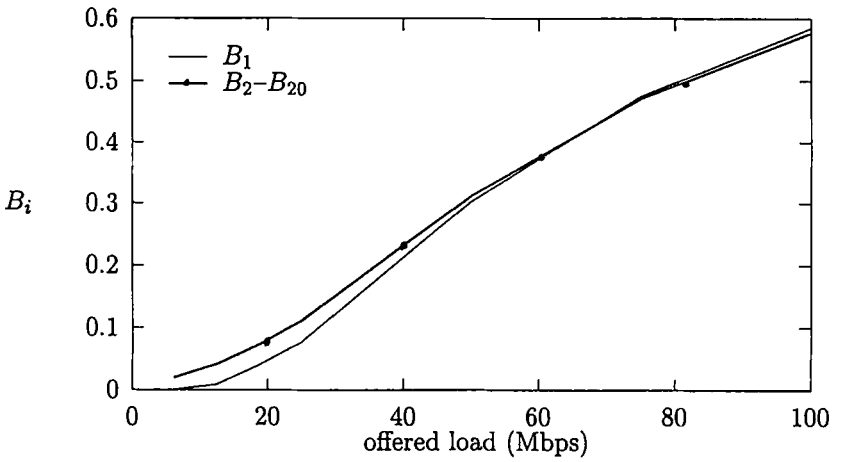

Fig. 3. The blocking probabilities B_1 and B_j for scenario 2 with varying imposed load.

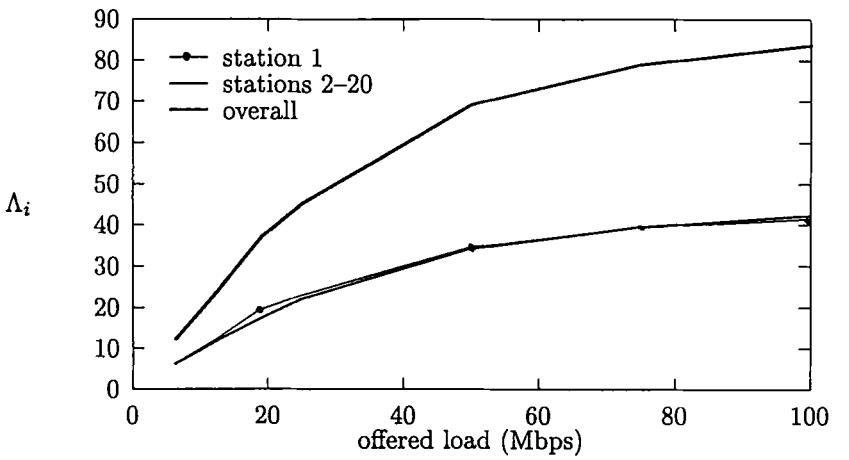

Fig. 4. The offered load versus the obtained throughput.

that $B_1 < B_j$ whenever the system is not overloaded and that the opposite holds once the system becomes heavily saturated.

Figure 4 shows the offered load to station 1, or equivalently, to all the other stations together, versus the obtained throughput, for station 1, the other stations and all stations together. As can be observed, the slotted-ring mechanism divides the available bandwidth amongst the stations according to the ratios of their requested throughputs.

The choice of the slot size. One of the parameters that influences the performance of a slotted-ring system is the number of slots into which the medium is divided. Having large slots implies that the number of slots to be used for a given application data

packet is small. On the other hand, having large slots implies having few slots given a fixed medium length. Consequently, the time it takes to get access to a slot is large. This trade-off has been observed before and has been the motivation for the design of a slotted-ring system with adjustable slot sizes [22,23].

Consider a system with $M = 50$ stations which are assumed to behave identically. With a limitation of 1 slot per station at any time, that is, $k_i = 1$, the number of slots the medium can be sensibly divided into ranges from 1 to 50. We assume that all divisions are possible and interpolate where this implies having slots with non-integral sizes.

Now assume that the offered load is such that whenever the number of slots equals $k_0 = 10$, the ratio $\rho_i = \lambda_i/\mu_i = 0.1$ Without loss of generality, assume that $\mu_i = 1$. Whenever we increase the number of slots by a factor f, the slot size decreases by a factor f. Consequently, the rate λ_i at which slot requests arrive has to increase by a factor f in order to ensure the same amount of application data is transmitted.

Table 8 shows the blocking probability B_i, λ_i and $\Lambda_{i,2}$ as a function of the number of slots k_0. The first thing to observe is that B_i shows a minimum for $k_0 = 7$, whereas $\Lambda_{i,2}$ is increasing with k_0. Thus, the above-mentioned trade-off is indeed in operation. We observe further that $\Lambda_{i,2} = \lambda_i(1 - B_i)$. Since the MVA only gives results

Table 8

The blocking probability B_i, λ_i and $\Lambda_{i,2}$ as a function of the number of slots k_0.

k_0	λ_i	B_i	$\Lambda_{i,2}$	k_0	λ_i	B_i	$\Lambda_{i,2}$
1	0.01	0.3333	0.0067	8	0.08	0.0892	0.0729
2	0.02	0.2048	0.0159	9	0.09	0.0925	0.0817
3	0.03	0.1460	0.0256	10	0.10	0.0973	0.0903
4	0.04	0.1151	0.0354	11	0.11	0.1032	0.0987
5	0.05	0.0987	0.0451	12	0.12	0.1097	0.1068
6	0.06	0.0908	0.0546	15	0.15	0.1310	0.1304
7	0.07	0.0882	0.0638	50	0.50	0.3333	0.3333

within reasonable time up to the case $k_0 = 15$ (which took 8 CPU minutes on our SPARC IPX workstation), we might use this equation to calculate $\Lambda_{i,2}$ in the case $k_0 = 50$, as follows.

When $k_0 = 50$, the stations operate independently. The only reason for station i to block is the unavailability of a token in place $P_{i,1}$ because $n_0 > 0$ as long as $n_{i,1} > 0$. The probability of this equals that of having one customer in an M/M/1/1 queue with customer arrival rate λ_i and service rate μ_i. Consequently, we have $B_i = \lambda_i/(\mu_i + \lambda_i)$ $= \rho_i/(1 + \rho_i) = 0.3333$. Thus, $\Lambda_{i,2} = 0.5(1 - 0.3333) = 0.3333$.

In the asymmetric case, we observe similar behaviour. Figure 5 shows the blocking probabilities B_1 and B_j $(j \neq 1)$ as a function of the number of slots in the

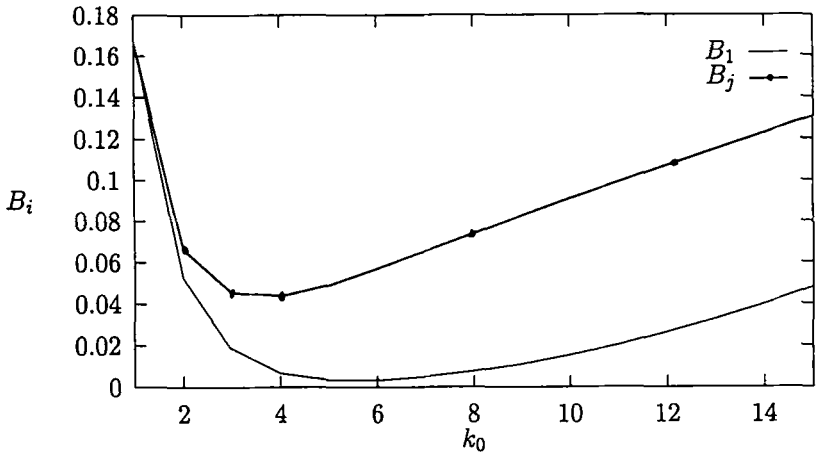

Fig. 5. The blocking probabilities B_1 and B_j ($j \neq 1$) as a function of the number of slots k_0.

system. As before, we adapted the request rates so that the overall application data volume to be transmitted remains constant. The other parameters are as follows. We have $M = 11$ stations, and the request rate of station 1 is as high as the sum of the request rates of stations 2 to 11. We set $k_j = 1$ ($j \neq 1$) and $k_1 = \min\{4, k_0\}$. In this way, we allow station 1 to take at most 4 slots or all slots (if less than 4) at any time.

It is interesting to observe that the optimum number of slots is different for station 1 and the other stations. For station 1, $k_0 = 5$ would be the best choice, whereas for stations 2 to 11, $k_0 = 4$ would be best. Notice that the number of slots that gives rise to the smallest blocking probability also gives rise to the largest throughput, by the simple relation $\Lambda_{i,2} = \lambda_i(1 - B_i)$.

In this case, we were able to calculate the blocking probabilities for up to $k_0 = 15$, that is, the "independent" case, with our MVA. In order to check the MVA results, we also calculated B_j from the probability of having one customer in an M/M/1/1 queue with customer arrival rate λ_i and service rate μ_i as before. Indeed, this calculation also reveals that $B_j = 0.1304$. Similarly, B_1 follows from the probability of having four customers in an M/M/4/4 queue with customer arrival rate λ_1 and service rate $k\mu_1$ if there are k customers and $\rho_1 = 1.5$, that is,

$$B_1 = K^{-1}\frac{\rho_1^4}{4!} = 0.0480 \quad \text{with} \quad K = \sum_{i=0}^{4} \rho_1^i/i!. \tag{13}$$

VALIDITY AND COMPARISON OF THE MODELS

The slotted-ring model presented here are rather abstract. Although they omit many system details, they *do* reveal behaviour patters that others have found after much more time-consuming simulation studies [22,23,28] or with numerical analyses [1,2,15,29]. The question is, we believe, not so much whether our models mimic

reality very precisely as whether they predict the overall performance behaviour, expressed in terms of a selection of performance measures (see section 3), correctly. We think our models satisfy this last requirement, also when comparing to other modelling attempts, as done briefly below.

Pasch et al. [22,23] evaluated the performance of slotted-ring system as a function of the slot size and studied the influence of complex (multimedia look-alike) workloads on the system performance. The latter studies can not be compared with ours, as they include much more complex workload models. The former (simpler) models assume Poissonian arrival streams combined with geometrically distributed slot-usage times (the discrete-time equivalent of our negative exponentially distributed times). All queues, however, are assumed to have infinite length. All models are described and solved using a standard simulation package. In his evaluations, Pasch concentrates on mean delay, whereas we derive blocking and throughput measures. Still, Pasch derives similar trade-offs as we do, albeit at much larger cost (simulation time): he also observes the trade-off that exists between having many short slots or a smaller number of longer slots, as we observe in section 5.

Zafirovic-Vukotic and Niemgeers present very detailed analytical slotted-ring models for the Cambridge Fast Ring and Orwell [28,29]. In their models, they include higher-layer protocol aspects by modelling so-called batch arrivals. A single application-oriented packet is then, at its arrival instance, split into mini-packets that are just large enough to fit into a single slot. Although the employed models can not be compared directly, also these authors observe similar trade-offs as we (and Pasch) have observed (see e.g. [29, fig. 16]).

Finally, Ajmone Marsan et al. [1,2] present SPN-based models of multi-server systems which closely correspond to our models. However, the models presented all suffer from largeness problems. These problems are overcome by exploiting symmetries. When not exploiting symmetries, only configurations with up to 6 stations and 3 slots can be evaluated; the presented curves in [1], however, are not derived for realistic parameters. When exploiting symmetries, much larger models can be addressed; however, numerical results are not reported in [2]. The symmetries exploited by these authors correspond to the symmetries that De Goei studied and evaluated using the standard UltraSAN symmetry exploitation methods (reduced base-model construction, see [7]). Unfortunately, also De Goei does not evaluate his models for realistic settings.

6. Summary and conclusions

In this paper, we have presented an MVA approach for the solution of product-form stochastic Petri net models suitable for the analysis of slotted-ring systems. We have presented the exact product-form solution and derived the MVA recursion scheme. Apart from the MVA recursive relations, we have also indicated how to implement these relations. Our MVA differs from earlier ones in that it allows for non-disjoint S-invariants and the batch-movement of tokens. We have also compared

our model outcomes with those of a more detailed non-product-form SPN. The much more costly-to-derive results from that model are roughly the same as from our product-form model.

The advantage of the MVA approach is that it allows for much larger models to be solved than would an alternative technique based on the global-balance equations. This is partly because less detailed performance measures are derived (only mean values, no marginal probabilities) and partly because symmetries in the models are easily exploited in an MVA implementation. It should be noted, however, that one can still construct models for which also the MVA approach becomes unattractive.

Another advantage lies in the fact that the MVA approach is numerically very stable and does not suffer from overflow problems as convolution and other numerical algorithms often do. Finally, the MVA approach might allow for intuitively-appealing approximations if the number of tokens per S-invariant is very large, as suggested in [24] for queueing network models.

A question that remains is why the PF approximation does so well? This question is very difficult, if at all possible, to answer. Let us touch upon a number of issues that play a role in the accuracy of PF approximations; for a more elaborate treatment of this, we refer to the book by Van Dijk [10].

We first of all have to note that the proposed approximate model is not the only possible PF approximation. One could, for example, also consider the approximate model (starting from the SPN in fig. 1) where all the transitions $t_{i,2}$ are made timed with a very high rate, and where arcs are added from P_0 to $t_{i,1}$ and from $t_{i,1}$ back to P_0. Also this model has a PF solution; whether it is better or worse than the one we employed is left for further study. Thus, the obtained accuracy with a PF approximation depends on the amount of non-PF characteristics the original model has, that is, on the amount of "PF repair" required to change the non-PF model into a PF variant. Secondly, the extent in which the non-PF characteristics surface in the measures of interest plays a role, as well as the actual numerical values used in the model. As can be understood, all these issues are very much case dependent.

Regarding slotted-ring systems, we have observed the following. With the slot limitations per station, that is, the k_i values, we are able to "control" the admission to the medium. The choice of the slot size highly influences the blocking probabilities, and therefore the throughputs, for the various stations. Moreover, the optimum number of slots differs from station to station in asymmetrically-loaded systems.

Our product-form results can easily be used for the analysis of ATM-based networks [20] (as also suggested by one of the reviewers). With the fixed cell/slot size standardised for ATM, that is, 53 octets or 424 bits, at the standardised transmission speed of 155 Mbps [21] and the usual propagation speed $c = 2 \times 10^7$ m/s, every cell comprises a length of 54.7 m. Depending on where the ATM network is going to be used, that is, in a LAN, MAN or WAN context, different medium lengths (number of slots), number of attached stations and station workloads and access rights can quickly be evaluated with the presented MVA.

Acknowledgements

The authors would like to thank the Australian Research Council (Grant No. 69132151), Telecom Australia, and Koninklijke/Shell (The Netherlands) for supporting this research. This paper was partly written while B.R. Haverkort, at that time an assistant professor at the University of Twente, The Netherlands, visited the Teletraffic Research Centre at the University of Adelaide in the spring of 1993. The authors would finally like to thank the reviewers and editors for their recommendations and suggestions that helped in bringing this paper to its final form.

References

[1] M. Ajmone Marsan, S. Donatelli and F. Neri, GSPN models of Markovian multi-server multi-queue systems, Perform. Eval. 11(1990)227–240.

[2] M. Ajmone Marsan, S. Donatelli, F. Neri and U. Rubino, On the construction of abstract GSPNs: An exercise in modelling, *Proc. 4th Int. Workshop on Petri Nets and Performance Models* (IEEE Computer Society Press, 1991) pp. 2–17.

[3] E. Biagioni, E. Cooper and R. Sansom, Designing a practical ATM LAN, IEEE Network 7(2) (1993)32–39

[4] J.-Y. Le Boudec, The asynchronous transfer mode: A tutorial, Comp. Networks and ISDN Syst. 24(1992)279–309.

[5] G. Ciardo, J. Muppala and K.S. Trivedi, SPNP: Stochastic Petri net package, *Proc. 3rd Int. Workshop on Petri Nets and Performance Models* (IEEE Computer Society Press, 1989) pp. 142–151.

[6] G. Ciardo and K.S. Trivedi, A decomposition approach for stochastic Petri net models, *Proc. 4th Int. Workshop on Petri Nets and Performance Models* (IEEE Computer Society Press, 1991) pp. 74–83.

[7] J.A. Couvillion, R. Freire, R. Johnson, W.D. Obal, II, A. Qureshi, M. Rai, W.H. Sanders and J.E. Tvedt, Performability modelling with UltraSAN, IEEE Software (Sept. 1991) 69–80.

[8] A.J. Coyle, W. Henderson and P.G. Taylor, Reduced load approximations for loss networks, Telecom. Syst. 2(1993)21–50.

[9] A.J. Coyle, W. Henderson, C.E.M. Pearce and P.G. Taylor, A general formulation for mean-value analysis in product-form batch-movement queueing networks, Queueing Systems 16(1993) 363–372.

[10] N.M. van Dijk, *Queueing Networks and Product Forms: A Systems Approach* (Wiley, 1993).

[11] S. Donatelli and M. Sereno, On the product form solutions for stochastic Petri nets, *Proc. 13th Int. Conf. on Applications and Theory of Petri Nets*, Sheffield, UK (1992) pp. 154–172.

[12] R.M. Falconer and J.L. Adams, Orwell, a protocol for an integrated services load network, Brit. Telecom. Tech. J. 3(4)(1985)27–35.

[13] D. Frosch and K. Natarajan, Product-form solutions for closed synchronized systems of stochastic sequential processes, *Proc. 1992 Int. Computer Conf.*, Taichung, Taiwan (1992) pp. 392–402.

[14] B.R. Haverkort, Approximate performability analysis using generalized stochastic Petri nets, *Proc. 4th Int. Workshop on Petri Nets and Performance Models* (IEEE Computer Society Press, 1991) pp. 300–309.

[15] J.A.F. de Goei, A modelling survey and performance analysis of slotted-ring communication networks, M.Sc. Thesis, Department of Computer Science, University of Twente (1993).

[16] W. Henderson, D. Lucic and P.G. Taylor, A net-level performance analysis of stochastic Petri nets, J. Austral. Math. Soc., Ser. B31(1989)176–187.

[17] W. Henderson and P.G. Taylor, Embedded processes in stochastic Petri nets, IEEE Trans. Software Eng. 17(1991)108–116.

[18] W. Henderson and D. Lucic, Aggregation and disaggregation through insensitivity in stochastic Petri nets, Perform. Eval. 17(1993)91–114.
[19] A. Hopper and R.M. Needham, The Cambridge fast ring networking system, IEEE Trans. Comp. 37(1988)1214–1223.
[20] I.M. Leslie, D.R. McAuley and D.L. Tennenhouse, ATM everywhere?, IEEE Network 7(2)(1993) 40–46.
[21] R.F. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues* (Artech House, 1993).
[22] H.-L. Pasch and I.G. Niemegeers, A performance analysis of a high-speed slotted-ring access mechanism with dynamically adaprive slot sizes, *Proc. GLOBECOM '91* (IEEE Computer Society Press, 1991) pp. 1102–1109.
[23] H.-L. Pasch, Design and analysis of a load-adaptive high-speed network, Ph.D. Thesis, Department of Computer Science, University of Twente (1993).
[24] M. Reiser and S.S. Lavenberg, Mean-value analysis of closed multichain queueing networks, J. ACM 27(1980)313–322.
[25] R. Rooholamini, V. Cherkassky and M. Garver, Finding the right ATM switch for the market, IEEE Comp. 27(4)(1994)16–28.
[26] W.H. Sanders and J.F. Meyer, Reduced base model construction methods for stochastic activity networks, IEEE J. Select. Areas Commun. SAC-9(1991)25–36.
[27] S. Temple, The design of the Cambridge ring, in: *Ring Tech. Local Area Networks*, eds. I.N. Dallas and E.B. Spratt (North-Holland, 1984) pp. 79–88.
[28] M. Zafirovic-Vukotic, Performand modelling and evaluation of high speed serial interconnection structures, Ph.D. Thesis, University of Twente (1988).
[29] M. Zafirovic-Vukotic and I.G. Niemegeers, A performance modeling and evaluation of the Cambridge fast ring, IEEE Trans. Comp. 41(1992)1110–1125.