Contents

On Different Models for Packet Flow in Multistage Interconnection Networks

Martin Dietzfelbinger
Fachbereich Informatik, Lehrstuhl II, Universität Dortmund
D–44221 Dortmund, Germany
email: dietzf@ls2.informatik.uni-dortmund.de

Anna Gambin* and Sławomir Lasota,
Institute of Informatics, Warsaw University
02-097 Warsaw, Banacha 2, Poland
email: {aniag, sl}@mimuw.edu.pl

Abstract

Multistage interconnection networks (MINs) have a number of applications in many areas, for example in parallel computing systems or high-speed communication networks. In the paper we define Markov chains describing several models of packet flow through the buffered MIN with a butterfly interconnection structure and 2×2 switching elements. We develop a notation together with a mathematical framework enabling to prove certain results relating the models. Moreover, we show the ergodicity of considered Markov chains and discuss relationships between stationary distributions. The important novelty is that our approach is compositional, which allows to keep the complexity of description of a very complicated network's behaviour on a reasonable and tractable level. Considerations are mostly independent of specific network topology and routing protocol, hence we expect our method to be applicable also in other contexts for stochastic models of massively parallel systems.

1 Introduction

Interconnection networks Multistage interconnection networks (MIN) have been widely used in constructing high speed communication networks like ATM networks [19] as well as for connecting processors in parallel computing systems. Popular examples of commercial parallel computers using multistage interconnection network for communication between processors are CM-5 (Thinking Machines) and CS-2 (Meiko). Other examples of applications of interconnection networks in today's parallel computers can be found in [2, 7, 16].

MINs consist (in general) of $(k \times k)$-switching elements (*switches*), grouped into several stages (cf. Fig. 1), and are used to route messages (packets) from input switches to output ones. The MINs in which there exists precisely one path from each input to each output form the broad class of *banyan networks* [6]. For an overview of different architectures of banyan networks see [13]. In this paper we are concerned with MINs having the butterfly interconnection structure containing (2×2)-switches with input buffers of capacity one (see Fig. 1).

*Research supported by DAAD (Deutscher Akademischer Austauschdienst).

Related research Theoretical work focuses mainly on the investigation of the efficiency of interconnection networks [3]. A number of papers studied the behaviour of butterfly MINs, aiming mainly at evaluation of their performance [12, 14, 22]. In most cases it is assumed that destination addresses of packets are random (with uniform distribution), which simplifies the efficiency analysis.

A large number of results have been achieved for the simplified case (*static model*) when every input link holds exactly one message, which is sent to a randomly chosen destination [1, 21, 17]. A model which is more closely related to the practical situation is so called *dynamic (continuous) model*, in which messages are generated continuously during run-time and routed to random addresses. The most frequently evaluated performance measure in this model is *throughput* (expected number of packets reaching their destinations per time step). Pure analytical results can be (up to now) found only for the *unbuffered* MIN [13, 15]. An analysis of the throughput in this case can take benefit from the fact that consecutive time-steps are independent from each other, because the network has no memory. The situation is much more complex in the case of *buffered networks*. The consecutive steps are not independent from each other, and compositional descriptions of the network's behaviour are in general not markovian, hence difficult to analyze. This is the reason why known approaches to buffered networks assume some independences between different parts of the network (single buffers [9], switches [20] or small groups of switches), which makes the analyses much easier. Another possible simplification is to introduce *relaxations*: some packets are deleted when they can not be routed immediately [12, 18]. This resolves the bottom-up dependencies between consecutive stages and enables an approximation of throughput by recursive formulas.

Results In this paper we present a formal description of the continuous routing process in the butterfly MIN without any simplifying assumptions or relaxations, i.e. in so called *full blocking model*. As far as we know, no such definitions have been proposed by now. We consider here butterfly MINs with (2×2)-switches having buffers of capacity one (for simplicity called MINs in the sequel), but all the results can be easily repeated for $(k \times k)$-switches $(k \geq 3)$ and/or buffers of greater capacity. The main motivation of the work is to formalize the descriptions of packet flow appearing in the literature, as they explicitly or implicitly admit several variations [20, 22]. We describe the network by means of Markov chain, since its global behaviour is history-independent. Our approach is *compositional*, i.e. we built the description of the whole network from descriptions of its small components. This is in contrast to former approaches, where behaviour of MIN was modeled by description of only small parts of it. As it was mentioned, in the full blocking model it is not sufficient to investigate only small pieces of the network. Instead, we give a systematic way of composing descriptions of single switches into a Markov chain describing the continuous routing process. An important advantage of the compositional method is that it is possible to keep the complexity of the presentation and reasoning about the Markov chains on a reasonable level.

We distinguish three different models of packet flow, being formalizations of the generic full blocking model, differing in a degree of parallelism of switches' actions. We found some links between the models, which show that they are closely related. In the paper some mathematical tools are developed, applied then in defining the aforementioned models of packet flow. Moreover, these tools are very useful in comparing different models, as they support compositional approach: we can restrict ourselves to analyzing only small parts of the network instead of the whole one. The proposed method is quite general, independent of specific network topology and protocol of packets routing. This means that usefulness of our approach is not bounded only to description of butterfly MINs: we expect it to be applicable also to other issues dealing with probabilistic description of dynamic behaviour of parallel systems, for instance to stochastic neuronal networks. As an example, we apply our notation to *relaxed blocking model* [18]. Studying relaxed blocking and full blocking in the common framework is promising in the context of further work on estimating throughput (cf. Section 10).

We investigate carefully properties of defined Markov chains, and show that all of them are ergodic. This justifies a steady state (equilibrium state) analysis of MIN's performance, by now tacitly assumed to be correct in many papers. Moreover, as corollaries from relating different models we obtain some links between stationary distributions of the considered Markov chains.

Organization of the paper The paper is organized as follows. Informal description of different models of continuous routing is given in Section 2. In Section 3 we define three Markov chains corresponding to the models mentioned above. We start by describing the local behaviour of one switch (in Section 3.1), then (in Section 3.2) we define the global-state Markov chains. In Section 4 we analogously deal with the relaxed blocking model. Section 5 is devoted to proving ergodicity of the Markov chains. Section 6 contains some auxiliary definitions and lemmas, useful in Sections 7, 8 and 9, in which we relate different models for full blocking as well as for relaxed blocking. We end with conclusions and a sketch of some directions for further work in the area in Section 10. Sections 3, 6 and 7 are extended version of [5].

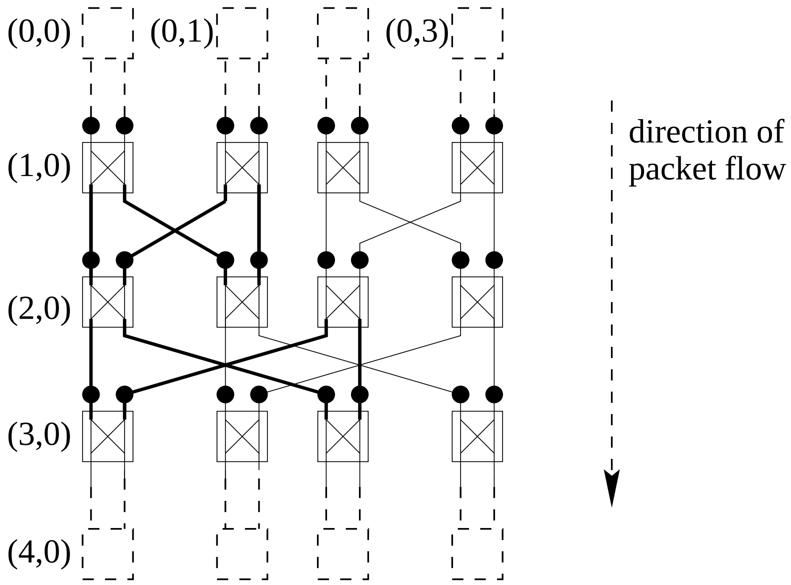

Figure 1: Butterfly network with three stages ($H = 3$); buffers are denoted by black circles. The border stages, depicted with dashed line, are introduced in Section 3 to represent an interaction of the network with the environment, i. e. input and flow-out of packets.

2 Informal description of models

Models of a packet flow MINs are used to route packets appearing in the switches at the first stage (input switches or input links) to appropriate output switches (addresses) at the last stage (see

Fig. 1). There is exactly one path from every input switch to every output switch, and each packet visits all the stages along its way through the network. We assume, following many related papers ([13, 20]), that an address of a packet is random (it is usually done to make a performance analysis possible) – this means that a packet after entering into each switch chooses its direction randomly. The computation of the network proceeds in synchronized steps more or less simultaneously by all the switches. The degree of parallelism varies in different models, which are described in this section.

One step consists of the flow of packets between every pair of neighbour stages. Locally, in one switch, such a step consists of two half-steps, (how they are related in time depends on a chosen model of packet flow):

1) *Flow-out phase*: some packets leave the switch (after resolving conflicts), but only if there is a place in the appropriate buffer of the switch at the next stage.

2) *Input phase*: packets from the previous stage enter to the empty buffers of the switch.

The *ripple model* is the most frequently used in the literature [14, 20, 22]. It has the property that the flow-out phase at the stage happens after the flow-out phase at the next stage has been completed. We can effectively simulate such a model (switches need not to wait) by means of a closely related *odd-even model* (which can be thought of as a pipelining implementation of the ripple model): the flow-out phases occur simultaneously at the odd stages (while even stages are idle), followed by input phases at the even stages, then in every consecutive step the odd and even stages change their roles. Another natural possibility is offered by the *simultaneous model*, in which the flow-out phase occurs simultaneously at all the stages (similarly input phase which follows flow-out one). In this model switches are never idle.

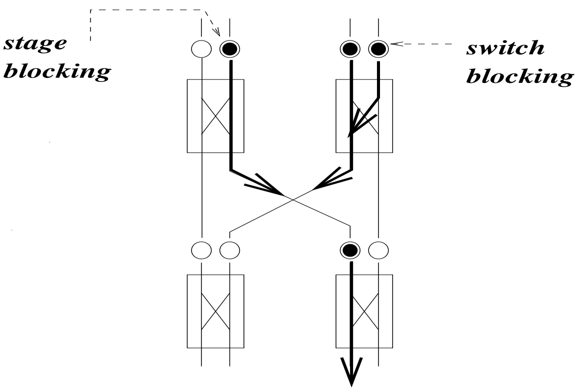

Figure 2: Switch blocking and stage blocking.

Full blocking versus relaxed blocking We distinguish between two cases of message blocking each other (cf. Fig. 2). *Switch blocking* occurs when both messages in the input buffers of a switch have to be routed to the same output link. In this case only one of these two messages can be routed in a current time step, the other has to stay in its buffer. The choice (conflict resolution) is made randomly during the flow-out phase. *Stage blocking* occurs if a buffer at the consecutive stage is occupied, disallowing another packet from the previous stage to be routed. This introduces bottom-up dependencies between the switches of the network. To simplify the analysis, so called *relaxed blocking* *model* (called here *relaxed model* in short) was proposed [18], where stage blocking is ignored: a message is deleted, if the receiving buffer is occupied.

In the previous paragraph we described three different models of packet flow, being variations of the full blocking model. In the sequel we also consider similar variations of relaxed model.

3 Full blocking

Throughout the rest of the paper let H denote the number of stages of the butterfly network under consideration. The structure of the network can be represented by the graph $G = (I, E)$, where the sets of nodes and edges are defined as follows:

$$I = \{1, 2, \ldots, H\} \times \{0, 1, \ldots, W\}, \qquad W = 2^{H-1} - 1,$$

$$E = \{((r, c), (r+1, c')) : 1 \leq r \leq H-1, \quad 0 \leq c \leq W, \quad c' = c \text{ or } c' = c^{(r)}\}$$

where $c^{(r)}$ is the unique number in $[0, W]$ that differs from c only in the rth bit, $0 \leq c \leq W$, $1 \leq r \leq H-1$. Nodes (r, c) and $(r, c^{(r)})$ have common neighbours $(r+1, c)$ and $(r+1, c^{(r)})$ at the $(r+1)$st stage.

In further definitions we will need the notion of c-map, which is a slight generalization of the notion of finite Markov chain:

Definition 3.1 *For finite sets A,B, a* **c-map** $P : A \rightsquigarrow B$ *is a function* $P : A \to [0,1]^B$ *such that for every* $a \in A$, $P(a)$ *is a probability distribution, i.e.* $\sum_{b \in B} P(a)(b) = 1$. *The sets A and B will be called* **the domain** *and* **the co-domain** *of P, respectively. We say that P is* **deterministic** *if for each a, there exists some b with* $P(a)(b) = 1$.

Note A concept of c-map coincides with the notion of the conditional probability distribution; a c-map $\{*\} \rightsquigarrow A$ from a singleton set can be seen as an unconditional probability distribution on A. When $A = B$, $P : A \rightsquigarrow A$ corresponds to the transition matrix of a finite Markov chain.

3.1 Local description of one switch

In this section we define c-maps describing the behaviour of one switch. A single switching element has two input ports (each equipped with one-element buffers) and two output ones, with each input port connected to each output one (see Fig. 1). Let the set $B = \{\circ, \downarrow, \searrow\}$ represent possible states of one buffer in the switch. Element "$\circ$" denotes an empty buffer, and the two other elements of B represent the two possible directions of a packet residing in the buffer ("$\downarrow$" denotes the straight direction, "$\searrow$" denotes the crossing direction, see Fig. 1). To improve readability, we introduce a redundant notation: "$\swarrow$"="$\searrow$". Let $S = B \times B$ denote the set of all possible states of (both buffers of) a switch.

As a single switching element cannot be described by a Markov chain $S \rightsquigarrow S$ (because its behaviour depends on states of its neighbours), we encode the behaviour of it by means of two c-maps, describing first and second half of the step, respectively:

$$\sigma : S \times \{0,1\}^2 \rightsquigarrow S \times \{0,1\}^2, \qquad \tau : S \times \{0,1\}^2 \rightsquigarrow S.$$

We define these c-maps in Tables 1 and 2.

The domain of σ consists of pairs $(s, (b_1, b_2))$, where s denotes the state of the switch before a step and (b_1, b_2) describes the contents of the two buffers of the neighbour switches at the next stage: $b_1 = 0$

initial S	IN:$(1,1)$		IN:$(0,1)$		IN:$(1,0)$		IN:$(0,0)$	
	new S	out	new S	out	new S	out	new S	out
$\circ\ \circ$	$\circ\ \circ$	$(0,0)$	$\circ\ \circ$	$(0,0)$	$\circ\ \circ$	$(0,0)$	$\circ\ \circ$	$(0,0)$
$\downarrow\circ$	$\downarrow\circ$	$(0,0)$	$\circ\ \circ$	$(1,0)$	$\downarrow\circ$	$(0,0)$	$\circ\ \circ$	$(1,0)$
$\searrow\circ$	$\searrow\circ$	$(0,0)$	$\searrow\circ$	$(0,0)$	$\circ\ \circ$	$(0,1)$	$\circ\ \circ$	$(0,1)$
$\circ\downarrow$	$\circ\downarrow$	$(0,0)$	$\circ\downarrow$	$(0,0)$	$\circ\ \circ$	$(0,1)$	$\circ\ \circ$	$(0,1)$
$\circ\swarrow$	$\circ\swarrow$	$(0,0)$	$\circ\ \circ$	$(1,0)$	$\circ\swarrow$	$(0,0)$	$\circ\ \circ$	$(1,0)$
$\downarrow\swarrow$	$\downarrow\swarrow$	$(0,0)$	$\frac{1}{2}\downarrow\circ$ $\frac{1}{2}\circ\swarrow$	$(1,0)$	$\downarrow\swarrow$	$(0,0)$	$\frac{1}{2}\downarrow\circ$ $\frac{1}{2}\circ\swarrow$	$(1,0)$
$\searrow\downarrow$	$\searrow\downarrow$	$(0,0)$	$\searrow\downarrow$	$(0,0)$	$\frac{1}{2}\circ\downarrow$ $\frac{1}{2}\searrow\circ$	$(0,1)$	$\frac{1}{2}\circ\downarrow$ $\frac{1}{2}\searrow\circ$	$(0,1)$
$\searrow\!\!\!\!\swarrow$	$\searrow\!\!\!\!\swarrow$	$(0,0)$	$\searrow\circ$	$(1,0)$	$\circ\swarrow$	$(0,1)$	$\circ\ \circ$	$(1,1)$
$\downarrow\downarrow$	$\downarrow\downarrow$	$(0,0)$	$\circ\downarrow$	$(1,0)$	$\downarrow\circ$	$(0,1)$	$\circ\ \circ$	$(1,1)$

Table 1: Definition of σ: the first half of the step (flow out of packets from a switch). For each state (in the left-most column) and content of buffers of consecutive neighbour nodes (in the top-most row) there is a new state (together with a probability) and output of packets. Note that output of packets turns out to be determined (probability 1).

$(b_2 = 0)$ means that the buffer is empty, otherwise the buffer is occupied by a packet. The co-domain of σ consists of pairs $(s,(o_1,o_2))$, where s denotes the state of the switch after the flow out of packets and a pair (o_1,o_2) describes how many packets flow out. Similarly, $o_1 = 1$ $(o_2 = 1)$ means that a packet flows out. Hence $\sigma(s,(b_1,b_2))$ "returns" an output of packets from a switch together with a new state of that switch, given an initial state s and (b_1,b_2) describing the contents of the two buffers of neighbours at the next stage.

Similarly, the domain of τ consists of pairs $(s,(i_1,i_2))$, where s is a state before an input of new packets (i.e. after σ), and a pair (i_1,i_2) describes how many packets enter the switch. $\tau(s,(i_1,i_2))$ describes a new state after packets have entered the switch in state s. Our description of one switch is in the spirit of the probabilistic automaton from [18].

For convenience let us introduce sets $I_r = \{(r,c) : 0 \le c \le W\}$, for $0 \le r \le H+1$. The set I_r represents all nodes at the rth stage. Stages 0 and $H+1$ are auxiliary and are introduced only for technical convenience: the stage 0 intuitively represents input of packets into the network and symmetrically stage $(H+1)$ receives packets which flow out from the network.

In the next section we will define Markov chains $V \rightsquigarrow V$, describing one step of the computation of the network, where $V = S^{I \cup I_{H+1}}$ is the set of states, $I = \bigcup_{1 \le r \le H} I_r$. Elements of V represent all configurations (states) of the network, i.e. all possible states of all the switches. We will also need a set V' of states in a half-step, defined as $V' = (S \times \{0,1\}^2)^{I_0 \cup I \cup I_{H+1}}$. Elements of V' describe the contents of all switches together with the packets which flowed out in the first half-step (and are placed on the links between stages).

For further definitions we need two families of auxiliary functions:

$$IN^{r,c} : V \to \{0,1\}^2, \quad \text{for} \quad (r,c) \in I \cup I_0,$$

$$in^{r,c} : V' \to \{0,1\}^2, \quad \text{for} \quad (r,c) \in I \cup I_{H+1}.$$

$IN^{r,c}(v)$ returns information about the contention of buffers of neighbours at the next stage for a switch at node (r,c) (the pair (b_1,b_2) above); similarly $in^{r,c}(v')$ returns information about packets coming into

	in:(0, 0)	in:(1, 0)	in:(0, 1)	in:(1, 1)
initial S	new S	new S	new S	new S
$\circ\ \circ$	$\circ\ \circ$	$\frac{1}{2}\downarrow\circ,\ \frac{1}{2}\searrow\circ$	$\frac{1}{2}\circ\downarrow,\ \frac{1}{2}\circ\swarrow$	$\frac{1}{4}\downarrow\swarrow,\ \frac{1}{4}\searrow\downarrow$ $\frac{1}{4}\searrow\!\!\!\!\swarrow,\ \frac{1}{4}\downarrow\downarrow$
$\downarrow\circ$	$\downarrow\circ$		$\frac{1}{2}\downarrow\swarrow,\ \frac{1}{2}\downarrow\downarrow$	
$\searrow\circ$	$\searrow\circ$		$\frac{1}{2}\searrow\downarrow,\ \frac{1}{2}\searrow\!\!\!\!\swarrow$	
$\circ\downarrow$	$\circ\downarrow$	$\frac{1}{2}\searrow\downarrow,\ \frac{1}{2}\downarrow\downarrow$		
$\circ\swarrow$	$\circ\swarrow$	$\frac{1}{2}\downarrow\swarrow,\ \frac{1}{2}\searrow\!\!\!\!\swarrow$		
$\downarrow\swarrow$	$\downarrow\swarrow$			
$\searrow\downarrow$	$\searrow\downarrow$			
$\searrow\!\!\!\!\swarrow$	$\searrow\!\!\!\!\swarrow$			
$\downarrow\downarrow$	$\downarrow\downarrow$			

Table 2: Definition of τ: the second half of the step: (input of packets). Empty fields could be here arbitrarily filled, as they correspond to situations which can never occur, although we will need to fill them in Section 4 to express relaxation.

the switch ((i_1, i_2) above). In the definition of these functions we are "looking through" the edges of graph G describing the structure of the network (cf. Fig 1):

$$IN^{r,c}(v) = \begin{cases} (occ(s{\downarrow_1}), occ(\bar{s}{\downarrow_1})) & \text{if } c < c^{(r)}, \\ (occ(\bar{s}{\downarrow_2}), occ(s{\downarrow_2})) & \text{otherwise,} \end{cases} \qquad \text{for } 0 < r < H;$$

where $s = v(r+1, c) \in B^2$, $\bar{s} = v(r+1, c^{(r)}) \in B^2$ and $(a,b){\downarrow_1} = a, (a,b){\downarrow_2} = b$ denote the usual projections. The auxiliary function $occ : B \to \{0,1\}$ tells us whether the buffer is occupied; $occ(\circ) = 0,\ occ(\downarrow) = occ(\searrow) = 1$.

$$in^{r,c}(v') = \begin{cases} (o{\downarrow_1}, \bar{o}{\downarrow_1}) & \text{if } c < c^{((r-1))}, \\ (\bar{o}{\downarrow_2}, o{\downarrow_2}) & \text{otherwise,} \end{cases} \qquad \text{for } 1 < r < H+1;$$

where $o = v'(r-1, c){\downarrow_2} \in \{0,1\}^2$, $\bar{o} = v'(r-1, c^{((r-1))}){\downarrow_2} \in \{0,1\}^2$. The definitions of $IN^{0,c}$,$in^{1,c}$, $IN^{H,c}$ and $in^{H+1,c}$ are slightly different, as they describe the "border" of the network:

$$IN^{r,c}(v) = (occ(v(r+1,c){\downarrow_1}), occ(v(r+1,c){\downarrow_2})), \quad \text{for } r = 0, H;$$

$$in^{r+1,c}(v') = v'(r,c){\downarrow_2} \in \{0,1\}^2, \quad \text{for } r = 0, H.$$

At this point we are prepared to introduce four families of c-maps $\sigma^{r,c}$, $\tau^{r,c}$, $\pi^{r,c}$, $\omega^{r,c}$, indexed by $(r,c) \in I$, which will be the atomic elements of the full Markov chains in the next section. The c-maps

$$\sigma^{r,c} : V \leadsto S \times \{0,1\}^2, \qquad \tau^{r,c} : V' \leadsto S$$

apply σ,τ to the switch at node (r,c):

$$\sigma^{r,c}(v)(s,o) = \sigma(v(r,c), IN^{r,c}(v))(s,o), \qquad \tau^{r,c}(v')(s) = \tau(v'(r,c){\downarrow_1}, in^{r,c}(v'))(s)$$

On the other hand $\pi^{r,c} : V \leadsto S \times \{0,1\}^2$, $\quad \omega^{r,c} : V' \leadsto S$ describe an idle phase of a switch at node (r,c) and are defined by:

$$\pi^{r,c}(v)(v(r,c),(0,0)) = 1, \qquad \omega^{r,c}(v')(v'(r,c){\downarrow_1}) = 1.$$

Note It is probably surprising that the domains of $\sigma^{r,c}, \tau^{r,c}, \pi^{r,c}$ and $\omega^{r,c}$ are so "large", but observe that these c-maps examine only a little part of their domains. For example $\sigma^{r,c}$ needs only its state (i.e. the state of the switch at (r,c)) and the state of some neighbours, and is "independent"[1] from the rest of the network. Such a blowup of domains is unavoidable in our setting, as a parallel composition of c-maps as defined in Section 3.2 works only for c-maps with equal domains.

To unify our notation we will also need some "border" c-maps describing input and output of packets into/from the network:

$$\sigma^{0,c}, \pi^{0,c}, \sigma^{H+1,c}, \pi^{H+1,c} : V \rightsquigarrow S \times \{0,1\}^2,$$

$$\tau^{H+1,c}, \omega^{H+1,c} : V' \rightsquigarrow S$$

defined as follows:

$$\sigma^{0,c}(v)(s,o) = \sigma((\downarrow,\downarrow), IN^{0,c}(v))(s,o),$$

$$\pi^{0,c}(v)((\downarrow,\downarrow),(0,0)) = 1,$$

$$\sigma^{H+1,c}(v)((\circ,\circ),(0,0)) = 1,$$

$$\pi^{H+1,c}(v)(v(H+1,c),(0,0)) = 1.$$

The c-maps $\tau^{H+1,c}$, $\omega^{H+1,c}$ are defined analogously to the corresponding c-maps for other stages:

$$\tau^{H+1,c}(v')(s) = \tau(v'(H+1,c)\downarrow_1, in^{H+1,c}(v'))(s),$$

$$\omega^{H+1,c}(v')(v'(H+1,c)\downarrow_1) = 1.$$

Remark In the above definitions of $\sigma^{r,c}, \pi^{r,c}, \tau^{r,c}, \omega^{r,c}$ we intentionally omitted some detail. The co-domains of $\sigma^{r,c}, \pi^{r,c}$ should be actually $(S \times \{0,1\}^2)^{\{(r,c)\}}$, i.e. a set of functions from a singleton set $\{(r,c)\}$; moreover, co-domains of $\tau^{r,c}, \omega^{r,c}$ should be $S^{\{(r,c)\}}$. In the next sections we assume that it actually is the case, as such co-domains are appropriate while composing these c-maps into more complex ones.

3.2 Global behaviour of the network

In this section we define three Markov chains Γ_{ripple}, Γ_{sim}, $\Gamma_{\text{odd-even}} : V \rightsquigarrow V$, which correspond to the described models of packet flow. The Markov chains will be composed of c-maps $\sigma^{r,c}$, $\tau^{r,c}$, $\pi^{r,c}$, $\omega^{r,c}$ as defined in the previous section by means of the following two operations on c-maps:

Definition 3.2 *For $P : A \rightsquigarrow B$ and $Q : B \rightsquigarrow C$, we define the* **sequential composition of** P **and** Q*, denoted by $P;Q$, as a c-map $A \rightsquigarrow C$ determined by the following formula:* $(P;Q)(a,c) = \sum_{b \in B} P(a,b) \cdot Q(b,c)$.
For $P : A \rightsquigarrow B$ and $Q : A \rightsquigarrow C$, the **parallel composition of** P **and** Q, $P \parallel Q : A \rightsquigarrow B \times C$, *is defined by* $(P \parallel Q)(a)(b,c) = P(a)(b) \cdot Q(a)(c)$.

Essentially ";" corresponds to matrix multiplication, while "$\|$" gives a family of product distributions. Both operations are associative and parallel composition is also commutative. While defining them we were inspired by the operators of CSP [8] – for example "$\|$" enforces component c-maps to "proceed" simultaneously like the operator of synchronized parallel composition in CSP.

[1]The formalization of the notion of independence is given in Section 6.

Notation In the following "$\|$" has higher priority than ";". We sometimes write $\binom{P}{Q}$ instead of $(P;Q)$ for the sake of simplicity (this allows us to better recognize c-maps concerned with certain stages of the network).

Remark Throughout the rest of the paper we deliberately ignore the difference between sets $A^{B\cup C}$ and $A^B \times A^C$; moreover we identify the sets A and $(A \times \{*\})$, where $\{*\}$ is an arbitrary singleton set. Due to such an identification "up to obvious bijection" we avoid some inessential technicalities.

Both operations of composition have identities; for any set A, we define c-maps $Id^{;}_A : A \rightsquigarrow A$ and $Id^{\|}_A : A \rightsquigarrow \{*\}$ as follows:

$$Id^{;}_A(a)(a) = 1, \quad Id^{\|}_A(a)(*) = 1, \quad \text{for any} \quad a \in A.$$

Proposition 3.3 *The families of c-maps $Id^{;}_A$ and $Id^{\|}_A$, are identities for ; and $\|$, respectively, i.e. for any c-map $P : B \rightsquigarrow C$:*

$$Id^{;}_B ; P = P = P ; Id^{;}_C,$$

$$Id^{\|}_B \parallel P = P.$$

The domain V can be split into subparts corresponding to the stages as follows:

$$V = V_1 \times V_2 \times \cdots \times V_{H+1},$$

where $V_r = S^{I_r}$. Analogously $V' = V'_0 \times V'_1 \times \cdots \times V'_{H+1}$, where $V'_r = (S \times \{0,1\}^2)^{I_r}$. Before defining the Markov chains, let us group together the c-maps $\sigma^{r,c}$, $\pi^{r,c}$ belonging to the rth stage into $\sigma^r, \pi^r : V \rightsquigarrow V'_r$:

$$\sigma^r = \|_{0\le c\le W}\ \sigma^{r,c}, \quad \pi^r = \|_{0\le c\le W}\ \pi^{r,c}, \quad \text{for } r = 0, 1, \ldots, H+1.$$

Analogously we define τ^r,$\omega^r : V' \rightsquigarrow V_r$ as a parallel composition of appropriate components:

$$\tau^r = \|_{0\le c\le W}\ \tau^{r,c}, \quad \omega^r = \|_{0\le c\le W}\ \omega^{r,c}, \quad \text{for } r = 1, 2, \ldots H+1.$$

Finally, as the result of hard work while preparing notation, we are able to define the aforementioned three Markov chains as follows:

$$\begin{array}{rl} \Gamma_{\text{sim}} = & (\boldsymbol{\sigma}^{\mathbf{0}} \| \boldsymbol{\sigma}^{\mathbf{1}} \| \boldsymbol{\sigma}^{\mathbf{2}} \| \cdots \| \boldsymbol{\sigma}^{\boldsymbol{H-1}} \| \boldsymbol{\sigma}^{\boldsymbol{H}} \| \boldsymbol{\sigma}^{\boldsymbol{H+1}}); \\ & (\boldsymbol{\tau}^{\mathbf{1}} \| \boldsymbol{\tau}^{\mathbf{2}} \| \cdots \| \boldsymbol{\tau}^{\boldsymbol{H-1}} \| \boldsymbol{\tau}^{\boldsymbol{H}} \| \boldsymbol{\tau}^{\boldsymbol{H+1}}). \end{array}$$

$$\begin{array}{rl} \Gamma_{\text{ripple}} = & (\pi^0 \| \pi^1 \| \pi^2 \| \cdots \| \pi^{H-1} \| \pi^H \| \boldsymbol{\sigma}^{\boldsymbol{H+1}}); \\ & (\omega^1 \| \omega^2 \| \cdots \| \omega^{H-1} \| \omega^H \| \omega^{H+1}); \\ & (\pi^0 \| \pi^1 \| \pi^2 \| \cdots \| \pi^{H-1} \| \boldsymbol{\sigma}^{\boldsymbol{H}} \| \pi^{H+1}); \\ & (\omega^1 \| \omega^2 \| \cdots \| \omega^{H-1} \| \omega^H \| \boldsymbol{\tau}^{\boldsymbol{H+1}}); \\ & (\pi^0 \| \pi^1 \| \pi^2 \| \cdots \| \boldsymbol{\sigma}^{\boldsymbol{H-1}} \| \pi^H \| \pi^{H+1}); \\ & (\omega^1 \| \omega^2 \| \cdots \| \omega^{H-1} \| \boldsymbol{\tau}^{\boldsymbol{H}} \| \omega^{H+1}); \\ & \cdots \\ & (\boldsymbol{\sigma}^{\mathbf{0}} \| \pi^1 \| \pi^2 \| \cdots \| \pi^{H-1} \| \pi^H \| \pi^{H+1}); \\ & (\boldsymbol{\tau}^{\mathbf{1}} \| \omega^2 \| \cdots \| \omega^{H-1} \| \omega^H \| \omega^{H+1}). \end{array}$$

$$
\begin{aligned}
\Gamma_{\text{odd-even}} = \; & (\boldsymbol{\sigma^0} \| \, \pi^1 \, \| \, \boldsymbol{\sigma^2} \| \, \pi^3 \, \| \cdots \; \| \, \boldsymbol{\sigma^{H-2}} \| \, \pi^{H-1} \, \| \, \boldsymbol{\sigma^H} \| \, \pi^{H+1} \,); \\
& (\boldsymbol{\tau^1} \, \| \, \omega^2 \, \| \, \boldsymbol{\tau^3} \| \cdots \; \| \, \omega^{H-2} \, \| \, \boldsymbol{\tau^{H-1}} \| \, \omega^H \, \| \, \boldsymbol{\tau^{H+1}}); \\
& (\pi^0 \, \| \, \boldsymbol{\sigma^1} \| \, \pi^2 \, \| \, \boldsymbol{\sigma^3} \| \cdots \; \| \, \pi^{H-2} \, \| \, \boldsymbol{\sigma^{H-1}} \| \, \pi^H \, \| \, \boldsymbol{\sigma^{H+1}}); \\
& (\omega^1 \, \| \, \boldsymbol{\tau^2} \| \, \omega^3 \, \| \cdots \; \| \, \boldsymbol{\tau^{H-2}} \| \, \omega^{H-1} \, \| \, \boldsymbol{\tau^H} \| \, \omega^{H+1} \,).
\end{aligned}
$$

Note that border stages proceed in accordance with other stages in each model. Let $\Gamma_1 : V \rightsquigarrow V$ denote the c-map consisting of two first lines of the definition of $\Gamma_{\text{odd-even}}$, $\Gamma_2^1 : V \rightsquigarrow V'$ denote the third line, $\Gamma_2^2 : V' \rightsquigarrow V$ denote the fourth, and let $\Gamma_2 = \Gamma_2^1 ; \Gamma_2^2$.

For the definition of $\Gamma_{\text{odd-even}}$ we assumed H to be even. When H is odd one should slightly modify the definition above as follows:

$$
\begin{aligned}
\Gamma_{\text{odd-even}} = & (\pi^0 \, \| \, \boldsymbol{\sigma^1} \| \, \pi^2 \, \| \, \boldsymbol{\sigma^3} \| \cdots \; \| \, \boldsymbol{\sigma^{H-2}} \| \, \pi^{H-1} \, \| \, \boldsymbol{\sigma^H} \| \, \pi^{H+1} \,); \\
& (\omega^1 \, \| \, \boldsymbol{\tau^2} \| \, \omega^3 \, \| \cdots \; \| \, \omega^{H-2} \, \| \, \boldsymbol{\tau^{H-1}} \| \, \omega^H \, \| \, \boldsymbol{\tau^{H+1}}); \\
& (\boldsymbol{\sigma^0} \| \, \pi^1 \, \| \, \boldsymbol{\sigma^2} \| \, \pi^3 \, \| \cdots \; \| \, \pi^{H-2} \, \| \, \boldsymbol{\sigma^{H-1}} \| \, \pi^H \, \| \, \boldsymbol{\sigma^{H+1}}); \\
& (\boldsymbol{\tau^1} \, \| \, \omega^2 \, \| \, \boldsymbol{\tau^3} \| \cdots \; \| \, \boldsymbol{\tau^{H-2}} \| \, \omega^{H-1} \, \| \, \boldsymbol{\tau^H} \| \, \omega^{H+1} \,).
\end{aligned}
$$

4 Relaxed blocking

In recent work by R. Diekmann et al. [18], in order to analyze the throughput of MINs, so called relaxed blocking model was proposed, allowing packets to disappear when stage blocking occurs (for detailed description of relaxation see Section 2). In this section we formally describe this phenomenon by using the same approach as in previous sections. It turns out that only slight modification of already defined Markov chains Γ_{ripple}, $\Gamma_{\text{odd-even}}$ and Γ_{sim} are necessary. In sections 5 and 9 we will return once more to relaxed models to study the existence of stationary distribution as well as the relationships between several rules of packet flow.

We start as in Section 3 by definition of "atomic components"

$$\sigma_* : S \times \rightsquigarrow S \times \{0,1\}^2, \qquad \tau_* : S \times \{0,1\}^2 \rightsquigarrow S$$

from which we build the description of the whole network. C-map σ_* is obtained from σ by applying the latter always to input $(0,0)$:

$$\sigma_*(s) = \sigma(s, (0,0)), \quad \text{for any} \quad s \in S.$$

This means that σ_* "ignores" the information about switches in the next stage, and always sends as many packets as possible.

The c-map τ_* remains almost the same as τ from Section 3.1; we only fill empty fields of Table 2 (corresponding there to impossible inputs for τ) to express the relaxation of packets entering into the switch. The full definition of τ_* can be found in Table 3.

Now, having already defined σ_* and τ_* we can define new families of c-maps $\sigma_*^{r,c}$, $\tau_*^{r,c}$ for $(r,c) \in I_0 \cup I \cup I_{H_1}$, as it was done in Section 3.1, with the only difference that functions $IN^{r,c}$ need no more be used. We similarly group switches in the same stage to obtain:

$$\sigma_*^r = \|_{0 \le c \le W} \; \sigma_*^{r,c}, \qquad \tau_*^r = \|_{0 \le c \le W} \; \tau_*^{r,c}.$$

Three Markov chains for the relaxed model, Γ^*_{ripple}, $\Gamma^*_{\text{odd-even}}$ and Γ^*_{sim} are defined precisely as corresponding Markov chains in Section 3.2, with σ^r replaced by σ_*^r and τ^r replaced by τ_*^r, for any $r = 0, 1, \ldots, (H+1)$. Note that the same c-maps for idle phases are used here.

initial S	in:$(0,0)$ new S	in:$(1,0)$ new S	in:$(0,1)$ new S	in:$(1,1)$ new S
$\circ\ \circ$	$\circ\ \circ$	$\frac{1}{2}\downarrow\circ,\ \frac{1}{2}\searrow\circ$	$\frac{1}{2}\circ\downarrow,\ \frac{1}{2}\circ\swarrow$	$\frac{1}{4}\downarrow\swarrow,\ \frac{1}{4}\searrow\downarrow$ $\frac{1}{4}\swarrow\!\!\!\!\!\searrow,\ \frac{1}{4}\downarrow\downarrow$
$\downarrow\circ$	$\downarrow\circ$	$\downarrow\circ$	$\frac{1}{2}\downarrow\swarrow,\ \frac{1}{2}\downarrow\downarrow$	$\frac{1}{2}\downarrow\swarrow,\ \frac{1}{2}\downarrow\downarrow$
$\searrow\circ$	$\searrow\circ$	$\searrow\circ$	$\frac{1}{2}\searrow\downarrow,\ \frac{1}{2}\swarrow\!\!\!\!\!\searrow$	$\frac{1}{2}\searrow\downarrow,\ \frac{1}{2}\swarrow\!\!\!\!\!\searrow$
$\circ\downarrow$	$\circ\downarrow$	$\frac{1}{2}\searrow\downarrow,\ \frac{1}{2}\downarrow\downarrow$	$\circ\downarrow$	$\frac{1}{2}\searrow\downarrow,\ \frac{1}{2}\downarrow\downarrow$
$\circ\swarrow$	$\circ\swarrow$	$\frac{1}{2}\downarrow\swarrow,\ \frac{1}{2}\swarrow\!\!\!\!\!\searrow$	$\circ\swarrow$	$\frac{1}{2}\downarrow\swarrow,\ \frac{1}{2}\swarrow\!\!\!\!\!\searrow$
$\downarrow\swarrow$	$\downarrow\swarrow$	$\downarrow\swarrow$	$\downarrow\swarrow$	$\downarrow\swarrow$
$\searrow\downarrow$	$\searrow\downarrow$	$\searrow\downarrow$	$\searrow\downarrow$	$\searrow\downarrow$
$\swarrow\!\!\!\!\!\searrow$	$\swarrow\!\!\!\!\!\searrow$	$\swarrow\!\!\!\!\!\searrow$	$\swarrow\!\!\!\!\!\searrow$	$\swarrow\!\!\!\!\!\searrow$
$\downarrow\downarrow$	$\downarrow\downarrow$	$\downarrow\downarrow$	$\downarrow\downarrow$	$\downarrow\downarrow$

Table 3: Definition of τ_*.

5 Stationary distributions

In this section we analyze the Markov chains for all the models. We prove that Γ_{ripple}, $\Gamma_{\text{odd-even}}$ and Γ_{sim} have stationary distributions. Moreover the Markov chain for Γ_{sim} is periodic with period 2. At the beginning some basic facts from the theory of finite Markov chains are recalled. (As a reference we suggest [4]).

By **the underlying graph** of a finite Markov chain $\mathcal{M} : U \rightsquigarrow U$ we mean the directed graph $\mathcal{G} = (U, E)$, where $(u_1, u_2) \in E$ if and only if $\mathcal{M}(u_1)(u_2) \neq 0$. We say that a probability distribution p on U (which can be seen as a c-map from a singleton set $p : \{*\} \rightsquigarrow U$) is **stationary** for $\mathcal{M}$ if $p ; \mathcal{M} = p$.

Definition 5.1 *A* **strong component** *of a directed graph $\mathcal{G}$ is a maximal subgraph of $\mathcal{G}$ such that for any two of its vertices u_1, u_2 there exists a path from u_1 to u_2. A (strong) component is an* **absorbing (strong) component** *if it has no outgoing edges.*

Definition 5.2 *A* **normal state** *of a finite Markov chain $\mathcal{M}$ is any state $u \in U$ reachable from all other states in the underlying graph of $\mathcal{M}$.*

The following facts are classical results in Markov chain theory.

Fact 5.3 *Let $\mathcal{M}$ be a finite Markov chain and $\mathcal{G}$ its underlying graph. If u is a normal state then $\mathcal{G}$ has exactly one absorbing strong component, which consists of all states that are reachable from u.*

Fact 5.4 *Let $\mathcal{M}$ be a Markov chain with underlying graph $\mathcal{G}$, and let $\mathcal{G}'$ be a strong component of $\mathcal{G}$. Then all states in $\mathcal{G}'$ have the same period (or are aperiodic).*

Fact 5.5 *Finite Markov chain $\mathcal{M}$ has a unique stationary distribution if the underlying graph of $\mathcal{M}$ has a unique absorbing strong component.*

Turning to our Markov chains Γ_{sim}, $\Gamma_{\text{odd-even}}$ and Γ_{ripple} (also the relaxed versions), we define three states, $v_{\text{full}}, v_{\text{even}}, v_{\text{odd}} \in V$, which will be proved to be normal states. In the state v_{full} all the switches are filled with two packets, each of them heading in the straight direction. v_{even} is identical to v_{full} on *even* stages, i. e. on stage H, stage $(H-2)$, and so on, and has empty switches on the other stages. State v_{odd} is similar to v_{even}, but now *odd* stages are full, while even ones are empty. The formal definitions:

$$v_{\text{full}}(i) = (\downarrow,\downarrow), \text{ for } i \in I \cup I_{H+1}$$

$$v_{\text{even}}((r,c)) = \begin{cases} (\downarrow,\downarrow) & \text{if } (H-r) \text{ is even,} \\ (\circ,\circ) & \text{otherwise,} \end{cases} \qquad \text{for } (r,c) \in I \cup I_{H+1};$$

$$v_{\text{odd}}((r,c)) = \begin{cases} (\downarrow,\downarrow) & \text{if } (H-r) \text{ is odd,} \\ (\circ,\circ) & \text{otherwise,} \end{cases} \qquad \text{for } (r,c) \in I \cup I_{H+1}.$$

Proposition 5.6 v_{full} *is an aperiodic normal state of* Γ_{ripple}.

Proof: We give a scenario of reaching, from any state, the state v_{full} (with a non-zero probability). First we force all packets in the network to choose only the straight direction in future steps (i.e. the crossing direction is not allowed from now on). In this way we avoid new collisions inside switches. There may still be some old collisions, but after few steps they disappear, if we give priority to packets with crossing direction (a collision is always between a straight directed packet and a crossing directed one). Now the packets are arranged in independent (non-colliding) paths, represented by the straight direction, and every packet follows its path (there is no blocking). Let us call any such a state *collision-free*.

If we continue entering packets into the network, which processes them continuously, we see that after H steps we obtain the state v_{full}. This state is aperiodic as with a non-zero probability we stay in the same state. □

Proposition 5.7 v_{even} *is an aperiodic normal state of* $\Gamma_{\text{odd-even}}$.

Proof: As in the proof of Proposition 5.6, we give a scenario of reaching v_{even}. In exactly the same way as there we can obtain a collision-free state. From now on all the paths proceed independently. We call a path *even* if it is filled by packets precisely as in the state v_{even}. We must argue that in H steps every path eventually becomes even. If a packet resides at an even stage and all packets below it as well, then all these packets will move down 2 stages in the next odd-even step. If a packet resides at an odd stage and all packets below it reside at even stages, then after one step all these packets will have reached even stages. By induction, one sees that from bottom to top, after at most H steps, all packets reach even stages and act simultaneously (see Fig. 3). The argument for aperiodicity is the same as in Proposition 5.6. □

Proposition 5.8 *Both* v_{even}, v_{odd} *are normal states of* Γ_{sim}. *They are periodic with period* 2.

Proof: We proceed as in the previous propositions to obtain a state consisting of independent paths (a collision-free state). We call a path *odd* (*even*) if it is filled by packets precisely as in the state v_{odd} (v_{even}). Similarly Proposition 5.7, we can reach a state in which all the paths are either odd or even. Let us consider one non-colliding path and the packet which have all packets below residing at even (odd) stages. If our packet resides also at an even (odd) stage then in the next simultaneous step all

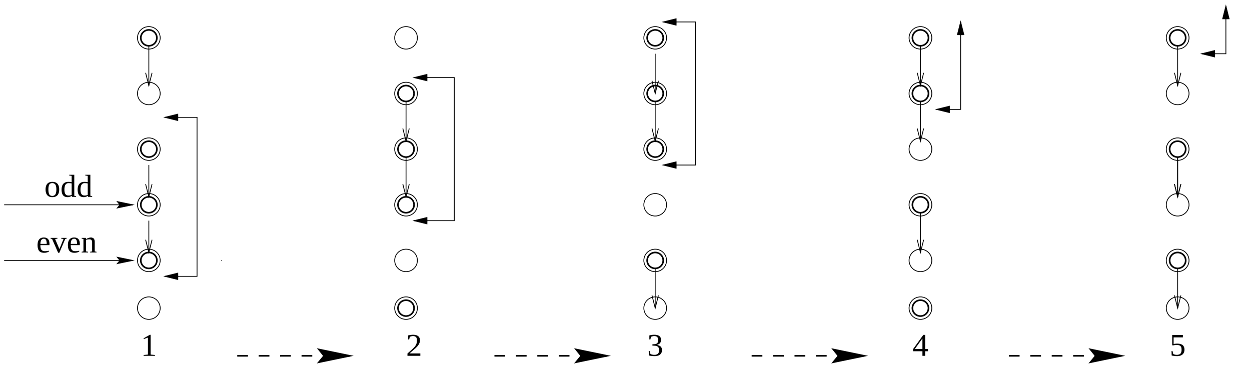

Figure 3: Initially blocked packets start moving continuously in one non-colliding path. The figure shows two steps of $\Gamma_{\mathrm{odd-even}}$ (i.e. four half-steps).

these packets move down and reach odd (even) stages. If the packet resides at an odd (even) stage – that means it is blocked by the packet below (via stage blocking) then after one simultaneous step it is not blocked anymore and in the next step it will move simultaneously with packets below it. By induction, one sees that after at most H step every path is either odd or even.

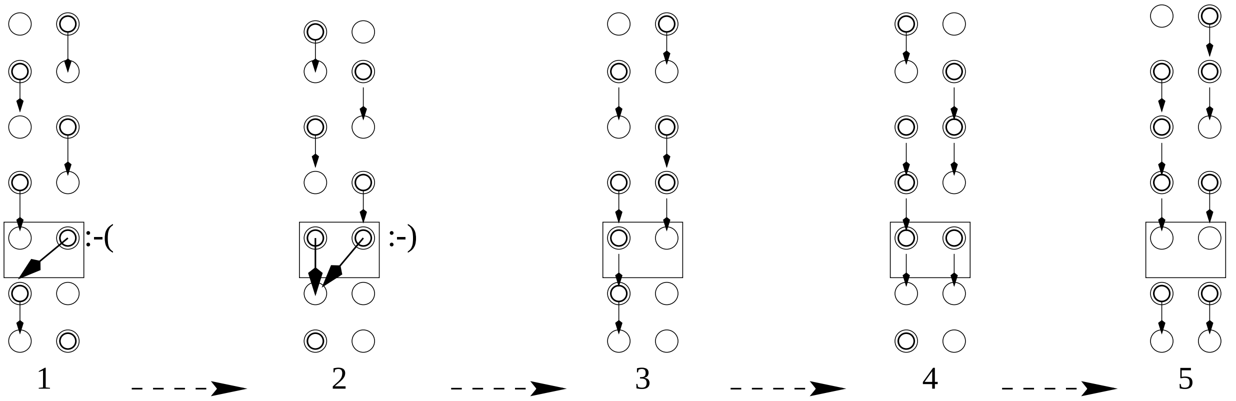

Figure 4: Changing the parity of one path. The packet depicted as :-(is blocked in the first step and it wins the collision in a switch during the second step :-).The left path preserves its parity, while the parity of the right one changes.

The situation is a bit more complicated than in Proposition 5.7, as we have to deal with paths of different parities. All the paths can be divided into two disjoint groups such that each two paths from the same group have at every moment the same parity.

Now we want to force all paths to have the same parity (as in the states v_{even} and v_{odd}). It turns out that we can change the parity of one path, assuming the existence of another path with different parity that shares one of its switches with our path. Figure 4 shows two paths which have a common switch. We want to change the parity of the right one. To this aim we force a packet from the right path to choose the crossing direction, which causes stage blocking. In the next step we have a collision in this switch. We give priority to the packet with the crossing direction, and after one more step obtain the situation marked in Fig. 4 as number 4. It is easy to observe that the right path changed its group, while the left one remained in the same group. There are still some packets on both paths which are clustered together (three packets on the left path and two on the right one), so precisely speaking the paths are not even nor odd at the moment, but as was argued in previous proof, clusters of packets will move up and finally disappear at the first stage.

Obviously not every pair of paths have a common switch, but we deal with this in the following way. Let us consider a graph with the set of all paths as its set of vertices and the set of switches as edges. There is an edge between two paths when they cross the same switch. It is easy to see that this graph is a hypercube. A hypercube is connected, so we conclude that if not all paths have the same parity, then there are two paths of different parity which cross the same switch. We can change the parity of one of these two paths, increasing or decreasing the number of even paths. Iterating this we can obtain any of the two states v_{even} and v_{odd}.

Now we prove the periodicity of these two states. We consider two sets of states of Γ_{sim}. The first one contains all states in which all even stages are stage blocked (it means there is no packet at these stages which can move). We call this set "after Γ_1", as the described situation appears after performing Γ_1. Analogously we have the set of states "after Γ_2". It contains all states with odd stages blocked. These sets are disjoint, because there exists always at least one switch which is not blocked. For example in state v_{full} $(H+1)$st stage is not blocked, dually in empty state the stage 0 is not blocked. In the underlying graph of Γ_{sim} these sets form a bipartite subgraph by Lemmas 7.1 and 7.3 (which are proved in Section 7). One of the components of this bipartite graph contains v_{odd}, the other contains v_{even}, which implies that these states are periodic with period 2.

□

As a consequence of Facts 5.3 and 5.5 and Propositions 5.6, 5.7 and 5.8 we obtain:

Corollary 5.9 *The Markov chains Γ_{ripple}, $\Gamma_{\mathrm{odd-even}}$ and Γ_{sim} have unique stationary distributions.*

Now we look at the Markov chains for the relaxed model.

Proposition 5.10 v_{full} *is an aperiodic normal state of $\Gamma^*_{\mathrm{ripple}}$ (and also of Γ^*_{sim}, which follows by Theorem 9.1 from Section 9).*

Proof: We can repeat here the proof of Proposition 5.6 without any changes.

□

Proposition 5.11 v_{even} *is an aperiodic normal state of $\Gamma^*_{\mathrm{odd-even}}$.*

Proof: The proof proceeds as for Proposition 5.7. The only difference is that after obtaining collision-free state, after only one step stage blocking disappears (due to relaxation of packets). Now if we continue entering packets into the network we see that after at most H steps we reach v_{even} state. □

The following corollary is a conclusion from Facts 5.3 and 5.5 and Proposition 5.10 and 5.11.

Corollary 5.12 *The Markov chains $\Gamma^*_{\mathrm{ripple}}$, Γ^*_{sim} and $\Gamma^*_{\mathrm{odd-even}}$ have unique stationary distributions.*

6 Properties of ";" and "$\|$"

We develop here mathematical tools consisting of some notation, definitions and technical facts, which we apply in Sections 7, 8 and 9 to prove the results linking different models.

Throughout this section, let $P : A \rightsquigarrow B_1$, $Q, Q' : B_1 \times B_2 \rightsquigarrow C$ be arbitrary c-maps, for fixed sets A, B_1, B_2. Omitted proofs are easy and follow from definitions. Some lemmas will be applied implicitly in the following sections, namely Lemmas 6.2, 6.4 and 6.6.

Sometimes we need two c-maps to be equal only on some part of their domain, for example when they are sequentially composed after another c-map; this motivates the following definition:

Definition 6.1
*Q **is equal to** Q' **after** P, denoted by $Q = Q'(\lhd P)$ if $\forall a, (b_1, b_2) : P(a, b_1) > 0$ implies $Q(b_1, b_2) = Q'(b_1, b_2)$.*

Equality after a composed c-map $(P_1 \parallel P_2)$ is weaker than equality after each of the components P_i $(i = 1, 2)$:

Lemma 6.2 *Let $P_i : A \rightsquigarrow B_i$ for $i = 1, 2$ and $R, R' : B_1 \times B_2 \times B_3 \rightsquigarrow C$. Then*

$$R = R'(\lhd P_1) \quad \textit{implies} \quad R = R'(\lhd(P_1 \parallel P_2)).$$

In Section 3, when we defined the Markov chains for different models of packet flow, we used several atomic c-maps σ^r, τ^r, sharing the property that their domain was much "larger" than their co-domain. Moreover, their domains seemed to be much larger than necessary. For example, the c-map σ^r "operates" only on the rth and $(r + 1)$st stage, and is "independent" from what happens on other stages. Such an independence from a part of a domain is formalized as follows:

Definition 6.3 *Q **is independent of** B_2 (denoted by $Q \dashv B_2$) if*

$$\forall b_1 \in B_1, \forall b_2, b_2' \in B_2 \;\; : \;\; Q((b_1, b_2)) = Q((b_1, b_2')).$$

When $Q \dashv B_2$, we could define the c-map Q on the restricted domain B_1 without any loss of information. The following lemma is similar to Lemma 6.2, although the implication goes in the opposite direction:

Lemma 6.4 *Let $R : A \times B_1 \times B_2 \rightsquigarrow C$. Then*

$$R \dashv (B_1 \times B_2) \quad \textit{implies} \quad R \dashv B_1.$$

To make our notation more usable, we define a generalized sequential composition of c-maps $P;Q$ (denoted also by ;), which allows the co-domain of P to be "smaller" than the domain of Q, when Q is independent from the rest of the domain.

Definition 6.5 *For $Q \dashv B_2$, the* **generalized sequential composition**

$$(P;Q) : A \rightsquigarrow C$$

is defined as follows: $(P;Q)(a, c) = \sum_{b_1 \in B_1} P(a)(b_1) \cdot Q((b_1, b_2))(c)$, where $b_2 \in B_2$ is arbitrary.

Lemma 6.6 below compares both types of sequential composition. It enables to remove some subcomponents of a c-map.

Lemma 6.6 *For each* $R : A \rightsquigarrow B_2$, $Q \dashv B_2$ *implies* $P;Q = (P \parallel R);Q$.

The following lemma reduces the equality of c-maps to the equality of their small sub-components after a common predecessor.

Lemma 6.7 *For any* $R : B_1 \times B_2 \rightsquigarrow D$, $Q = Q'(\lhd P)$ *implies*

$$\begin{pmatrix} P \\ Q \parallel R \end{pmatrix} = \begin{pmatrix} P \\ Q' \parallel R \end{pmatrix}.$$

For further use let us slightly generalize the notion of the independence.

Definition 6.8 *We say that* Q **is independent of** P *(denoted* $Q \dashv P$*) if*

$$\forall a \in A, \forall b_1, b_1', b_2 \in B_2 : \; P(a,b_1), P(a,b_1') > 0 \Longrightarrow Q((b_1,b_2)) = Q((b_1',b_2))$$

Independence from a domain is stronger than independence from a c-map.

Lemma 6.9 $Q \dashv B_1$ *implies* $Q \dashv P$.

The following proposition essentially allows us to decompose a c-map into a parallel composition of independent components. It will be very useful for proofs of some results in the following sections.

Proposition 6.10 *Let* $P_i : A \rightsquigarrow B_i$, $Q_i : B_1 \times B_2 \rightsquigarrow C_i$, *for* $i = 1, 2$, *such that* $Q_1 \dashv P_2$, $Q_2 \dashv P_1$. *Then*

$$\begin{pmatrix} P_1 \parallel P_2 \\ Q_1 \parallel Q_2 \end{pmatrix} = \begin{pmatrix} P_1 \parallel P_2 \\ Q_1 \end{pmatrix} \parallel \begin{pmatrix} P_1 \parallel P_2 \\ Q_2 \end{pmatrix}.$$

Proof: Let $\mathcal{L}, \mathcal{R}$ denote the left and the right hand side of the equation, respectively. By definitions it follows (let B denotes $\{(b_1,b_2) | P_1(a,b_1) > 0 \text{ and } P_2(a,b_2) > 0\}$):

$$\mathcal{R}(a,(c_1,c_2)) = \left(\sum_{(b_1,b'_2) \in B} P_1(a,b_1) \cdot P_2(a,b'_2) \cdot Q_1((b_1,b'_2),c_1) \right) \cdot$$

$$\cdot \left(\sum_{(b_1',b_2) \in B} P_1(a,b_1') \cdot P_2(a,b_2) \cdot Q_2((b_1',b_2),c_2) \right) =$$

$$= \sum_{(b_1,b_2',b_1',b_2) \in B \times B} P_1(a,b_1) \cdot P_2(a,b_2) \cdot Q_1((b_1,b_2'),c_1) \cdot Q_2((b_1',b_2),c_2) \cdot P_1(a,b_1') \cdot P_2(a,b_2').$$

Now by the independences $Q_1 \dashv P_2$ and $Q_2 \dashv P_1$ we have

$$Q_1((b_1,b'_2),c_1) = Q_1((b_1,b_2),c_1), \quad \text{for any } \; b_2' \text{ such that } P_2(a,b_2') > 0,$$

and analogously

$$Q_2((b_1',b_2),c_2) = Q_2((b_1,b_2),c_2), \quad \text{for any } \; b_1' \text{ such that } P_1(a,b_1') > 0,$$

thus, we obtain:

$$\mathcal{R}(a,(c_1,c_2)) = \sum_{(b_1,b_2) \in B} [P_1(a,b_1) \cdot P_2(a,b_2)] \cdot [Q_1((b_1,b_2),c_1) \cdot Q_2((b_1,b_2),c_2)] \cdot \tag{1}$$

$$\cdot \sum_{(b'_1,b'_2)\in B} P_1(a,b'_1)\cdot P_2(a,b'_2).$$

Now, as the following holds:

$$\sum_{(b'_1,b'_2)\in B} P_1(a,b'_1)\cdot P_2(a,b'_2) = \sum_{b\in B}(P_1 \parallel P_2)(a,b) = 1,$$

we conclude that the sum in (1) is equal to $\mathcal{L}(a,(c_1,c_2))$, which completes the proof. $\square$

When we replace the assumption on independence from a c-map in Proposition 6.10 by the stronger notion of independence from a domain, the c-map $(P_1 \parallel P_2)$ can be replaced by any other, needing not to be a parallel composition:

Proposition 6.11 *Let $R : A \rightsquigarrow B_1 \times B_2$, $Q_i : B_1 \times B_2 \rightsquigarrow C_i$, for $i = 1,2$, such that $Q_1 \dashv B_2$, $Q_2 \dashv B_1$. Then*

$$\begin{pmatrix} R \\ Q_1 \parallel Q_2 \end{pmatrix} = \begin{pmatrix} R \\ Q_1 \end{pmatrix} \parallel \begin{pmatrix} R \\ Q_2 \end{pmatrix}.$$

Proof: The proof is analogous to the proof of Proposition 6.10. The only difference is that the set $\{b_1|P_1(a,b_1)>0\}$ (resp. $\{b_2|P_2(a,b_2)>0\}$) should be replaced by the whole co-domain B_1 (resp. B_2) of P_1 (resp. P_2). $\square$

The following lemma regards some nice preservation properties of parallel composition:

Lemma 6.12 *Let $Q_i, Q'_i : B_1 \times B_2 \rightsquigarrow C_i$, for $i = 1,2$ Then*

(i) $Q_1 = Q'_1(\lhd P)$ *and* $Q_2 = Q'_2(\lhd P)$ *implies* $(Q_1 \parallel Q_2) = (Q'_1 \parallel Q'_2)(\lhd P)$;

(ii) $Q_1 \dashv B_1$ *and* $Q_2 \dashv B_1$ *implies* $(Q_1 \parallel Q_2) \dashv B_1$;

(iii) $Q_1 \dashv P$ *and* $Q_2 \dashv P$ *implies* $(Q_1 \parallel Q_2) \dashv P$.

Lemma 6.12 allows us to analyze (parallelly) composed c-map by looking only at smaller components.

7 Odd-even versus simultaneous

This section contains some results comparing the odd-even model with the simultaneous model, both with stage blocking. They are an extension of results from [5].

Recall the meaning of the components Γ_1 and Γ_2 of $\Gamma_{\mathrm{odd-even}}$ introduced in Section 3. Analogously let Γ^1_{sim} and Γ^2_{sim} denote components of Γ_{sim}, i.e. $\Gamma_{\mathrm{sim}} = \Gamma^1_{\mathrm{sim}};\Gamma^2_{\mathrm{sim}}$. In the following lemma we show that after Γ_1, the full power of a simultaneous step has the same effect as Γ_2.

Lemma 7.1 $\Gamma_1;\Gamma_{\mathrm{sim}} = \Gamma_1;\Gamma_2$.

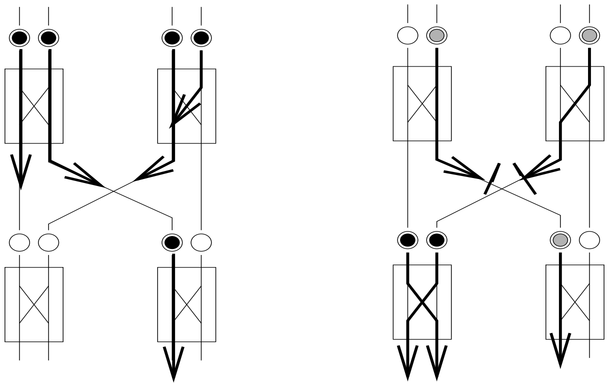

Figure 5: One even group of switches before and after Γ_1; on the right side the black packet moved during Γ_1, while the grey ones did not move.

Proof: We start with some intuitive arguments, the formal proof follows. All the switches can be divided into groups of four (nodes (r,c), $(r,c^{(r)})$, $(r+1,c)$, $(r+1,c^{(r)})$). We call a group *even* or *odd*, depending on the parity of $H-r$. Each switch belongs to precisely two groups (Fig. 5 shows an even group). During Γ_1 even groups are active, while odd ones stay idle. During Γ_2 the roles change: only odd groups perform their actions. During Γ_{sim} odd groups do precisely the same as during Γ_2. We only should observe that after Γ_1 even groups, during Γ_{sim} do nothing. This can be established by a careful analysis of possible cases. As in Γ_1 groups operate independently, we will consider one group. We should only think about non-empty buffers at a switch at stage r. Let us consider one such a buffer. The packet staying in it could not have come during Γ_1 as odd groups were idle. There are two possible causes why it did not move forward during Γ_1. If the reason was stage blocking, i.e. another packet staying in the appropriate buffer at the next stage, then this packet still remains in its buffer (as it is still blocked). If the reason was a collision in the switch, then our packet is now blocked by the winner. In both cases our packet can not move forward.

The formal proof is divided in two steps. The assumption is made here that H is even, although for odd H the proof runs analogously.

1) First we prove that $\Gamma_1;\Gamma^1_{\text{sim}} = \Gamma_1;\Gamma^1_2$. From Lemma 6.7 we conclude that it is enough to prove that $\sigma^r = \pi^r(\lhd\Gamma_1)$, for any even r. The principal idea now is to carefully separate the fragments of both c-maps of which σ^r, τ^r, π^r and ω^r are independent. Some observations are put in the following lemma as they are used in other sections as well:

Lemma 7.2 *For $1 \le r \le H+1$ and $0 \le c \le W$ we have the following:*

(1) $\tau^{r,c} \dashv (S\times\{0,1\}^2)^{D'_{r,c}}, \quad \omega^{r,c} \dashv (S\times\{0,1\}^2)^{(I\cup I_0\cup I_{H+1})\setminus\{(r,c)\}},$
where $D'_{r,c} = (I\cup I_0\cup I_{H+1})\setminus\{(r,c),(r-1,c),(r-1,c^{((r-1))})\}.$

(2) $\tau^r \dashv (S\times\{0,1\}^2)^{(I\cup I_0\cup I_{H+1})\setminus(I_r\cup I_{r-1})}, \quad \omega^r \dashv (S\times\{0,1\}^2)^{(I\cup I_0\cup I_{H+1})\setminus I_r}.$

(3) $\tau^{r,c} \dashv \sigma^{r-1,c}, \quad \tau^{r,c} \dashv \sigma^{r-1,c^{((r-1))}}.$

(4) $\tau^r \dashv \sigma^q, \quad \omega^r \dashv \sigma^q,$ *where* $q \ne r.$

(5) $\tau^r \dashv \pi^q, \quad \omega^r \dashv \pi^q,$ *for any* $q.$

Moreover we have:

(1') $\sigma^{r,c} \dashv S^{D_{r,c}}$, *where* $D_{r,c} = (I \cup I_{H+1}) \setminus \{(r,c),(r+1,c),(r+1,c^{(r)})\}$,
$\pi^{r,c} \dashv S^{(I \cup I_{H+1}) \setminus \{(r,c)\}}$.

(2') $\sigma^{r} \dashv S^{(I \cup I_{H+1}) \setminus (I_r \cup I_{r+1})}$ $(r \leq H)$, $\sigma^{H+1} \dashv S^{I}$,
$\pi^{r} \dashv S^{(I \cup I_{H+1}) \setminus I_r}$.

(5') $\sigma^r \dashv \omega^q$, $\pi^r \dashv \omega^q$, *for any* q.

Proof: (of Lemma 7.2) Points (1) and (3) follow from the definitions. Intuitive justification for (3) is as follows: output of packets given to $\tau^{r,c}$ by $\sigma^{r-1,c}$ (or $\sigma^{r-1,c^{(r-1)}}$) is deterministic (in any initial state). (2) is implied by (1) and Lemma 6.12. (4) is a consequence of (2) and Lemma 6.9, while (5) holds as π^q is deterministic, for any q.

Arguments for (1'), (2') and (5') are similar to (1), (2) and (5), respectively. □

Now we are prepared to apply Proposition 6.10 together with Lemmas 6.9 and 6.12 in order to divide Γ_1 into independent parts. We obtain:

$$\Gamma_1 = \left(\begin{array}{ll} \sigma^0 & \| \pi^1 \\ & \tau^1 \end{array} \right) \| \left(\begin{array}{ll} \sigma^2 & \| \pi^3 \\ \omega^2 & \| \tau^3 \end{array} \right) \| \cdots \| \left(\begin{array}{ll} \sigma^r & \| \pi^{r+1} \\ \omega^r & \| \tau^{r+1} \end{array} \right) \| \cdots$$

It is enough to prove:

$$\sigma^r = \pi^r \left(\lhd \left(\begin{array}{ll} \sigma^r & \| \pi^{r+1} \\ \omega^r & \| \tau^{r+1} \end{array} \right) \right).$$

We want to consider as small pieces of the network as possible. Once more we use Proposition 6.10 together with the lemmas to see that it remains to show the following:

$$\sigma^{r,c} = \pi^{r,c} \left(\lhd \left(\begin{array}{llll} \sigma^{r,c} & \| \sigma^{r,c^{(r)}} & \| \pi^{r+1,c} & \| \pi^{r+1,c^{(r)}} \\ \omega^{r,c} & \| \omega^{r,c^{(r)}} & \| \tau^{r+1,c} & \| \tau^{r+1,c^{(r)}} \end{array} \right) \right).$$

This can easily be observed by carefully looking at the definitions of σ and τ.

Remark A slight difference arises for $r = 0$ or $r = H$, but the argument is essentially the same.

2) Second, we prove that $\Gamma_2^1 ; \Gamma_{\rm sim}^2 = \Gamma_2$. By the decomposition analogous to the above it is sufficient to show the following:

$$\left(\begin{array}{ll} \pi^r & \| \sigma^{r+1} \\ & \tau^{r+1} \end{array} \right) = \left(\begin{array}{ll} \pi^r & \| \sigma^{r+1} \\ & \omega^{r+1} \end{array} \right), \quad \text{for even } r,$$

and after further decomposition:

$$\left(\begin{array}{lll} \pi^{r,c} & \| \pi^{r,c()} & \| \sigma^{r+1,c} \\ & & \tau^{r+1,c} \end{array} \right) = \left(\begin{array}{lll} \pi^{r,c} & \| \pi^{r,c()} & \| \sigma^{r+1,c} \\ & & \omega^{r+1,c} \end{array} \right), \quad \text{for even } r \text{ and any } c,$$

which can be verified from the definitions.

From 1) and 2) we have $\Gamma_1 ; \Gamma_{\rm sim} = \Gamma_1 ; \Gamma_2^1 ; \Gamma_{\rm sim}^2 = \Gamma_1 ; \Gamma_2$.

□

The following lemma can be proved entirely analogously to Lemma 7.1.

Lemma 7.3 $\Gamma_2 ; \Gamma_{\text{sim}} = \Gamma_2 ; \Gamma_1$.

By $(P)^k$, for any Markov chain $P : A \rightsquigarrow A$, we mean the composition $P ; P ; \cdots ; P$ (k times). The following theorem describes the relation between the simultaneous model and the odd-even model:

Theorem 7.4 $(\Gamma_{\text{odd-even}})^k = \Gamma_1 ; (\Gamma_{\text{sim}})^{2k-1}$ *for any* $k \geq 1$.

Proof: We apply Lemmas 7.1 and 7.3 alternatingly to obtain the desired equality.

$$(\Gamma_{\text{odd-even}})^k = (\Gamma_1 ; \Gamma_2)^k \overset{\text{Lemma 7.1}}{=} (\Gamma_1 ; \Gamma_2)^{k-1} ; \Gamma_1 ; \Gamma_{\text{sim}} \overset{\text{Lemma 7.3}}{=} \cdots$$

$$\cdots \overset{\text{Lemma 7.3}}{=} \Gamma_1 ; (\Gamma_{\text{sim}})^{2k-1}.$$

$\square$

Let us combine already proved results about the equivalences between models with the knowledge about the existence of stationary distributions of appropriate Markov chains.

Corollary 7.5 *Let p be the stationary distribution of $\Gamma_{\text{odd-even}}$. Then p and $(p ; \Gamma_1)$ are both stationary distributions of $(\Gamma_{\text{sim}})^2$ (we treat p as a c-map from a singleton set).*

Proof:

$$p = p ; (\Gamma_{\text{odd-even}})^2 \overset{\text{Theorem 7.4}}{=} p ; \Gamma_1 ; (\Gamma_{\text{sim}})^3 =$$

$$(p ; \Gamma_1 ; \Gamma_{\text{sim}}) ; (\Gamma_{\text{sim}})^2 \overset{\text{Lemma 7.1}}{=} (p ; \Gamma_{\text{odd-even}}) ; (\Gamma_{\text{sim}})^2 = p ; (\Gamma_{\text{sim}})^2,$$

and

$$p ; \Gamma_1 = p ; \Gamma_{\text{odd-even}} ; \Gamma_1 = p ; \Gamma_1 ; \Gamma_2 ; \Gamma_1 \overset{\text{Lemmas 7.3,7.1}}{=} (p ; \Gamma_1) ; (\Gamma_{\text{sim}})^2.$$

$\square$

8 Odd-even versus ripple

In this section we compare the odd-even model and the ripple model, both with stage blocking.

In the following we will need projection c-maps $Id_{W_1 \times W_2 \to W_1} : W_1 \times W_2 \rightsquigarrow W_1$, which are defined analogously to the sequential identities $Id^{;}$ from Section 3.2:

$$Id_{W_1 \times W_2 \to W_1}((w_1, w_2), w_1) = 1.$$

The following lemma will be used, together with results from Section 6, to prove the main theorem of this section comparing the two models.

Lemma 8.1 *Let $P, S : W_1 \times W_2 \rightsquigarrow W_1$, $Q : W_1 \times W_2 \rightsquigarrow W_2$, be such that $P \dashv W_2$, $S \dashv W_2$, $Q \dashv W_1$. Then*

$$\left(\begin{array}{ll} P & \| \, Id_{W_1 \times W_2 \to W_2} \\ S & \| \, Q \end{array} \right) = \left(\begin{array}{ll} P & \| \, Q \\ S & \| \, Id_{W_1 \times W_2 \to W_2} \end{array} \right)$$

Proof: Let us denote the left side of the equality by $\mathcal{L}$ and the right one by $\mathcal{R}$. We have:

$$\mathcal{L} \overset{(1)}{=} \begin{pmatrix} P & \| \, Id_{W_1 \times W_2 \to W_2} \\ S & \end{pmatrix} \| \begin{pmatrix} P & \| \, Id_{W_1 \times W_2 \to W_2} \\ & Q \end{pmatrix} \overset{(2)}{=}$$

$$\overset{(2)}{=} \begin{pmatrix} P \\ S \end{pmatrix} \| \begin{pmatrix} Id_{W_1 \times W_2 \to W_2} \\ Q \end{pmatrix} \overset{(3)}{=} \begin{pmatrix} P \\ S \end{pmatrix} \| \begin{pmatrix} Q \\ Id_{W_1 \times W_2 \to W_2} \end{pmatrix}$$

$$\overset{(4)}{=} \begin{pmatrix} P & \| \, Q \\ S & \end{pmatrix} \| \begin{pmatrix} P & \| \, Q \\ & Id_{W_1 \times W_2 \to W_2} \end{pmatrix} \overset{(5)}{=} \mathcal{R}$$

Equality (1) follows from Proposition 6.10, since we have $S \dashv W_2$ and $Q \dashv W_1$. In (2) simplify the components by removing P and $Id_{W_1 \times W_2 \to W_2}$, as justified by Lemma 6.6. It can be verified from the definitions that Q and $Id_{W_1 \times W_2 \to W_2}$ can be exchanged (3). Steps (4) and (5) are similar to (1) and (2), but the direction is inverted. $\square$

In our simulation we use the c-map $\triangle : V \rightsquigarrow V$ defined as:

$$\begin{array}{rl} \triangle = & (\pi^0 \| \, \pi^1 \| \, \pi^2 \| \, \pi^3 \| \cdots \| \, \pi^{H-2} \, \| \, \pi^{H-1} \, \| \, \pi^H \, \| \, \boldsymbol{\sigma^{H+1}})\,; \\ & (\omega^1 \, \| \, \omega^2 \| \, \omega^3 \| \cdots \| \, \omega^{H-2} \, \| \, \omega^{H-1} \, \| \, \omega^H \, \| \, \omega^{H+1}\,)\,; \\ & (\pi^0 \| \, \pi^1 \| \, \pi^2 \| \, \pi^3 \| \cdots \| \, \pi^{H-2} \, \| \, \pi^{H-1} \, \| \, \boldsymbol{\sigma^H} \| \, \pi^{H+1}\,)\,; \\ & (\omega^1 \, \| \, \omega^2 \| \, \omega^3 \| \cdots \| \, \omega^{H-2} \, \| \, \omega^{H-1} \, \| \, \omega^H \, \| \, \boldsymbol{\tau^{H+1}})\,; \\ & (\pi^0 \| \, \pi^1 \| \, \pi^2 \| \, \pi^3 \| \cdots \| \, \pi^{H-2} \, \| \, \boldsymbol{\sigma^{H-1}} \| \, \pi^H \, \| \, \boldsymbol{\sigma^{H+1}})\,; \\ & (\omega^1 \, \| \, \omega^2 \| \, \omega^3 \| \cdots \| \, \omega^{H-2} \, \| \, \omega^{H-1} \, \| \, \boldsymbol{\tau^H} \| \, \omega^{H+1}\,)\,; \\ & (\pi^0 \| \, \pi^1 \| \, \pi^2 \| \, \pi^3 \| \cdots \| \, \boldsymbol{\sigma^{H-2}} \| \, \pi^{H-1} \, \| \, \boldsymbol{\sigma^H} \| \, \pi^{H+1}\,)\,; \\ & (\omega^1 \, \| \, \omega^2 \| \, \omega^3 \| \cdots \| \, \omega^{H-2} \, \| \, \boldsymbol{\tau^{H-1}} \| \, \omega^H \, \| \, \boldsymbol{\tau^{H+1}})\,; \\ & \quad \vdots \qquad\qquad\qquad\qquad \vdots \\ & \\ & (\pi^0 \| \, \pi^1 \, \| \, \boldsymbol{\sigma^2} \| \, \pi^3 \, \| \cdots \| \, \boldsymbol{\sigma^{H-2}} \| \, \pi^{H-1} \, \| \, \boldsymbol{\sigma^H} \| \, \pi^{H+1}\,)\,; \\ & (\omega^1 \, \| \, \omega^2 \, \| \, \boldsymbol{\tau^3} \| \cdots \| \, \omega^{H-2} \, \| \, \boldsymbol{\tau^{H-1}} \| \, \omega^H \, \| \, \boldsymbol{\tau^{H+1}})\,; \\ & (\pi^0 \| \, \boldsymbol{\sigma^1} \| \, \pi^2 \, \| \, \boldsymbol{\sigma^3} \| \cdots \| \, \pi^{H-2} \, \| \, \boldsymbol{\sigma^{H-1}} \| \, \pi^H \, \| \, \boldsymbol{\sigma^{H+1}})\,; \\ & (\omega^1 \, \| \, \boldsymbol{\tau^2} \| \, \omega^3 \, \| \cdots \| \, \boldsymbol{\tau^{H-2}} \| \, \omega^{H-1} \, \| \, \boldsymbol{\tau^H} \| \, \omega^{H+1}\,)\,; \end{array}$$

It consists, intuitively speaking, of consecutive odd-even steps (Γ_2 followed by Γ_1, this followed by Γ_2 and so on), which are partially inactive. First Γ_2 acts only on the $(H+1)$st stage (idling on other stages), next Γ_1 acts on the Hth and the $(H+1)$st stage. This it is continued until the last step, which is simply full Γ_2, assumed that H is even. When H is odd, we should remove the last step (equal to Γ_2) from $\triangle$.

The following theorem gives a link between the odd-even model and the ripple model.

Theorem 8.2 $\triangle \,; \Gamma_{\text{odd-even}} = \Gamma_{\text{ripple}} \,; \triangle$.

Proof: The proof is given for even H; for odd H only a slight modification is needed. If we look carefully at the definition of $A = \triangle \,; \Gamma_{\text{odd-even}}$ and $B = \Gamma_{\text{ripple}} \,; \triangle$, we notice that the number of c-maps σ, τ (at any stage of the network) is the same in both of them. Moreover, in $(\triangle \,; \Gamma_{\text{odd-even}})$ we can distinguish two fragments, one (consisting of the earliest σ's and τ's from all stages) being exactly

Γ_{ripple} and another, which is equal to $\triangle$. Below we have c-map A (the part in bold should correspond to the "triangle" component of B). For clarity, the idle actions ω and π are suppressed.

$$
\begin{array}{rl}
\triangle\,;\Gamma_{\text{odd-even}} = & (\;\;\|\quad\|\quad\|\quad\|\cdots\|\quad\|\quad\|\quad\|\,\sigma^{H+1}\,);\\
& \quad(\quad\|\quad\|\quad\|\cdots\|\quad\|\quad\|\quad\|\quad);\\
& (\;\;\|\quad\|\quad\|\quad\|\cdots\|\quad\|\quad\|\,\sigma^{H}\,\|\quad);\\
& \quad(\quad\|\quad\|\quad\|\cdots\|\quad\|\quad\|\quad\|\,\tau^{H+1}\,);\\
& (\;\;\|\quad\|\quad\|\quad\|\cdots\|\quad\|\,\sigma^{H-1}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H+1}});\\
& \quad(\quad\|\quad\|\quad\|\cdots\|\quad\|\quad\|\,\tau^{H}\,\|\quad);\\
& (\;\;\|\quad\|\quad\|\quad\|\cdots\|\,\sigma^{H-2}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H}}\|\quad);\\
& \quad(\quad\|\quad\|\quad\|\cdots\|\quad\|\,\tau^{H-1}\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{H+1}});\\
& \qquad\vdots \qquad\qquad\qquad\qquad\qquad \vdots\\
& \\
& (\;\;\|\quad\|\,\sigma^{2}\,\|\quad\|\cdots\|\,\boldsymbol{\sigma}^{\boldsymbol{H-2}}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H}}\|\quad);\\
& \quad(\quad\|\quad\|\,\tau^{3}\,\|\cdots\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{H-1}}\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{H+1}});\\
& (\;\;\|\,\sigma^{1}\,\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{3}}\|\cdots\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H-1}}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H+1}});\\
& \quad(\quad\|\,\tau^{2}\,\|\quad\|\cdots\|\,\boldsymbol{\tau}^{\boldsymbol{H-2}}\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{H}}\|\quad);\\
& (\sigma^{0}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{2}}\|\quad\|\cdots\|\,\boldsymbol{\sigma}^{\boldsymbol{H-2}}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H}}\|\quad);\\
& \quad(\tau^{1}\;\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{3}}\|\cdots\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{H-1}}\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{H+1}});\\
& (\;\;\|\,\boldsymbol{\sigma}^{\boldsymbol{1}}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{3}}\|\cdots\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H-1}}\|\quad\|\,\boldsymbol{\sigma}^{\boldsymbol{H+1}});\\
& \quad(\quad\|\,\boldsymbol{\tau}^{\boldsymbol{2}}\|\quad\|\cdots\|\,\boldsymbol{\tau}^{\boldsymbol{H-2}}\|\quad\|\,\boldsymbol{\tau}^{\boldsymbol{H}}\|\quad)
\end{array}
$$

The only problem is that in $(\triangle\,;\Gamma_{\text{odd-even}})$ the ripple and "triangle" parts proceed "almost simultaneously", while in $(\Gamma_{\text{ripple}}\,;\triangle)$ they are composed sequentially. We will prove that A can be obtained from B by moving the triangle $\triangle$ up, in such a way that the the c-map is unchanged. This will be done in several steps, each of them consisting of an atomic exchange allowed by Lemma 8.1.

We can view the Markov chain Γ_{ripple} as a sequential composition of c-maps working on consecutive stages:

$$\Gamma_{\text{ripple}} = \Gamma^{H+2}_{\text{ripple}}\,;\Gamma^{H+1}_{\text{ripple}}\,;\,\cdots\,;\Gamma^{1}_{\text{ripple}}.$$

$\Gamma^{i}_{\text{ripple}}$ denotes the flow out of packets from the $(i-1)$st stage. Analogously we decompose $\triangle$ into several sequential phases:

$$\triangle = \triangle^{H+1}\,;\triangle^{H}\,;\,\cdots\,;\triangle^{1}.$$

We number atomic steps (exchanges) (i,j) where $i=(H+1),\ldots,2$, $j=1,\ldots,(i-1)$. During step (i,j) we move $\triangle^{i}$ up and compose it in parallel with $\Gamma^{j}_{\text{ripple}}$.

Formally we apply Lemma 8.1 to the following c-maps:

$$P = \begin{pmatrix} \pi^{0} & \|\,\pi^{1} & \|\cdots & \|\,\pi^{j-2} & \|\,\sigma^{j-1} & \|\,\pi^{j} \\ & \omega^{1} & \|\cdots & \|\,\omega^{j-2} & \|\,\omega^{j-1} & \|\,\tau^{j} \end{pmatrix},$$

$$S = \begin{pmatrix} \pi^{0} & \|\,\pi^{1} & \|\cdots & \|\,\sigma^{j-2} & \|\,\pi^{j-1} & \|\,\pi^{j} \\ & \omega^{1} & \|\cdots & \|\,\omega^{j-2} & \|\,\tau^{j-1} & \|\,\omega^{j} \end{pmatrix}, \quad \text{and}$$

$$Q = \begin{pmatrix} \pi^{j+1} & \|\,\pi^{j+2} & \|\cdots & \|\,\sigma^{i} & \|\,\pi^{i+1} & \|\,\sigma^{i+2} & \|\,\pi^{i+3} & \|\,\sigma^{i+4} & \|\cdots \\ \omega^{j+1} & \|\,\omega^{j+2} & \|\cdots & \|\,\omega^{i} & \|\,\tau^{i+1} & \|\,\omega^{i+2} & \|\,\tau^{i+3} & \|\,\omega^{i+4} & \|\cdots \end{pmatrix}.$$

An idling c-map $J = Id_{V \to V_{j+1} \times \ldots \times V_{H+1}}$ is a projection defined as

$$J = \begin{pmatrix} \pi^{j+1} & \| \, \pi^{j+2} & \| \cdots & \| \, \pi^{H+1} \\ \omega^{j+1} & \| \, \omega^{j+2} & \| \cdots & \| \, \omega^{H+1} \end{pmatrix},$$

where $V_r = S^{I_r}$ represents rth stage of the network. Notice that P and S are simply fragments of two neighbouring ripple stages ($\Gamma^{j}_{\text{ripple}}$ and $\Gamma^{j-1}_{\text{ripple}}$, respectively) and Q is a fragment of $\triangle^{i}$. The c-map $\Gamma^{j}_{\text{ripple}}$ can be decomposed into $P \| J$ (by Lemma 7.2 from Section 7), and similarly we obtain $S \| Q$. For $j = 1$, the c-map S should be slightly modified: σ^{j-2} and τ^{j-1} should be replaced by π^{j-2} and ω^{j-1}, respectively. Moreover, in this case, after applying Lemma 8.1, c-map $S \| J$ consists only of π's and ω's, thus $S \| J = Id^{\,\vdots}_{V}$, so we remove it (by Proposition 3.3). $\square$

Remark In the proof above we need only very general properties of c-maps $\sigma^{r,c}$ and $\tau^{r,c}$, mainly the independence from fragments of its domains (as in Lemma 7.2). This means that the proof applies to any other model fulfilling this requirement. Especially it will be adapted in Section 9 to the relaxed model of packet flow.

The following corollary from Theorem 8.2 can be easily proved by induction:

Corollary 8.3 $\triangle ; (\Gamma_{\text{odd-even}})^k = (\Gamma_{\text{ripple}})^k ; \triangle$, *for any* $k \geq 1$.

Theorem 8.2 gives us a kind of simulation of the ripple model by the odd-even model:

Corollary 8.4 *Let p, q be probability distributions on V. Then*

$$p ; (\Gamma_{\text{ripple}})^k = q \qquad \textit{implies } (p ; \triangle) ; (\Gamma_{\text{odd-even}})^k = q ; \triangle.$$

Having clarified the relation between Γ_{ripple} and $\Gamma_{\text{odd-even}}$, we conclude with the comparison of stationary distributions.

Corollary 8.5 *Let p_{ripple} and $p_{\text{odd-even}}$ denote stationary distributions of Markov chains Γ_{ripple} and $\Gamma_{\text{odd-even}}$, respectively. Then we have:*

$$p_{\text{ripple}} ; \triangle = p_{\text{odd-even}}$$

(p_{ripple} and $p_{\text{odd-even}}$ are treated here as c-maps from a singleton set).

9 Results for the relaxed models

The equivalences in the relaxed case are not the same as in full blocking. Surprisingly, it turns out that the ripple model and the simultaneous model are identical, which we show in Theorem 9.1. On the other hand Theorem 8.2 can be adapted to the relaxed model, which is done in Theorem 9.2.

Theorem 9.1 *Ripple and simultaneous models are equal in the relaxed case, i.e.*

$$\Gamma^{*}_{\text{ripple}} = \Gamma^{*}_{\text{sim}}.$$

Proof: We show by induction that all rows of Γ^*_{ripple} can be "collapsed" into 2 rows of Γ^*_{sim}. In the proof we suppress the indices of c-maps π and ω, to improve readability. We start with the first four rows of Γ^*_{ripple} (base of induction) and show the following:

$$\begin{pmatrix} \pi^H \| \, \sigma_*^{H+1} \\ \omega \| \, \omega \\ \sigma_*^H \| \, \pi \\ \omega \| \, \tau_*^{H+1} \end{pmatrix} = \begin{pmatrix} \sigma_*^H \| \, \sigma_*^{H+1} \\ \omega \| \, \tau_*^{H+1} \end{pmatrix} \tag{2}$$

(The c-maps in (2) are only fragments of the appropriate rows of Γ^*_{ripple} and Γ^*_{sim}, but it is easy to observe that they can be "separated" (by Proposition 6.10) from the rest, consisting in both cases of only idling c-maps, thus equal to a projection $Id_{V \to V_1 \times \cdots \times V_{H-1}}$ cf. Section 8.)

Notice that the last rows in (2) are the same on both sides. Consider the first three rows of the left side. We have, by Propositions 6.10 and 6.6:

$$\begin{pmatrix} \pi \| \, \sigma_*^{H+1} \\ \omega \| \, \omega \\ \sigma_*^H \| \, \pi \end{pmatrix} = \begin{pmatrix} \pi \| \, \sigma_*^{H+1} \\ \omega \| \, \omega \\ \sigma_*^H \end{pmatrix} \| \begin{pmatrix} \pi \; \| \sigma_*^{H+1} \\ \omega \; \| \; \omega \\ \pi \end{pmatrix} =$$

$$= \begin{pmatrix} \pi \\ \omega \\ \sigma_*^H \end{pmatrix} \| \begin{pmatrix} \sigma_*^{H+1} \\ \omega \\ \pi \end{pmatrix} = \begin{pmatrix} \sigma_*^H \| \, \sigma_*^{H+1} \end{pmatrix},$$

which proves (2). We could apply Proposition 6.10 and Lemma 6.6, since σ_*^r is independent from domains just as σ^r and moreover we have: $\sigma_*^r \dashv V_{r+1}$, for $r \le H$. As the induction step we should show the following:

$$\begin{pmatrix} \pi & \| \, \sigma_*^i \| \, \mathcal{S}^{i+1} \\ \omega & \| \, \omega \, \| \, \mathcal{T}^{i+1} \\ \sigma_*^{i-1} & \| \, \pi \, \| \, \pi \\ \omega & \| \, \tau_*^i \| \, \omega \end{pmatrix} = \begin{pmatrix} \sigma_*^{i-1} & \| \, \sigma_*^i \| \, \mathcal{S}^{i+1} \\ \omega & \| \, \tau_*^i \| \, \mathcal{T}^{i+1} \end{pmatrix}, \tag{3}$$

where $\mathcal{S}^{i+1} = \|_{j=i+1\cdots,H+1} \, \sigma_*^j$ and $\mathcal{T}^{i+1} = \|_{j=i+1,\cdots,H+1} \, \tau_*^j$. Similarly as in (2), we omit here some part of Γ^*_{ripple} and Γ^*_{sim} consisting of idling c-maps, which is equal to a projection $Id_{V \to V_1 \times \cdots \times V_{i-2}}$. In the following let $\mathcal{L}$ and $\mathcal{R}$ denote the left and right hand side of (3).

Let us begin by transforming the left hand side of (3). The first two rows can be decomposed as follows:

$$\begin{pmatrix} \pi \| \, \sigma_*^i \| \, \mathcal{S}^{i+1} \\ \omega \| \, \omega \, \| \, \mathcal{T}^{i+1} \end{pmatrix} = \begin{pmatrix} \pi \| \, \sigma_*^i \| \, \mathcal{S}^{i+1} \\ \omega \| \, \omega \end{pmatrix} \| \begin{pmatrix} \pi \| \, \sigma_*^i \| \, \mathcal{S}^{i+1} \\ \mathcal{T}^{i+1} \end{pmatrix} \tag{4}$$

$$= \begin{pmatrix} \pi \| \, \sigma_*^i \\ \omega \| \, \omega \end{pmatrix} \| \begin{pmatrix} \sigma_*^i \; \| \mathcal{S}^{i+1} \\ \mathcal{T}^{i+1} \end{pmatrix}.$$

Now we take the third and the fourth row:

$$\begin{pmatrix} \sigma_*^{i-1} \| \, \pi \, \| \, \pi \\ \omega \quad \| \, \tau_*^i \| \, \omega \end{pmatrix} = \begin{pmatrix} \sigma_*^{i-1} \| \, \pi \, \| \, \pi \\ \omega \quad \| \, \tau_*^i \end{pmatrix} \| \begin{pmatrix} \sigma_*^{i-1} \| \, \pi \| \, \pi \\ \omega \end{pmatrix} \tag{5}$$

$$= \begin{pmatrix} \sigma_*^{i-1} \| \, \pi \\ \omega \quad \| \, \tau_*^i \end{pmatrix} \| \begin{pmatrix} \pi \\ \omega \end{pmatrix}.$$

We combine (4) and (5) to obtain the following, after applying Proposition 6.10 and Lemma 6.6 once more:

$$\mathcal{L} = \begin{pmatrix} \pi & \| \sigma_*^i \\ \omega & \| \omega \\ \sigma_*^{i-1} & \| \pi \\ \omega & \| \tau_*^i \end{pmatrix} \| \begin{pmatrix} \sigma_*^i \| \mathcal{S}^{i+1} \\ \mathcal{T}^{i+1} \\ \pi \\ \omega \end{pmatrix}. \tag{6}$$

Look at the right side of (3):

$$\mathcal{R} = \begin{pmatrix} \sigma_*^{i-1} \| \sigma_*^i \| \mathcal{S}^{i+1} \\ \omega \quad \| \tau_*^i \end{pmatrix} \| \begin{pmatrix} \sigma_*^{i-1} \| \sigma_*^i \| \mathcal{S}^{i+1} \\ \| \mathcal{T}^{i+1} \end{pmatrix} \tag{7}$$

$$= \begin{pmatrix} \sigma_*^{i-1} \| \sigma_*^i \\ \omega \quad \| \tau_*^i \end{pmatrix} \| \begin{pmatrix} \sigma_*^i \| \mathcal{S}^{i+1} \\ \| \mathcal{T}^{i+1} \end{pmatrix}$$

We see that the right parallel components in (6) and (7) are identical (as $\binom{\pi^{i+1}}{\omega^{i+1}}$ is a projection c-map $Id_{V \to V_{i+1}}$). To complete the proof we should prove the equality of the left components:

$$\begin{pmatrix} \pi \| \sigma_*^i \\ \omega \| \omega \\ \sigma_*^{i-1} \| \pi \\ \omega \| \tau_*^i \end{pmatrix} = \begin{pmatrix} \sigma_*^{i-1} \| \sigma_*^i \\ \omega \| \tau_*^i \end{pmatrix},$$

but this is exactly the equality (2) we have already proved at the beginning (with σ_*^i replaced by σ_*^{H+1} and τ_*^i by τ_*^{H+1}, but the argument remains the same). □

Note In the proof above we used only a very general property of relaxation, mainly that σ_*^r is independent from every stage different than r. This means that the proof applies to any other relaxed model fulfilling this requirement.

Theorem 9.2 *Let $\triangle^*$ be defined analogously to $\triangle$ (we replace σ,τ, by σ_*,τ_*, resp.). The analogue to Theorem 8.2 holds:*

$$\triangle^* ; \Gamma^*_{\mathrm{odd-even}} = \Gamma^*_{\mathrm{ripple}} ; \triangle^* = \Gamma^*_{\mathrm{sim}} ; \triangle^*.$$

Proof: We can repeat here the proof of Theorem 8.2, replacing σ by σ_*, since in the proof of Theorem 8.2 it is only assumed that σ is independent from certain parts of its domain, and σ_* also satisfies this requirement. □

And finally for the relaxed case by Theorem 9.1 and 9.2 we obtain, analogously as in Corollary 8.5 in Section 8:

Corollary 9.3 *The Markov chains $\Gamma^*_{\mathrm{ripple}}$ and Γ^*_{sim} have the same stationary distribution (say p) and the stationary distribution of $\Gamma^*_{\mathrm{odd-even}}$ is equal to $(p ; \triangle^*)$ (p is treated here as a c-map from a singleton set).*

10 Conclusions and open problems

We have presented the formal description of several models of packet flow in the butterfly MINs by means of ergodic Markov chains. By varying the definitions of c-maps σ and τ one can obtain the number of different models, without introducing any changes in the main definitions. For example the relaxed blocking model of [18] was easily obtained in this way. It is interesting to observe how the correspondence between ripple, odd-even and simultaneous changes when we modify σ and τ.

We considered butterfly MINs with (2×2)-switches having buffers of capacity one, but all the results can be easily repeated for $(k \times k)$-switches, $k \geq 3$. All the considerations can be repeated for buffers of greater capacity as well, assumed that bandwidth of connecting links is equal to this capacity (otherwise Theorem 7.4 and associated lemmas do not hold).

There still remains some work, for instance the verification of applicability of our framework in other fields. The approach seems to be quite general and it is reasonable to attempt to apply it for describing the behaviour of parallel systems. Another possible continuation of the work is to examine the usefulness of the presented notation in the analysis of the performance of full blocking MIN. Since many years this is a challenging problem, and by now there are no results on throughput in the full blocking MIN model with finite buffers. As there are asymptotic results on throughput in relaxed blocking model [18], the promising direction is to study relationships between relaxed and non-relaxed models, aiming at relating throughputs.

The only available way of estimating performance in full blocking model is by time consuming computer simulations. This motivates the question whether the Markov chains defined by us are *rapidly mixing* [10, 11], i.e. whether they converge quickly to their stationary distributions. The positive answer to this question would justify simulations and give effective polynomial time algorithms for approximating throughput and other efficiency measures.

References

[1] K. Batcher, *Sorting networks and their applications*, Proc. of the AFIPS Spring Joint Computing Conference, vol. 32, 1968, pp. 307-314.

[2] BBN, *Butterfly^{TM} Parallel Processor Overview*, BBN Report No. 6148, Version 1, Cambridge Mass. 1986.

[3] R. Diekmann, R. Lüling, B. Monien, *Communication Throughput of Interconnection Networks*, Proc. MFCS 1994, Springer LNCS 841, pp. 72-86.

[4] W. Feller, "An Introduction to Probability Theory and Its Applications", John Wiley & Sons, 1971.

[5] A. Gambin, S. Lasota, *On the Semantics of Multistage Interconnection Networks*, Proc. XXIII-rd Seminar on Current Trends in Theory and Practice of Informatics SOFSEM, 1996, LNCS 1175, pp. 359-368.

[6] G.F. Goke, G.J. Lipovski, *Banyan Network for Partitioning Multiprocessor Systems*, Proc. 1st Ann. Symp. on Computer Architecture, 1973, pp. 21-28.

[7] A. Gottlieb, *An overview of the NYU Ultracomputer Project*, in J.J. Dongarra, Experimental Parallel Computing Architectures, Elsevier Amsterdam, 1987, pp. 25-95.

[8] C.A.R. Hoare, "Communicating Sequential Processes," Prentice – Hall, 1985.

[9] Y.C. Jenq, *Performance analysis of a packet switch based on a single buffered banyan network*, IEEE Journal on Selected Areas of Comm., vol. SAC-3, 1983, pp. 1014-1021.

[10] M. R. Jerrum, A.J. Sinclair, *Approximate counting, uniform generation and rapidly mixing Markov chains.* Information and Computation 1989, 82, pp. 93-133.

[11] R. Kannan, *Markov chains and polynomial times algorithms*, Proc. 35th IEEE Symposium on Foundations of Computer Science 1994, 656-671.

[12] R.R. Koch, *Increasing the Size of a Network by a Constant Factor Can Increase Performance by More Than a Constant Factor*, Proc. 29th Ann. Symposium on Foundations of Computer Science 1988, 221-231.

[13] T.F. Leighton, "Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes," Morgan Kaufmann, 1992.

[14] Y. Mun, H. Yong, *Performance Analysis of Finite Buffered Multistage Interconnection Networks*, IEEE Trans. on Computers, vol. C-43, 1994, pp. 152-162.

[15] J.H. Patel, *Performance of processor-memory interconnections for multiprocessors*, IEEE Trans. on Computing, vol. C-30, 1981, pp. 771-780.

[16] G.F. Pfister, W.C. Brantley, D.A. George, S.L. Harvey, W.J. Kleinfelder, K.P. McAuliffe, E.A. Melton, V.A. Norton, J. Weiss, *Introduction to the IBM Research Parallel Processor Prototype (RP3)*, in J.J. Dongarra, Experimental Parallel Computing Architectures, Elsevier Amsterdam, 1987, pp. 123-140.

[17] A.G. Ranade, *How to emulate shared memory*, Proc. 28th Ann. Symposium on Foundations of Computer Science 1987, pp. 185-194.

[18] R. Rehrmann, B. Monien, R. Lüling, R. Diekmann, *On the Communication Throughput of Buffered Multistage Interconnection Networks*, Proc. 8th ACM Symposium on Parallel Algorithms and Architectures, 1996.

[19] R. Rooholaminiight, *Finding the Right ATM Switch for the Market*, IEEE Journal Comp., 1994, pp. 16-28.

[20] T.H. Theimer, E.P. Rathgeb, M.N. Huber, *Performance Analysis of Buffered Banyan Networks* IEEE Trans. on Communication, vol. C-39, 1991, pp. 269-277.

[21] L.G. Valiant, *A scheme for fast parallel communication*, SIAM Journal on Computing, vol. 11, no. 2, 1982, pp. 350-361.

[22] H.Y. Yoon, K.Y. Lee, M.T. Liu, *Performance Analysis of Multibuffered Packet-switching Networks in Multiprocessor Systems*, IEEE Trans. on Computers, vol. C-39, 1990, pp. 319-327.