

EbDa: A New Theory on Design and Verification of Deadlock-free Interconnection Networks

Masoumeh Ebrahimi* Masoud Daneshtalab*[†]

Royal Institute of Technology, Sweden* Mälardalen University, Sweden[†]

mebr@kth.se; masoud.daneshtalab@mdh.se

ABSTRACT

Freedom from deadlock is one of the most important issues when designing routing algorithms in on-chip/off-chip networks. Many works have been developed upon Dally's theory proving that a network is deadlock-free if there is no cyclic dependency on the channel dependency graph. However, finding such acyclic graph has been very challenging, which limits Dally's theory to networks with a low number of channels. In this paper, we introduce three theorems that directly lead to routing algorithms with an acyclic channel dependency graph. We also propose the partitioning methodology, enabling a design to reach the maximum adaptiveness for the n -dimensional mesh and k -ary n -cube topologies with any given number of channels. In addition, deadlock-free routing algorithms can be derived ranging from maximally fully adaptive routing down to deterministic routing. The proposed theorems can drastically remove the difficulties of designing deadlock-free routing algorithms.

CCS CONCEPTS

• Computer systems organization → Interconnection architectures;

KEYWORDS

Cyclic Dependencies, Deadlock-free Routing Algorithms, Interconnection Networks

ACM Reference format:

Masoumeh Ebrahimi* Masoud Daneshtalab*[†] Royal Institute of Technology, Sweden* Mälardalen University, Sweden[†]. 2017. EbDa: A New Theory on Design and Verification of Deadlock-free Interconnection Networks. In *Proceedings of ISCA '17, Toronto, ON, Canada, June 24-28, 2017*, 13 pages. <https://doi.org/10.1145/3079856.3080253>

1 INTRODUCTION

An interconnection network consists of a set of routers and links where a topology such as mesh, torus [20], k -ary n -cube [5] determines the arrangements of links and routers [4, 13, 38]. Deadlock may occur in the network due to a cyclic dependency between channels such that each packet holds a channel needed by another

packet [4, 13]. Virtual channels (VC) can be used to remove the cyclic dependency between channels and thus avoiding deadlock [6, 33, 35]. VCs are theoretically formed by splitting a physical channel and allocating a dedicated buffer to each of them. VCs can be also used to improve network performance and throughput through sharing resources and providing alternative paths to route packets [19, 29, 43, 48].

Turn models [18] are built upon Dally's theory, providing a way to check for cyclic dependencies. Based on turn models, there are two types of abstract cycles that can be formed in the network, known as clockwise and counterclockwise cycles. A cycle may lead to deadlock and thus it should be avoided. In turn models, certain turns are prohibited from each abstract cycle in order to break all cyclic dependencies among channels.

Packet switching techniques can be implemented in three ways as store-and-forward (SAF), virtual-cut-through (VCT) and wormhole (WH) [11, 16, 23, 31, 34]. In SAF [11, 16], the whole packet should be stored in an input buffer before proceeding to the next router. In VCT [11, 23, 32], the packet is forwarded to the next router as soon as there is enough space to accommodate the packet. Unlike SAF, the packet does not necessarily need to be stored in the input buffer before proceeding to the next router. In both approaches, the buffer size must be large enough to accommodate the longest possible packet in the network. In WH [31, 34], however, packets traverse in a pipelined fashion. No specific limitation is applied on the size of buffers, the number of packets, and the location of the header flit in the buffer. WH eliminates the need of large buffers in intermediate routers along the path. Since SAF and VCT are special cases of WH, the proof of deadlock freedom for WH is also valid for SAF and VCT.

Routing algorithms can be classified into *deterministic* and *adaptive* [4, 13, 36]. In deterministic routing a fixed path is used for each pair of source and destination routers. Adaptive routing algorithms can be further classified into *partially adaptive* and *fully adaptive*. In fully adaptive routing, packets are allowed to take any minimal paths available between a source and destination pair while in partially adaptive routing such adaptiveness is limited to fewer paths. In this paper, we also refer to *maximally fully/partially adaptive routing* when including U- and I-turns in the set of allowable turns [17]. U- and I-turns are defined in Section 3.2.

In this paper, we do *NOT* intend to introduce a new routing algorithm but instead we show a roadmap to design deadlock-free routing algorithms in a wormhole switching network. In fact, we introduce three theorems that altogether remove all cyclic dependencies on the channel dependency graph and directly suggest deadlock-free routing options. The combination of these theorems is called EbDa¹.

¹EbDa is derived from the first two letters of the authors' last names

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISCA '17, June 24-28, 2017, Toronto, ON, Canada

© 2017 Association for Computing Machinery.

ACM ISBN 978-1-4503-4892-8/17/06...\$15.00

<https://doi.org/10.1145/3079856.3080253>

We design an acyclic channel dependency graph in the network by an inspiration from a mathematical observation on the formation of cycles in a geometrical space. Based on this observation, the necessary condition to form a cycle is the availability of channels in both positive and negative directions. For example, a square or rectangular shape cannot be formed if any of the X^+ , X^- , Y^+ , and Y^- channels is missing. In short, the whole process of designing a deadlock-free routing algorithm is as simple as dividing channels into disjoint cycle-free partitions and then tracing the partitions in an ascending or descending order. Based on these theorems different deadlock-free routing algorithms can be designed or algorithms can be verified on their freedom from deadlock. Moreover, we introduce a systematic approach of extracting partitions and thus algorithms for any given number of channels. Derivations from these algorithms lead to various deadlock-free algorithms with different levels of adaptiveness. The entire process results in extracting deterministic, partially adaptive, fully adaptive and maximally adaptive routing algorithms. We do not put any limitation on the network (ir)regularity, dimension, and number of VCs.

The remainder of this paper is organized as follows. In Section 2, the related work is given. In Section 3, theorems are introduced. Section 4 describes designs with maximum adaptiveness and minimum number of channels. In Section 5, we propose methodologies for optimal partitioning. In Section 6, several case studies are investigated and finally we conclude the paper in the last section.

2 RELATED WORK

In the area of on-chip/off-chip interconnection networks, there are two main theories, proposed by William J. Dally in 1987 [7] and José Duato in 1993 [9]. Dally proposed a methodology and applied it to deterministic routing, showing that a necessary and sufficient condition to design a *deadlock-free routing algorithm* is to remove cyclic dependencies on the channel dependency graph [7]. The turn model concept, defined by Glass and Ni, has enabled Dally's theory to be adapted for adaptive routing [18]. Since then, turn models have been extensively utilized in designing both deterministic and adaptive routing algorithms [1, 3, 15, 21, 24, 26, 40, 42, 47]. Dally's theory does not put any restriction on the wormhole switching network.

On the other hand, based on Duato's theory a necessary and sufficient condition to design a *deadlock-free fully adaptive routing* is the existence of a cycle-free subset of channels. All the other channels can be used with no restrictions. In more details, two types of channels are used as adaptive and escape. Packets can use the adaptive channels in any order but in a case of blockage, packets are transferred to escape channels that are cycle-free [28]. Duato's theory is utilized to design many fully adaptive routing algorithms [12, 19, 25, 29, 41, 44]. This theory [9, 10] is valid under the assumption that an input buffer holds only the flits of *one* packet. This is to ensure that if the packet is blocked, the header flit is always in the head of the queue and thus the packet can be moved to the escape channels to avoid deadlock. This assumption poses a strong limitation on wormhole switching as an input buffer must be emptied before receiving the header of a new packet. This issue has been investigated in [30] arguing that the majority of packets are short in on-chip networks. Through modifying the VC allocator unit at the cost of more resources, multiple packets can be resided in an input

buffer. However, the input buffer should have enough space to hold the entire packet, implying the VCT technique.

On the differences between two theories, Dally's theory has been initially proposed for deterministic routing and it does not put any restriction on WH. Many works have adapted the theory to design adaptive routing [2, 18, 27]. But due to the complexity of finding acyclic channel dependency graph by applying turn models, the usage of adaptive routing based on this theory is limited to networks with a low number of channels [1, 3, 14, 21, 24, 26, 40]. By assuming two abstract cycles (each consists of four 90-degree turns) in a 2D network and removing one turn from each cycle, $16(4^2)$ different combinations should be examined to verify whether the design is deadlock-free or not [17]. By adding one VC per dimension, the number of combinations increases to $65,536(4^8)$. In the 3D network with no VC, the number of combinations is $29,696(4^6)$ while adding one VC per dimension increases this number to more than 8 billion combinations.

On the other hand, the focus of Duato's theory is on adaptive routing that is fairly scalable to different network sizes. However, Duato's theory strongly limits the wormhole switching technique as multiple packets cannot be resided in an input buffer.

The theories in this paper differ from both Duato and Dally theories. The key different aspects and assumptions between our theory and Duato's are as:

- (1) Duato's theory is based on cyclic adaptive channels and acyclic escape channels so that in the case of deadlock among adaptive channels, packets use cycle-free escape channels to avoid deadlock. But the concept behind our theory is the creation of an acyclic channel dependency graph where no escape channel is needed. Since packets can use all allowable turns simultaneously, a better distribution of packets among channels can be obtained.
- (2) In Assumption 3 of [9], it is stated that "A queue cannot contain flits belonging to different messages. A header flit will always occupy the head of a queue. If it is not satisfied, it is easy to define a deadlocked configuration which invalidates that theorem". However, our proposal is general and does not impose any limitations on the number of packets in a buffer and the packet header can be stored in any buffer slot.

On the differences between EbDa and Dally's theory:

- (1) Dally's theory defines conditions to verify whether a network is deadlock-free or not. Later on in [18], turn-models were introduced to make such verification easier. Contrarily, our theory show how to directly design an acyclic channel dependency graph given the available channels in the network.
- (2) Dally's theory is limited to small network sizes where it is feasible to check all possible channel dependencies. We solve the scalability limitations of Dally's theorem to networks with arbitrary large dimensions.

We argue that the theorems and methodologies in this paper open a new direction in the design of routing algorithms from maximally fully adaptive routing algorithms down to deterministic routing algorithms. In addition, the theorems and methodologies are general and

can be applied to a wide area of interconnection networks, covering both on-chip and off-chip networks.

3 ASSUMPTIONS, DEFINITIONS, AND THEOREMS

3.1 Assumptions

- **Assumption1:** A WH switching network is assumed while theorems can be applied to VCT and SAF as well.
- **Assumption2:** Packets can have arbitrary lengths.
- **Assumption3:** n -dimensional mesh and k -ary n -cube topologies are assumed where k and n can be arbitrarily large. Theorems are valid on both regular and irregular networks. As our future work, we investigate other topologies such as fat-tree, dragonflies and those of the Dodec network [49].
- **Assumption4:** Theorems are valid for any number of VCs from 0 to n on each dimension where n can be arbitrarily large.
- **Assumption5:** VCs are considered as disjoint channels. For example, $X1$ and $X2$ represent two VCs in the X dimension with no channel dependency between them. Note that the term “channel” is used to represent either a physical or virtual channel.

3.2 Definitions

- **Definition1:** A dimension has two directions, positive and negative. In an n -dimensional network, a positive and negative direction of the D dimension is called D^+ and D^- , respectively, while D^* represents both positive and negative directions of the D dimension where $\mathbf{D} = \{X, Y, \dots, D_n\}$. For example in Figure 1(a), the X dimension covers X^+ and X^- directions. Each of the X^+ and X^- directions is called a channel. These channels are disjoint with no dependency between them.
- **Definition2:** A partition covers a set of channels in an n -dimensional network where packets can take any channels inside the partition arbitrarily and repeatedly. As an example in Figure 1(b), a partition may cover the X^+, X^-, Y^+ , and Z^- channels in a 3D network where a packet can take any of these channels in any order. The terms arbitrarily and repeatedly represent the maximum movement of packets within a partition.
- **Definition3:** A D-pair is so-called completed if both the positive and negative directions of the D dimension exist inside a partition: $\mathbf{P} = \{D^+, D^-\}$. As an example, an X -pair is shown in Figure 1(c), covering both the X^+ and X^- channels. In addition, channels in opposite directions but with different VC numbers also form a pair. For example in Figure 1(d), $X2^+$ and $X1^-$ represent a complete X -pair where 2 and 1 refer to two different VCs along the X dimension.
- **Definition4:** An I-turn (or 0-degree turn) represents a transition from one channel to another along the same direction (Figure 1(e)). Obviously, I-turns are formed only when there are several channels in the same dimension, either physical or virtual.
- **Definition5:** A U-turn (or 180-degree turn) represents a transition from one channel to another in the opposite direction (Figure 1(f)).
- **Definition6:** Two partitions are disjoint if they do not share any common channel with each other. Independence of channels may have different forms, for example:

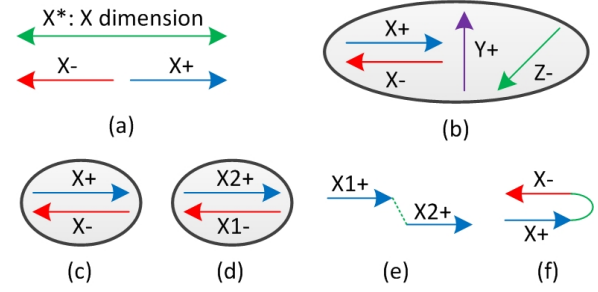


Figure 1: (a) The X dimension, represented by X^* , composed of two disjoint channels as X^+ and X^- ; (b) a partition covering the X^+, X^-, Y^+ , and Z^- channels; (c) an X -pair; (d) an X -pair with different VC numbers on positive and negative directions; (e) I-turn formed by a transition from $X1^+$ to $X2^+$; (f) U-turn formed by a transition from X^+ to X^- .

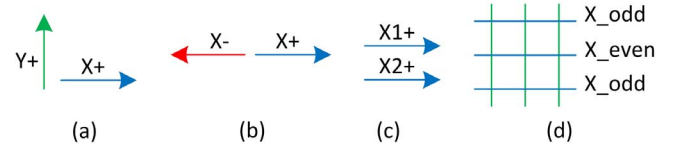


Figure 2: (a) Channels in different dimensions are disjoint such as X^+ and Y^+ ; (b) channels in opposite directions are disjoint such as X^+ and X^- ; (c) channels with different VC numbers are disjoint such as $X1^+$ and $X2^+$; (d) channels located in different rows are disjoint such as X_{even} and X_{odd} .

- Channels in different dimensions are disjoint such as X^+ and Y^+ in Figure 2(a).
- Opposite channels along the same dimension are disjoint such as X^+ and X^- in Figure 2(b).
- Channels with different VC numbers are disjoint such as $X1^+$ and $X2^+$ in Figure 2(c).
- Channels in different columns/rows are disjoint such as X_{even} and X_{odd} in Figure 2(d).

3.3 Theorems

- **Theorem1:** A partition is cycle-free if it covers at most one complete D -pair in an n -dimensional network where $\mathbf{D} = \{X, Y, \dots, D_n\}$ as long as no U- and I-turns are concerned.

We first prove the theorem for the base case of $k=2$ where k is the network dimension. Then, we show that the theorem is valid for any k value.

Base case statement $k=2$: A partition is cycle-free if it covers at most one complete D -pair in a 2-dimensional network.

- **Proof:** In a 2-dimensional network, the largest partition A covers four directions called X^+ (E), X^- (W), Y^+ (N), and Y^- (S). A necessary condition to form an XY -cycle is the existence of four channels and four 90-degree turns. We argue that a cycle cannot be formed by excluding any of these channels from a partition. In other words, a partition is cycle-free if it covers at most one complete D -pair (e.g. X -pair: X^+ and X^-) in a 2-dimensional network. Accordingly, the partition will be deadlock-free if it covers three channels in a 2D network, e.g. X^+, X^- , and Y^- .

Statement for $k=n$: A partition is cycle-free if it covers at most one complete D -pair in an n -dimensional network.

- **Proof:** To close a cycle, there should be at least two complete D-pairs within a partition otherwise a path cannot return to its initial point and a cycle cannot be closed. It is obvious that any arbitrary dimension A has an angle greater than zero from any other arbitrary dimension B ; otherwise dimension A and B represent the same dimension. A necessary condition to close a cycle inside a partition is the existence of both negative and positive directions in at least two dimensions. The necessary condition is broken if a partition covers at most one complete D-pair. In other words, a partition is cycle-free if it covers any number of channels from any dimension in an n -dimensional network but includes at most *one* complete D-pair. For example in a 4-dimensional network, if a partition covers the X^+ , Y^+ , Y^- , Z^+ , and T^- channels and all of them can be taken arbitrary and repeatedly, the partition is cycle-free because it contains only one complete D-pair, i.e. Y -pair. The other channels have some degrees from each other and thus the path cannot return to its initial point to close a cycle.

Note to Theorem1: VCs refer to disjoint channels with 0-degree angle from each other. For example, $X1$ and $X2$ can be considered as two disjoint channels, similar to two parallel lines that never reach each other. Positive and negative directions of the same dimension but with different VC numbers represent one D-pair. For example the partition $P = \{X1^+ X2^- Y1^+ Y2^-\}$ is not cycle-free as it covers two complete D-pairs; one pair along the X dimension: $(X1^+, X2^-)$ and one along the Y dimension: $(Y1^+, Y2^-)$. On the other hand, the partition $P = \{X1^+ Y1^+ Y1^- Y2^+ Y2^-\}$ is cycle-free as it contains one D-pair (i.e. Y -pair) regardless of the number of Y -pairs that can be formed. U- and I-turns will be discussed in Theorem2.

Note to Theorem1: The maximum number of channels that can be grouped inside a partition is $n + 1$ in an n -dimensional network when no redundancy is taken into account (i.e. no VC is considered). Out of $n + 1$ channels, only two channels belong to the same dimension, representing a D-pair.

Corollary of Theorem1: Any sub-partition of a cycle-free partition is also cycle-free.

Corollary of Theorem1: Considering several disjoint partitions, as long as packets are limited to use the channels of their partitions, no new cycle can be formed, and the network remains deadlock-free.

Example of Theorem1: To form a cycle in an XY plane, it is necessary to take all X^+ , X^- , Y^+ , and Y^- directions. In other words, if only one direction is not taken by packets, deadlock cannot be formed. For example as shown in Figure 3, there is no possibility of deadlock if packets cannot be forwarded to the north but all the other directions arbitrarily and repeatedly. The combination of three channels in one partition enables four allowable 90-degree turns as WS , SE , ES , and SW .

- **Theorem2:** A partition is cycle-free if one U-turn is allowed per complete pair, taken in an ascending order.

In Theorem1, we argued that only one complete D-pair is allowed inside a partition to keep the partition deadlock-free. By satisfying the conditions of Theorem1, all channels inside a partition can be taken in any order as long as no U- or I-turn

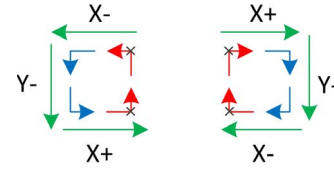


Figure 3: A missing direction breaks the cycle in a partition. The formed turns by three available channels X^+ , X^- , Y^- are as WS , SE , ES , and SW .

are concerned. Using Theorem2, some U- and I-turns are also allowed without forming cycles.

Base case statement $k=1$: A partition is cycle-free if one U-turn is allowed per complete pair where k stands for the number of complete pairs.

- **Proof:** One complete D-pair consists of two opposite directions along the D dimension, so-called Xa^+ and Xb^- where a and b refer to VC numbers that could be the same or different. To close a cycle within a partition at least two channels X^+ and X^- and two U-turns are necessary. By removing one U-turn from a partition, the necessary condition to form a cycle does not meet. In other words, permitting one U-turn and prohibiting another U-turn prevent forming a cycle. The allowed U-turn can be used in combination with other channels with no restriction.

Base case statement $k=n$: A partition is cycle-free if one U-turn is allowed per complete pair where the number of complete pairs is n .

- **Proof:** The proof is similar to that of Up*/Down* routing [40] where no cycle is introduced when channels are taken in a strictly ascending order. If channels of all complete pairs are numbered from 1 to $2K$, and traced in an ascending order (or descending order), the formed U-turns do not lead to deadlock.

Note to Theorem2: Enabling U-turns is essentially important in fault-tolerant designs or where rerouting brings an advantage (see [22]). In addition, topologies with wrap-around channels can utilize U-turns. For example in a Torus topology, each wrap-around channel in the X dimension can be seen as two unidirectional channels (X^+ and X^-) and two U-turns.

Corollary of Theorem2: Similar to the proof of U-turns, I-turns can be taken in an ascending order without introducing a cycle. This is the case when a complete pair is presented along the D dimension. In the other dimensions, all I-turns inside the partition are allowed without creating cycles. This is due to the fact that a dimension with a missing positive or negative direction cannot directly contribute in closing a cycle and thus all the transactions in the single direction is permissible. In short I-turns should be taken in an ascending order if a complete pair is presented along the D dimension. Otherwise, all I-turns within a partition are allowed.

Example of Theorem2: Figure 4(a) shows an example when three VCs are presented along the Y dimension. Channels are first numbered from 1 to 6 in an arbitrary order and then U- and I-turns can be extracted by tracing channels in an ascending order e.g. the first channel has a transition to any of the other five channels and the second channel has a transition to the next four channels

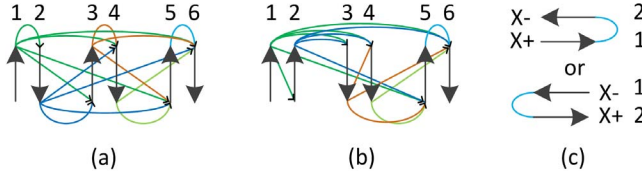


Figure 4: (a) The formed U- and I-turns by the availability of three VCs inside a partition; (b) an alternative way of channel arrangement; (c) the formed U-turns by the existence of a complete pair inside a partition, from which one can be selectively chosen.

and so on. Thereby, the total number of U- and I-turns can be calculated by $n(n-1)/2$ where n is the number of channels. It means that half of U- and I-turns are permitted. In this example, nine U- and six I-turns are formed. The number of U-turns can be measured by $a \times b$ while the number of I-turns can be calculated by $\binom{a}{2} + \binom{b}{2}$. a and b refer to the number of channels in the positive and negative directions, respectively, where $n = a + b$. It can be easily shown that:

$$\frac{n(n-1)}{2} = ab + \frac{a!}{2(a-2)!} + \frac{b!}{2(b-2)!}$$

As illustrated in Figure 4(b), channels can be sorted in a different order, but still resulting in nine U- and six I-turns. In the example of Theorem1 where $\mathbf{P} = \{X^+ X^- Y^+\}$, the complete D-pair is presented along the X dimension. Thereby, the X^+ and X^- channels are numbered and traced in an ascending order, shown in Figure 4(c). Based on the numbering principle, either the U-turn from X^+ to X^- or from X^- to X^+ is allowable.

- **Theorem3:** Transitions between disjoint acyclic partitions in a consecutive order do not form a cycle.

Theorem1 defines the conditions to form a cycle-free partition without considering U- and I-turns. A set of 90-degree turns can be extracted by enabling all combinations of channels inside a partition. Theorem2 defines the conditions to include U- and I-turns to the set of allowable turns. Theorem3 adds new 90-degree, U- and I-turns to the set of permissible turns. This is obtained by allowing packets to use the channels of other partitions in an ascending order. We prove this theorem using mathematical induction. We first prove that the statement holds for the base case of $k = 2$ where k is the number of partitions. Then, we show that the theorem is valid for any number of partitions.

Base case statement $k=2$: Transitions between two disjoint acyclic partitions in a consecutive order do not form a cycle.

- **Proof:** Let us assume that channels are divided into two disjoint acyclic partitions as PA and PB . A selective combination of channels within each partition can be modelled as a connection of straight lines in an acyclic form. The lines representative of different partitions do not overlap as partitions are completely disjoint. By transition between partitions, the lines will be connected in one end. The other end cannot be connected as after using any channels of PB , the channels of PA cannot be used any longer. Since transitions between partitions result in a longer acyclic line, the statement is valid for two partitions.

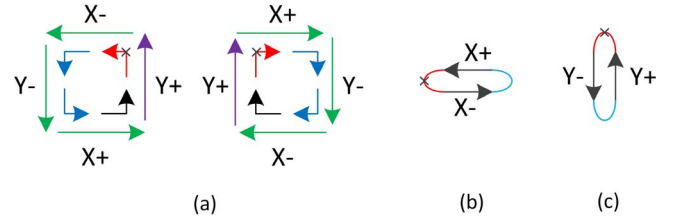


Figure 5: (a) Taking all directions is not a sufficient condition to form a cycle; the NW and NE turns cannot be taken as the transition from PB to PA is prohibited (b) one allowed U-turn in the X dimension from X^- to X^+ ; (c) one allowed U-turn in the Y dimension from Y^- to Y^+ .

Inductive step: Transitions between n number of disjoint acyclic partitions in a consecutive order do not form a cycle.

- **Proof:** According to the base case where $k = 2$, transition from one partition to the next one can be modelled as an acyclic line. Now the transition can advance to the third partition, forming a longer acyclic line as there is no dependency from the third partition to any of the first and second partitions. It means that when the statement holds for $k = n$, it also holds for $k = n + 1$ and the theorem is proved.

Corollary of Theorem3: Transitions between partitions can be done in any ascending order.

Corollary of Theorem3: Transitions enable new U- and I-turns where the channels representing the same dimension are located in different partitions.

Example of Theorem3: Let us assume a 2D network with four channels as X^+ , X^- , Y^+ , and Y^- . The channels can be divided into two partitions where the first partition is $PA = \{X^+ X^- Y^-\}$ and the second partition is $PB = \{Y^+\}$. Based on Theorem1, both partitions are deadlock-free as they cover at most one complete D-pair. The allowable 90-degree turns and U-turns in PA are illustrated in blue in Figure 5(a) and Figure 5(b), respectively. Theorem3 allows transitions from one partition to another in an ascending order. Assuming that this transition is made from PA to PB , new 90-degree turns are formed as EN and WN , shown in black in Figure 5(a). In addition, this transition enables a U-turn from Y^- in PA to Y^+ in PB (Figure 5(c)). The U-turn from Y^+ to Y^- is naturally avoided as no transition from PB to PA is possible. As shown in Figure 5(a), after transition from PA to PB , all directions are used but this does not lead to deadlock as a loop is not yet closed until a channel from PA is taken again. As was stated, the transition is only made from PA to PB and there is no possibility of a closed loop. In simple words, taking all directions is not a sufficient condition for the possibility of deadlock. It is worth noting that the obtained 90-degree turns (Figure 5(a)) by applying Theorem1 and Theorem3 lead to the north-last routing algorithm [18]. The allowable U-turns by Theorem2 and Theorem3 are additional turns that can be taken without forming a cycle.

4 MAXIMUM ADAPTIVENESS WITH MINIMUM NUMBER OF CHANNELS

Four channels in a 2D network can be divided into disjoint partitions in different ways, each leads to a new deadlock-free routing algorithm. Among the others, partitioning can be done in the following forms:

$$\mathbf{P1} = \{PA[X^+] \rightarrow PB[X^-] \rightarrow PC[Y^+] \rightarrow PD[Y^-]\}$$

$$\mathbf{P2} = \{PA[Y^-] \rightarrow PB[X^-] \rightarrow PC[Y^+X^+]\}$$

$$\mathbf{P3} = \{PA[X^-] \rightarrow PB[X^+Y^+Y^-]\}$$

$$\mathbf{P4} = \{PA[X^-Y^-] \rightarrow PB[X^+Y^+]\}$$

$$\mathbf{P5} = \{PA[X^-] \rightarrow PB[X^+Y1^+Y1^-Y2^+Y2^-]\}$$

Considering P1, transitions from (PA to PC), (PA to PD), (PB to PC), and (PB to PD) enable packets to reach the NE , SE , SW , and NW regions, respectively. All the formed turns are shown in Figure 6(a), defining the XY routing algorithm. The partitioning form of P2 (Figure 6(b)) offers a degree of adaptiveness to route packets. The channels covered by PC can be taken in any order, leading to a fully adaptive routing in the NE region. A deterministic routing is applied to the SE , SW , and NW regions as the channel of PA should be taken earlier than the channels of PB and PC . In general, channels grouped into a partition can be translated as a fully adaptive routing for the region they cover. This is due to the fact that channels can be taken in any order inside a partition, creating all possible turns in that region. In Figure 6(c), channels are divided into two partitions where one partition covers one channel and another partition covers three channels. The combination of channels inside PB and the transition from PA to PB lead to the same turns as the west-first routing algorithm [18]. In Figure 6(d), channels are still divided into two partitions, but each covers two channels. By this assignment, fully adaptive routing is provided for two regions. The formed turns by the partitioning strategy of P4 is the same as those of the negative-first routing algorithm [18]. The number of partitions cannot be reduced to one as by placing all channels in one partition, two complete D-pairs will be presented in the partition that violates Theorem1. Both P3 and P4 divide all channels into the minimum number of disjoint partitions, leading to the maximum number of allowable turns, i.e. six 90-degree turns and two U-turns. In other words, dividing channels into a minimum number of partitions leads to the maximum adaptiveness. As shown in Figure 6(e), adding VCs in PB does not enhance the adaptiveness in minimal paths but increases the number of identical turns as well as U- and I-turns.

The maximum number of channels that can be grouped inside a partition is $n + 1$ in an n -dimensional network where channels are belonging to different dimensions except the dimension with a complete pair. Adding more channels into the partition either violates Theorem1 or does not increase the adaptiveness. Thereby the objective of designing a fully adaptive routing in an n -dimensional network with the minimum number of channels is to group channels into a minimum number of disjoint partitions. Each partition provides the maximum adaptiveness for the region it covers.

With this introduction, we prove that the minimum number of channels to provide a fully adaptive routing in an n -dimensional

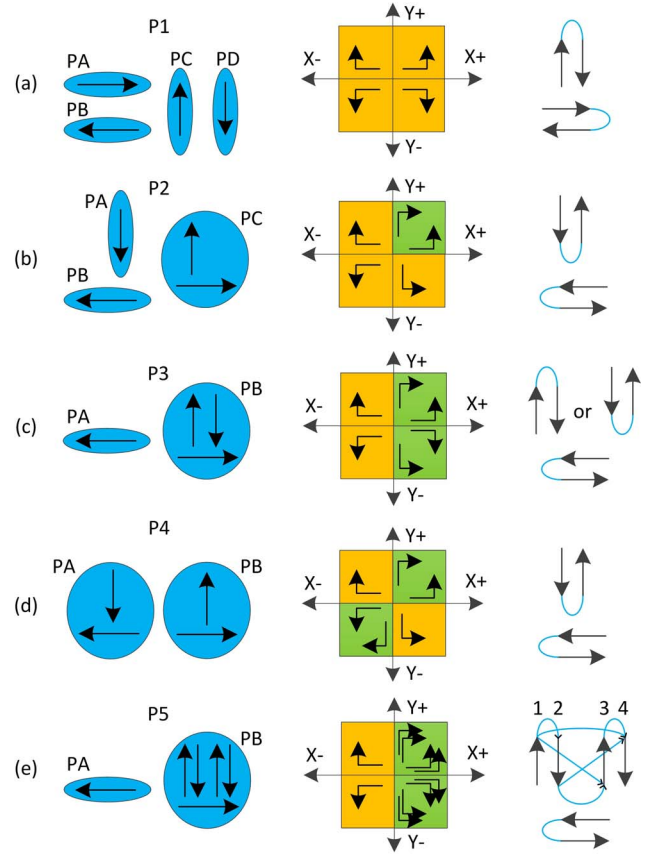


Figure 6: Different partitioning strategies leading to (a) XY -routing; (b) partially adaptive routing; (c) West-First routing; (d) Negative-First routing; (e) VCs do not enhance adaptiveness when presented inside the same partition.

network can be calculated by:

$$N_{Channel} = (n + 1) \times 2^{(n-1)}$$

where n is the network dimension.

Base case statement $n=2$: The minimum number of channels to provide a fully adaptive routing in a 2D network is $N = 6$.

- **Proof:** Assuming that a dimension divides a geometrical space into two regions, 1, 2, 3, and n dimensional network covers 2^1 , 2^2 , 2^3 , and 2^n regions, respectively. In a 2D network there are four regions as NE , NW , SE , and SW . The simplest way to design a fully adaptive routing is to consider one partition per region and assign dedicated channels to each partition to make them disjoint from each other. As shown in Figure 7(a), one possible partitioning is to allocate $Y1^+$ and $X1^+$ channels to the NE region; $Y1^-$ and $X2^+$ channels to the SE region; $Y2^-$ and $X2^-$ channels to the SW region and finally $Y2^+$ and $X1^-$ channels to the NW region. Based on this assignment, the set of partitions can be written as:

$$\mathbf{P} = \{PA[X1^+Y1^+]; PB[X2^+Y1^-]; PC[X2^-Y2^-]; PD[X1^-Y2^+]\}.$$

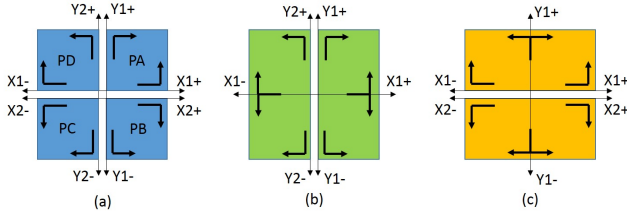


Figure 7: Four regions are divided into (a) four partitions, requiring 2 and 2 VCs along both dimensions; (b) two partitions, requiring 1 and 2 VCs along the X and Y dimensions, respectively; (c) two partitions, requiring 2 and 1 VCs along the X and Y dimensions, respectively.

All partitions are disjoint from each other as they cover channels with different directions or VC numbers. By this channel allocation, a fully adaptive routing is provided in all regions of the network and in total four channels are required (i.e. two channels along the X and two channels along the Y dimension). However, the number of channels can be reduced by applying Theorem 1. Based on Theorem 1, each partition can cover $n+1$ channels, including one complete pair. In a 2D network, maximum three channels can be grouped into a partition while more channels will not increase the adaptiveness. This implies that the number of partitions can be reduced to two. One possible partitioning strategy is $\mathbf{P1} = \{PA[X1^+ Y1^+ Y1^-]; PB[X1^- Y2^+ Y2^-]\}$, shown in Figure 7(b), suggesting the same routing algorithm as DyXY [24]. Another strategy might be $\mathbf{P2} = \{PA[X1^+ X1^- Y1^+]; PB[X2^+ X2^- Y1^-]\}$, illustrated in Figure 7(c). The number of partitions cannot be reduced any further as each partition covers the maximum number of channels according to Theorem 1. Therefore, the minimum number of channels to provide a fully adaptive routing in a 2D network is 6 and the statement is valid.

Base case statement $n=3$: The minimum number of channels to provide a fully adaptive routing in a 3D network is $N = 16$.

- **Proof:** A 3D network can be divided into eight regions as NEU , NWU , SEU , SWU , NED , NWD , SED , and SWD and each region is covered by one partition. One way of assigning channels to partitions is shown in Figure 9(a):

$$\begin{aligned} \mathbf{P} = \{ & PA[X1^+ Y1^+ Z1^+]; PB[X1^- Y2^+ Z4^+]; \\ & PC[X2^+ Y1^- Z2^+]; PD[X2^- Y2^- Z3^+]; \\ & PE[X3^+ Y3^+ Z1^-]; PF[X3^- Y4^+ Z4^-]; \\ & PG[X4^- Y4^- Z3^-]; PH[X4^+ Y3^- Z2^-] \}. \end{aligned}$$

By this assignment, 24 channels are required. To reduce the number of partitions, we take advantage of Theorem 1 by including a complete pair inside a partition and thus allowing to merge each two regions. For example two neighbouring regions of NEU and NED can be merged and cover a bigger region where the corresponding partition will be $\mathbf{PA} = \{X1^+ Y1^+ Z1^+ Z1^-\}$. This partition is deadlock-free and since it covers the maximum number of channels (i.e. four channels in a 3D network), adding more channels does not increase adaptiveness. The remaining partitions can be

formed in a similar way, and thus the number of partitions is reduced to four and the number of channels to 16. As illustrated in Figure 9(b), 2, 2, and 4 virtual channels are required along the X , Y , and Z dimensions, respectively. An alternative partitioning strategy, shown in Figure 9(c), leads to 3, 2, and 3 virtual channels along the X , Y , and Z dimensions, respectively. A fully adaptive routing cannot be provided with a lower number of channels than 16 as it leads to two complete pairs inside a partition, violating Theorem 1. Taking the partitioning strategy of Figure 9(b), all the formed turns by applying Theorem 1, 2 and 3 are shown in Figure 8. For simplicity, $X1^+$, $X1^-$, $Y1^+$, $Y1^-$, $Z1^+$, and $Z1^-$ are replaced with E , W , N , S , U , and D , respectively, so that for instance a turn from $Y2^+$ to $X1^-$ is represented by $N2W1$. All these turns can be taken simultaneously without forming a cycle. This is the maximum amount of turns that offers a deadlock-free network while adding any more turn creates the possibility of deadlock.

Base case statement n : The minimum number of channels to provide a fully adaptive routing in an n -dimensional network is $N = (n+1) \times 2^{(n-1)}$.

- **Proof:** An n -dimensional space can be divided into 2^n regions with one partition allocated to one region. Each two neighbouring regions can be merged and thus the number of partitions is reduced to 2^{n-1} . On the other hand, each partition can cover $(n+1)$ channels to provide a fully adaptive routing. The number of partitions cannot be reduced any further and the degree of adaptiveness cannot be increased inside a partition. Therefore, the minimum number of channels to ensure a fully adaptive routing in an n -dimensional network is $N = (n+1) \times 2^{(n-1)}$.

5 OPTIMAL PARTITIONING STRATEGY

So far we learned that the whole process of designing a deadlock-free routing algorithm is as simple as dividing channels into disjoint partitions and utilizing channels with no limitation inside a partition. In addition, transitions are allowed between partitions in a consecutive order. In this section, we define a systematic way to extract deadlock-free routing algorithms for the given number of channels from maximally adaptive routing algorithms down to deterministic routing algorithms. As was discussed in Section 4, by placing the maximum number of channels in a partition, the maximum adaptiveness is guaranteed for that region of space. Conversely, breaking a partition into multiple partitions reduces the degree of adaptiveness. For example placing channels in three partitions in Figure 6(b) leads to higher adaptiveness than placing them in four partitions in Figure 6(a)). Therefore, to reach the maximum adaptiveness for the given number of channels, the objective is to divide the channels into a minimum number of partitions. Since the number of deadlock-free routing algorithms can be relatively large, in Subsection 5.1 we propose strategies for arranging dimensions in specific orders and then extracting partitions from them (Subsection 5.2 and Subsection 5.3). Finally in Subsection 5.4 we discuss the overhead of developing routing algorithms.

Let us first explain the procedure with a simple example where the goal is to design a routing algorithm in a network with 3, 2, and

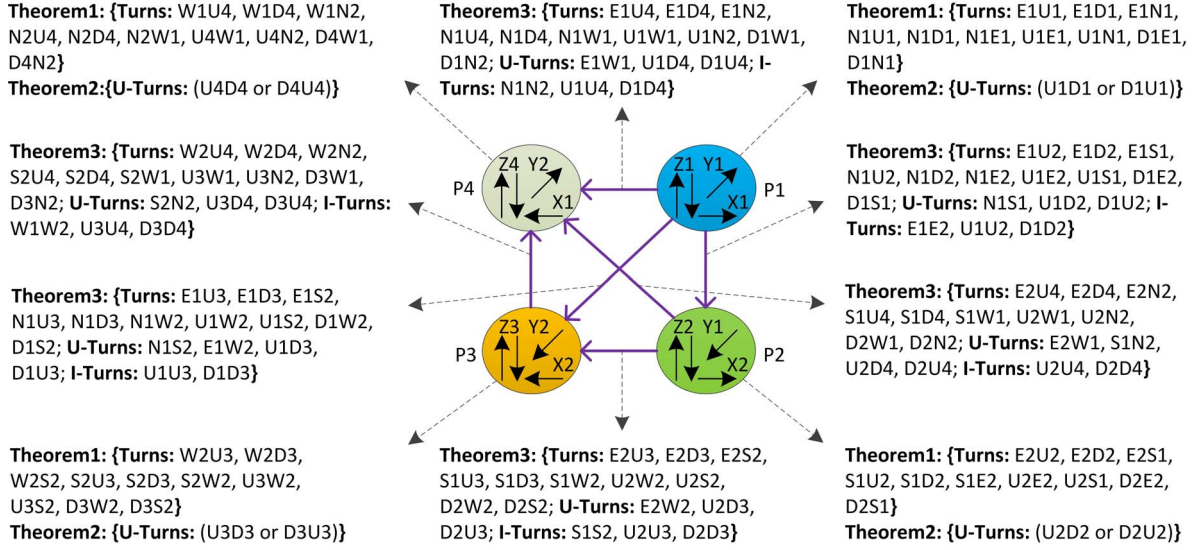


Figure 8: Extracting turns using Theorem1, 2, and 3.

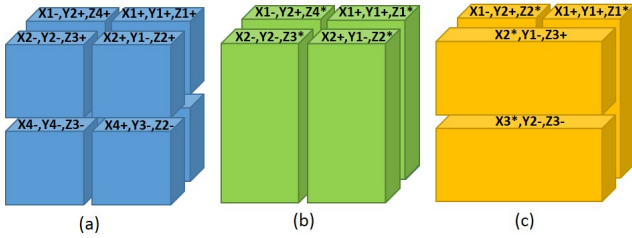


Figure 9: Eight regions are divided into (a) eight partitions, requiring 24 bidirectional channels; (b)-(c) four partitions, requiring 16 bidirectional channels.

3 virtual channels along the X , Y , and Z dimensions, respectively. We order the dimensions based on the number of D-pairs they cover. In this example both Z and X dimensions cover three D-pairs while the Y dimension has only two D-pairs and thus either the Z or X dimension is placed first. In this example we choose the Z dimension as Set1.

$$\begin{aligned}\text{Set1} : D_Z &= \{Z1^+ Z1^- Z2^+ Z2^- Z3^+ Z3^-\} \\ \text{Set2} : D_X &= \{X1^+ X1^- X2^+ X2^- X3^+ X3^-\} \\ \text{Set3} : D_Y &= \{Y1^+ Y1^- Y2^+ Y2^-\}\end{aligned}$$

The first partition forms by placing one Z -pair and one channel from each of the X and Y dimensions so that $\mathbf{PA} = \{Z1^+ Z1^- X1^+ Y1^+\}$, covering the NEU and NED regions. After assigning channels to a partition, they are removed from the sets and we get:

$$\begin{aligned}\text{Set1} : D_Z &= \{Z2^+ Z2^- Z3^+ Z3^-\} \\ \text{Set2} : D_X &= \{X1^- X2^+ X2^- X3^+ X3^-\} \\ \text{Set3} : D_Y &= \{Y1^- Y2^+ Y2^-\}\end{aligned}$$

Since each of the Z and X covers two D-pairs, no reordering is needed. By this arrangement, the second partition forms by placing one Z -pair and one channel from each of the X and Y dimensions,

and thus we get $\mathbf{PB} = \{Z2^+ Z2^- X1^- Y2^+\}$. We select $Y2^+$ instead of $Y2^-$ to cover the neighbouring regions of PA (NWU and NWD). The remaining channels will be:

$$\begin{aligned}\text{Set1} : D_X &= \{X2^+ X2^- X3^+ X3^-\} \\ \text{Set2} : D_Z &= \{Z3^+ Z3^-\} \\ \text{Set3} : D_Y &= \{Y1^- Y2^-\}\end{aligned}$$

Now the X dimension covers two D-pairs while the Z dimension has only one D-pair, so Set1 and Set2 are reordered. The third partition forms by placing one X -pair and one channel from each of the Z and Y dimensions so $\mathbf{PC} = \{X2^+ X2^- Z3^+ Y1^-\}$. The remaining channels will be:

$$\begin{aligned}\text{Set1} : D_X &= \{X3^+ X3^-\} \\ \text{Set2} : D_Z &= \{Z3^-\} \\ \text{Set3} : D_Y &= \{Y2^-\}\end{aligned}$$

Finally, the last partition is $\mathbf{PD} = \{X3^+ X3^- Z3^- Y2^-\}$. Since all channels are assigned to the partitions, the procedure terminates. The set of partitions will be similar to Figure 9(c) as:

$$\begin{aligned}\mathbf{P} &= \{\mathbf{PA}[Z1^+ X1^+ Y1^+]; \mathbf{PB}[Z2^+ X1^- Y2^+]; \\ &\quad \mathbf{PC}[X2^+ Z3^+ Y1^-]; \mathbf{PD}[X3^+ Z3^- Y2^-];\}\end{aligned}$$

It is obvious that set arrangement affects the resulted partitioning option and the number of formed partitions. Also, changing the order of channels inside a set results in a different channel partitioning. By this introduction, we describe three main steps that can be taken to reach the routing algorithms with different levels of adaptiveness. The first step (Subsection 5.1) defines the set arrangements from which partitions can be extracted. The second step (Subsection 5.2) explains how partitions can be formed using the defined sets. The third step (Subsection 5.3) describes the ways to generate new partitions, directing the algorithms with different degrees of adaptiveness.

5.1 Arranging sets

We define different set arrangements that each contributes to a unique set of partitions. The idea behind this arrangement is that partitions with similar D-pairs lead to symmetric algorithms while different D-pairs may lead to non-symmetric ones.

- **Arrangement 1:** Sets should be ordered based on the number of D-pairs they cover. Assuming that $q > s > \dots > r$, the sets are ordered as:

$$\text{Set1} : D_Y = \{Y1^+Y1^-, \dots Yq^+Yq^-\}$$

$$\text{Set2} : D_T = \{T1^+T1^-, \dots Ts^+Ts^-\}$$

...

$$\text{Setn} : D_X = \{X1^+X1^-, \dots Xr^+Xr^-\}$$

This set arrangement will be used in Algorithm 1 to extract the partitions.

- **Arrangement 2:** If one or multiple sets have the same number of channels as Set1, the sets can be reordered and thus partitions with different D-pairs can be formed. For example if $q > s > \dots > r$, the Set1 and Set2 can be replaced.
- **Arrangement 3:** If Set1 includes VCs, different D-pairs can be defined that affects the partitioning option. For example the channels inside Set1 can be ordered as **Set1** = $\{Y1^+Y1^-, Y2^+Y2^-\}$ or **Set1** = $\{Y2^+Y1^-, Y1^+Y2^-\}$. Consequently, D-pairs can be reorganized in $q!$ ways where q is the number of VCs. Each of these set arrangements will be an input to Algorithm 1.

5.2 Extracting partitions from the arranged sets

5.2.1 Main procedure. Each arranged set is used to extract the set of partitions. The first partition takes the first D-pair from Set1 and the first channel from each of Set2 to Setn. Channels are removed from the sets, and sets are reordered if necessary. Then the second partition is formed by covering the D-pair from Set1 and the second channel from each of Set2 to Setn. This procedure is repeated until all sets are empty. Depending on the given number of channels, last partition(s) may include a lower number of channels than the rest of partitions. If the region covered by these small partitions are a subset of other partitions, they should be merged with the existing ones in order to keep the number of partitions low. The pseudo code of the partitioning procedure is given in Algorithm 1.

Algorithm 1 Partitioning Procedure

```

1: Procedure Partitioning (Set1, Set2, ..., Setn, i){
2:   if (All sets are empty) then
3:     Merge matching partitions and exit;
4:   else
5:      $P_i = \{(\text{Set1}[1]\text{Set1}[2]); \text{Set2}[1]; \dots \text{Setn}[1]\};$ 
6:     Set1 is pair-wise left-shifted;
7:     Set2 to Setn are channel-wise left-shifted;
8:     Sets are reordered if necessary;
9:     CALL Partitioning (Set1, Set2, ..., Setn, i + 1);
10:  end if; }
```

5.2.2 An exceptional case. When no VC is presented, a new partitioning set can be formed by dividing channels into two partitions

where none of them cover a complete pair. For example, positive channels are placed in one partition and negative channels are placed in the second partition. This assignment neither violates Theorem1 nor increases the number of partitions than the minimum of two. By the availability of VCs, this approach leads to unnecessary identical turns and thus limits the adaptiveness. By exchanging channels between these two partitions, all alternative sets can be derived. The total number of combinations is 2^n where n is the number of partitions. For example in a 3D network with no VC, channels can be divided into two partitions with no partition covering a complete pair. Two partitions can be formed by placing one channel per dimension in PA and the rest of channels in PB. Assuming the following set arrangement:

$$\text{Set1} : D_X = \{X1^+X1^-\}$$

$$\text{Set2} : D_Y = \{Y1^+Y1^-\}$$

$$\text{Set3} : D_Z = \{Z1^+Z1^-\}$$

There are eight partitioning options in total, from which four are listed as:

$$P1 = \{PA[X1^+Y1^+Z1^+] \rightarrow PB[X1^-Y1^-Z1^-]\}$$

$$P2 = \{PA[X1^+Y1^+Z1^-] \rightarrow PB[X1^-Y1^-Z1^+]\}$$

$$P3 = \{PA[X1^+Y1^-Z1^+] \rightarrow PB[X1^-Y1^+Z1^-]\}$$

$$P4 = \{PA[X1^+Y1^-Z1^-] \rightarrow PB[X1^-Y1^+Z1^+]\}$$

The remaining four partitioning options can be obtained by switching from PBs to PAs:

5.3 Alternative partitioning options

By rearranging channels inside the sets, increasing the number of partitions, and tracing the partitions in different consecutive orders, various partitioning options can be derived.

5.3.1 Reordering channels inside the sets. Considering a set arrangement, by reordering channels inside the sets, various algorithms can be obtained. Algorithm 2 represents such strategy without violating Theorem1. Based on this algorithm, Set1 is circularly shifted left by two and other sets are circularly shifted left by one. As a result, new sets of partitions are formed by applying the partitioning procedure (Algorithm 1).

Algorithm 2 Derivation

```

1: for i=1 to q do
2:   for j=1 to s do
3:     ...
4:     for k=1 to n-1 do
5:       CALL Partitioning (Set1, Set2, ..., Setn, 1);
6:       Set(n-1) is channel-wise left-circular-shifted
7:     end for;
8:     ...
9:     Set2 is channel-wise left-circular-shifted;
10:  end for;
11:  Set1 is pairwise left-circular-shifted;
12: end for;
```

5.3.2 Increasing the number of partitions. By increasing the number of partitions, other routing algorithms can be derived and the degree of adaptiveness is reduced accordingly. If all partitions cover only one channel, a deterministic routing is suggested.

5.3.3 Tracing partitions in different orders. New routing algorithms can be designed by tracing partitions in other consecutive orders. For example, different turns are formed by a transition from PA to PB rather than from PB to PA .

It should be noted that depending on the application and design objectives, one partitioning strategy may be preferred over others. Although we described the partitioning strategy to extract different deadlock-free routing algorithms but a proper partitioning set cannot be strictly defined and varies over the context and the targeted application [8, 37, 45].

5.4 Routing algorithm

As was illustrated in Figure 8, all allowable 0-degree, U- and I-turns can be extracted from the partitions and the routing algorithm can be developed based on them. These turns are already supported in the current architectures and thus there is no need to modify the control units except the routing unit. In other words, the EbDa theory simplifies the procedure of finding compatible turns that otherwise might be found by heuristic methods (e.g. trial and error). In the example of Figure 8, a router architecture with 2, 2, and 4 VCs along the X , Y , and Z dimensions can be adapted for a various number of turns (e.g. 20 vs. 50 turns) while the router architecture in all cases remains nearly the same. Regarding the routing unit, turns can be expressed by some *if-else* statements [13]. Considering the XY routing algorithm, only one turn is allowed in the NE region. The routing algorithm for this region can be written as:

```

if  $X_{offset} > 0$  and  $Y_{offset} > 0$  then
  if  $X_{offset} > 0$  then  $Channel \leftarrow E$ ;
  else  $Channel \leftarrow N$ ;
end if;
end if;

```

where X_{offset} and Y_{offset} represent the distance to the destination in the X and Y dimensions, respectively. Now assuming a fully adaptive routing for the NE region, two turns are allowed as the EN and NE turns. The routing algorithm can be written as:

```

if  $X_{offset} > 0$  and  $Y_{offset} > 0$  then  $Channel \leftarrow E$  or  $N$ ;
else if  $X_{offset} > 0$  then  $Channel \leftarrow E$ ;
else  $Channel \leftarrow N$ ;
end if;

```

The implementation of these two algorithms shows that more allowable turns do not necessarily lead to a larger overhead or a more complex routing algorithm. Adding turns may simplify (combining *if-else* statements) or complicate (adding *if-else* statements) the routing logic.

6 CASE STUDIES

6.1 From maximum adaptiveness down to deterministic routing

In this section, we explain the extraction of deadlock-free routing algorithms in a 2D network where four channels are presented as

X^+ , X^- , Y^+ , and Y^- . By applying the strategies of Section 5.1, two set arrangements are suggested:

Arrangement 1 : $\{\text{Set1} = [X^+X^-]; \text{Set2} = [Y^+Y^-]\}$

Arrangement 2 : $\{\text{Set1} = [Y^+Y^-]; \text{Set2} = [X^+X^-]\}$

The main procedure in Section 5.2.1 results in the first two entries in the 1st row of Table 1. In this table “ \rightarrow ” represents a transition. Similar entries in the 2nd row can be extracted by running Algorithm 2 and similar entries in the 3rd and 4th rows are achieved by changing the order of transitions between partitions (Section 5.3.3). The last column refers to the exceptional case that is unique to the networks with no VCs (Section 5.2.2). Thereby, in total 12 partitioning options are suggested, guaranteeing the maximum adaptiveness. The resulted turns from these partitioning options (can be extracted similar to Figure 8) are the same as those obtained by applying turn models. Based on the turn model rules, one cycle is removed from each abstract cycle. There are $16(4^2)$ possible combinations of removing one turn from each abstract cycle. For each combination, the possibility of forming simple as well as complex cycles is carefully verified. As reported on [13, 18], out of 16 combinations, 12 are deadlock-free and 3 are unique if symmetry is taken into account, so-called north-last, west-first, and negative-first. The entries related to these algorithms are highlighted in the table.

Table 1: Partitioning options leading to maximum adaptiveness

$X^+X^-Y^+ \rightarrow Y^-$	$Y^+Y^-X^+ \rightarrow X^-$	$X^+Y^+ \rightarrow X^-Y^-$
$X^+X^-Y^- \rightarrow Y^+$	$Y^+Y^-X^- \rightarrow X^+$	$X^+Y^- \rightarrow X^-Y^+$
$Y^- \rightarrow X^+X^-Y^+$	$X^- \rightarrow Y^+Y^-X^+$	$X^-Y^- \rightarrow X^+Y^+$
$Y^+ \rightarrow X^+X^-Y^-$	$X^+ \rightarrow Y^+Y^-X^-$	$X^-Y^+ \rightarrow X^+Y^-$

By increasing the number of partitions to three (Section 5.3.2), a set of new algorithms with a lower degree of adaptiveness can be reached. We listed four options in Table 2 where symmetric ones can be extracted by changing the transition order between partitions. These algorithms have not been of interest in literature as they neither suggest deterministic routing nor provide the maximum adaptiveness.

Table 2: Partitioning options leading to some degrees of adaptiveness

$X^+Y^+ \rightarrow X^- \rightarrow Y^-$	$X^+Y^- \rightarrow X^- \rightarrow Y^+$
$X^-Y^+ \rightarrow X^+ \rightarrow Y^-$	$X^-Y^- \rightarrow X^+ \rightarrow Y^+$

As shown in Table 3, by further increasing the number of partitions to four, deterministic routing algorithms are derived. By transition between partitions, different deterministic routing algorithms are obtained, for example XY (or YX) routing can be reached by switching from the partitions covering the X channels to those partitions covering the Y channels (or vice versa).

Table 3: Partitioning options leading to deterministic routing

$X^+ \rightarrow Y^+ \rightarrow X^- \rightarrow Y^-$	$X^+ \rightarrow Y^- \rightarrow X^- \rightarrow Y^+$
$X^- \rightarrow Y^+ \rightarrow X^+ \rightarrow Y^-$	$X^- \rightarrow Y^- \rightarrow X^+ \rightarrow Y^+$
$X^+ \rightarrow X^- \rightarrow Y^+ \rightarrow Y^-$	$Y^+ \rightarrow Y^- \rightarrow X^+ \rightarrow X^-$

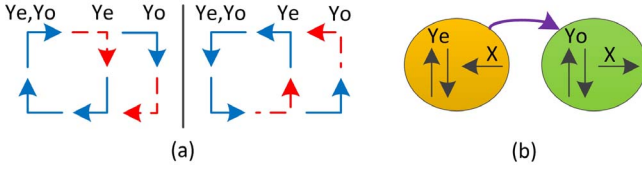


Figure 10: (a) Odd-Even turns; (b) the partitioning option.

6.2 Applying partitioning strategy to the Odd-Even routing algorithm and Hamiltonian path strategy

We show how the EbDa theory can be applied to design the Odd-Even turn model [3]. The rules of the Odd-Even turn model are as follows, shown in Figure 10(a):

- **Rule 1:** Packets are not allowed to take EN_e and ES_e turns at any nodes located in an even column (e).
- **Rule 2:** Packets are not allowed to take NO_w and SO_w turns at any nodes located in an odd column (o).

In all the previous examples, we performed the partitioning based on the direction of channels. However, as it was already mentioned in Definition 6, partitioning can be done in different ways such as *channels located in different columns/rows*. The Odd-Even turn model can be designed by dividing channels into two partitions as: $PA = \{X^- Y_e^*\}$ and $PB = \{X^+ Y_o^*\}$ where Y_e and Y_o refer to the Y channels in the even and odd columns, respectively. Both partitions satisfy Theorem 1 as each partition covers at most one complete pair. Two partitions are also completely disjoint from each other because: 1) partitions are disjoint in the X dimension due to covering two opposite directions; 2) partitions are disjoint in the Y dimension as PA covers the even columns and PB covers the odd columns. Since the partitions are disjoint, packets can move from PA to PB , resulting in all the turns that are offered by the Odd-Even turn model, listed in Table 4. In this table, the formed turns in PA and PB are shown in the 2nd and 3rd rows while the turns formed by the transition from PA to PB is given in the 4th row. In addition to 90-degree turns, some U- and I-turns can be safely taken, reported on the last column of the table. The highlighted entry shows some allowable movement, but since there is no direct connection between even and odd columns, these movements can never be used in the mesh topology. The Odd-Even routing algorithm allows 12 different turns divided into odd and even columns, offering the same level of adaptiveness as those of the west-first routing algorithm with 6 turns in total.

Table 4: Allowable turns in Odd-Even [3]

Extracting turns	90-degree turns	U- & I-turns
in PA	WN_e, WS_e, N_eW, S_eW	$(N_eS_e \text{ or } S_eN_e)$
in PB	EN_o, ES_o, N_oE, S_oE	$(N_oS_o \text{ or } S_oN_o)$
by transition from PA to PB	WN_o, WS_o, N_eE, S_eE	$N_eN_o, N_eS_o, S_eN_o, S_eS_o$

Similarly, it can be easily shown that turns in the Hamiltonian-path strategy [26, 46] can be extracted by the EbDa theory. In the Hamiltonian-path strategy, channels are traced row by row and in total eight 90-degree turns are utilized. By dividing channels into

two disjoint partitions as: $PA = \{X_e^+ X_o^- Y^+\}$ and $PB = \{X_e^- X_o^+ Y^-\}$ twelve 90-degree turns are allowed including all the eight ones suggested by the Hamiltonian-path strategy.

6.3 Applying partitioning strategy to a vertically partially connected 3D network

To show an example of an irregular network, we investigate the Elevator-First routing algorithm which is proposed for vertically partially connected 3D network [14]. Elevator-First is a deterministic routing algorithm using 2, 2, and 1 virtual channels along the X , Y , and Z dimensions, respectively. To deliver a packet to the destination, first the packet has to be delivered to a vertical connection using the XY routing algorithm. After delivery to the destination layer, the packet utilizes the XY routing algorithm again to reach the destination router. Thereby, the Elevator-First routing is built based on sixteen turns as $E1N1, E1S1, W1N1, W1S1, N1U, N1D, S1U, S1D, UE2, UW2, DE2, DW2, E2N2, E2S2, W2N2, W2S2$.

Using the partitioning approach, the same objective can be satisfied with a lower number of VCs while offering a higher degree of adaptiveness (i.e. the design is intensively investigated in [39]). By dividing channels into two partitions as: $P = \{PA[X1^+ Y1^* Z1^+]; PB[X1^- Y2^* Z1^-]\}$, thirty 90-degree turns are allowed (Table 5) in addition to six U- and I-turns. For example fully adaptive routing can be utilized in four regions as NEU, SEU, NWD, SWD and partially adaptive routing can be used in the other four regions as NED, SED, NWU , and SWU . Moreover, many identical turns with different VC numbers can be extracted that can be used to improve the performance.

Table 5: Allowable turns in [39]

Extracting turns	90-degree turns
in PA	$EN1, ES1, EU, N1E, N1U, S1E, S1U, UE, UN1, US1$
in PB	$WN2, WS2, WD, N2W, N2D, S2W, S2D, DW, DN2, DS2$
by transition from PA to PB	$EN2, ES2, ED, N1W, N1D, S1W, S1D, UW, UN2, US2$

7 CONCLUSION

Dally's theory has been used during the last three decades, saying that a design is deadlock-free if there is no cycle in the channel dependency graph. This theory was initially proposed for deterministic routing while it has been extended to adaptive routing by the concept of turn models. In this paper, we introduced three theorems, called EbDa, that allow designing an acyclic channel dependency graph. The turns can be directly extracted from this graph without dealing with turn models any longer. We proposed methodologies to divide channels into partitions in order to reach a design with the maximum adaptiveness for any number of channels and network dimension. Instead of proposing a new routing algorithm, we rather tried to show a roadmap to design deadlock-free routing algorithms with different degrees of adaptiveness in a wormhole switching network. We believe that the usage of the proposed theory is not limited to interconnection networks but to the other areas where designing an acyclic channel dependency graph is an objective.

REFERENCES

- [1] Giuseppe Ascia, Vincenzo Catania, Maurizio Palesi, and Davide Patti. 2008. Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip. *IEEE Trans. Comput.* 57, 6 (June 2008), 809–820. <https://doi.org/10.1109/TC.2008.38>
- [2] Andrew A. Chien and Jae H. Kim. 1995. Planar-adaptive Routing: Low-cost Adaptive Networks for Multiprocessors. *J. ACM* 42, 1 (Jan. 1995), 91–123. <https://doi.org/10.1145/200836.200856>
- [3] Ge-Ming Chiu. 2000. The Odd-Even Turn Model for Adaptive Routing. *IEEE Trans. Parallel Distrib. Syst.* 11, 7 (July 2000), 729–738. <https://doi.org/10.1109/71.877831>
- [4] William Dally and Brian Towles. 2003. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [5] William J. Dally. 1990. Performance analysis of k-ary n-cube interconnection networks. *Computers, IEEE Transactions on* 39, 6 (Jun 1990), 775–785. <https://doi.org/10.1109/12.53599>
- [6] William J. Dally. 1992. Virtual-channel flow control. *Parallel and Distributed Systems, IEEE Transactions on* 3, 2 (Mar 1992), 194–205. <https://doi.org/10.1109/71.127260>
- [7] William J. Dally and Charles L. Seitz. 1987. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *Computers, IEEE Transactions on* C-36, 5 (May 1987), 547–553. <https://doi.org/10.1109/TC.1987.1676939>
- [8] Reetuparna Das, Soumya Eachempati, Asit K. Mishra, Vijaykrishnan Narayanan, and Chita R. Das. 2009. Design and evaluation of a hierarchical on-chip interconnect for next-generation CMPs. In *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*. 175–186. <https://doi.org/10.1109/HPCA.2009.4798252>
- [9] José Duato. 1993. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Trans. Parallel Distrib. Syst.* 4, 12 (Dec. 1993), 1320–1331. <https://doi.org/10.1109/71.250114>
- [10] José Duato. 1995. A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Trans. Parallel Distrib. Syst.* 6, 10 (Oct. 1995), 1055–1067. <https://doi.org/10.1109/71.473515>
- [11] José Duato. 1996. A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks. *Parallel and Distributed Systems, IEEE Transactions on* 7, 8 (Aug 1996), 841–854. <https://doi.org/10.1109/71.532115>
- [12] José Duato and Timothy M. Pinkston. 2001. A General Theory for Deadlock-Free Adaptive Routing Using a Mixed Set of Resources. *IEEE Trans. Parallel Distrib. Syst.* 12, 12 (Dec. 2001), 1219–1235. <https://doi.org/10.1109/71.970556>
- [13] José Duato, Sudhakar Yalamanchili, and Lionel Ni. 2002. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [14] Florentine Dubois, Abbas Sheibanyrad, Frederic Petrot, and Maryam Bahmani. 2013. Elevator-First: A Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs. *IEEE Trans. Comput.* 62, 3 (March 2013), 609–615. <https://doi.org/10.1109/TC.2011.239>
- [15] Binzhang Fu, Yinhe Han, Jun Ma, HuaWei Li, and Xiaowei Li. 2011. An abacus turn model for time/space-efficient reconfigurable routing. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*. 259–270.
- [16] David Gelernter. 1981. A DAG-Based Algorithm for Prevention of Store-and-Forward Deadlock in Packet Networks. *Computers, IEEE Transactions on* C-30, 10 (Oct 1981), 709–715. <https://doi.org/10.1109/TC.1981.1675690>
- [17] Christopher J. Glass and Lionel M. Ni. 1992. Maximally Fully Adaptive Routing in 2D Meshes. In *International Conference on Parallel Processing, volume 1*. 101–104.
- [18] Christopher J. Glass and Lionel M. Ni. 1992. The Turn Model for Adaptive Routing. In *Proceedings of the 19th Annual International Symposium on Computer Architecture (ISCA '92)*. ACM, New York, NY, USA, 278–287. <https://doi.org/10.1145/139669.140384>
- [19] Paul. Gratz, Boris Grot, and Stephen W. Keckler. 2008. Regional congestion awareness for load balance in networks-on-chip. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*. 203–214. <https://doi.org/10.1109/HPCA.2008.4658640>
- [20] Luis Gravano, Gustavo D. Pifarré, Pablo E. Berman, and Jorge L.C. Sanz. 1994. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks. *Parallel and Distributed Systems, IEEE Transactions on* 5, 12 (Dec 1994), 1233–1251. <https://doi.org/10.1109/71.334898>
- [21] Jingcao Hu and Radu Marculescu. 2004. DyAD - smart routing for networks-on-chip. In *Design Automation Conference, 2004. Proceedings. 41st*. 260–263.
- [22] Letian Huang, Junshi Wang, Masoumeh Ebrahimi, Masoud Daneshmand, Xiaofan Zhang, Guangjun Li, and Axel Jantsch. 2016. Non-Blocking Testing for Network-on-Chip. *IEEE Trans. Comput.* 65, 3 (March 2016), 679–692. <https://doi.org/10.1109/TC.2015.2489216>
- [23] Parviz Kermani and Leonard Kleinrock. 1979. Virtual cut-through: A new computer communication switching technique. *Computer Networks (1976)* 3, 4 (1979), 267 – 286. [https://doi.org/10.1016/0376-5075\(79\)90032-1](https://doi.org/10.1016/0376-5075(79)90032-1)
- [24] Ming Li, Qing-An Zeng, and Wen-Ben Jone. 2006. DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In *Design Automation Conference, 2006 43rd ACM/IEEE*. 849–852. <https://doi.org/10.1109/DAC.2006.229242>
- [25] Xiaola Lin, Philip K. McKinley, and Lionel M. Ni. 1993. The Message Flow Model for Routing in Wormhole-Routed Networks. In *Parallel Processing, 1993. ICPP 1993. International Conference on*, Vol. 1. 294–297. <https://doi.org/10.1109/ICPP.1993.176>
- [26] Xiaola Lin and Lionel M. Ni. 1991. Deadlock-free Multicast Wormhole Routing in Multicomputer Networks. *SIGARCH Comput. Archit. News* 19, 3 (April 1991), 116–125. <https://doi.org/10.1145/115953.115965>
- [27] Daniel H. Linder and Jim C. Harden. 1991. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. *Computers, IEEE Transactions on* 40, 1 (Jan 1991), 2–12. <https://doi.org/10.1109/12.67315>
- [28] Pedro Lopez, Juan M. Martinez, and José. Duato. 1998. A very efficient distributed deadlock detection mechanism for wormhole networks. In *High-Performance Computer Architecture, 1998. Proceedings., 1998 Fourth International Symposium on*. 57–66. <https://doi.org/10.1109/HPCA.1998.650546>
- [29] Sheng Ma, Natalie Enright Jerger, and Zhiying Wang. 2011. DBAR: An Efficient Routing Algorithm to Support Multiple Concurrent Applications in Networks-on-chip. In *Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA '11)*. ACM, New York, NY, USA, 413–424. <https://doi.org/10.1145/2000064.2000113>
- [30] Sheng Ma, Natalie Enright Jerger, and Zhiying Wang. 2012. Whole Packet Forwarding: Efficient Design of Fully Adaptive Routing Algorithms for Networks-on-chip. In *Proceedings of the 2012 IEEE 18th International Symposium on High-Performance Computer Architecture (HPCA '12)*. IEEE Computer Society, Washington, DC, USA, 1–12. <https://doi.org/10.1109/HPCA.2012.6169049>
- [31] Prasant Mohapatra. 1998. Wormhole Routing Techniques for Directly Connected Multicomputer Systems. *ACM Comput. Surv.* 30, 3 (Sept. 1998), 374–410. <https://doi.org/10.1145/292469.292472>
- [32] Shubendu S. Mukherjee, Peter Bannan, Steven Lang, Aaron Spink, and David Webb. 2001. The Alpha 21364 Network Architecture. In *Proceedings of the The Ninth Symposium on High Performance Interconnects (HOTI '01)*. IEEE Computer Society, Washington, DC, USA, 113–. <http://dl.acm.org/citation.cfm?id=572719.876330>
- [33] Robert Mullins, Andrew West, and Simon Moore. 2004. Low-Latency Virtual-Channel Routers for On-Chip Networks. In *Proceedings of the 31st Annual International Symposium on Computer Architecture (ISCA '04)*. IEEE Computer Society, Washington, DC, USA, 188–. <http://dl.acm.org/citation.cfm?id=998680.1006717>
- [34] Lionel M. Ni and Philip K. McKinley. 1993. A Survey of Wormhole Routing Techniques in Direct Networks. *Computer* 26, 2 (Feb. 1993), 62–76. <https://doi.org/10.1109/2.191995>
- [35] Chrysostomos A. Nicopoulos, Dongkook Park, Jongman Kim, N. Vijaykrishnan, Mazin S. Yousif, and Chita R. Das. 2006. ViChAr: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 39)*. IEEE Computer Society, Washington, DC, USA, 333–346. <https://doi.org/10.1109/MICRO.2006.50>
- [36] Maurizio Palesi and Masoud Daneshmand. 2016. *Routing Algorithms in Networks-on-Chip*. Springer Publishing Company, Incorporated.
- [37] Partha P. Pande, Cristian Grecu, Michael Jones, Andre Ivanov, and Resve Saleh. 2005. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. *IEEE Trans. Comput.* 54, 8 (Aug. 2005), 1025–1040. <https://doi.org/10.1109/TC.2005.134>
- [38] Li-Shiuan Peh and William J. Dally. 2001. A Delay Model and Speculative Architecture for Pipelined Routers. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA '01)*. IEEE Computer Society, Washington, DC, USA, 255–. <http://dl.acm.org/citation.cfm?id=580550.876446>
- [39] Ronak Salamat, Misagh Khayambashi, Masoumeh Ebrahimi, and Nader Bagherzadeh. 2016. A Resilient Routing Algorithm with Formal Reliability Analysis for Partially Connected 3D-NoCs. *IEEE Trans. Comput.* 65, 11 (Nov 2016), 3265–3279. <https://doi.org/10.1109/TC.2016.2532871>
- [40] Michael D. Schroeder, Andrew D. Birrell, Michael Burrows, Hal Murray, Roger M. Needham, Thomas L. Rodeheffer, Edwin H. Satterthwaite, and Charles P. Thacker. 1991. Autonet: a high-speed, self-configuring local area network using point-to-point links. *Selected Areas in Communications, IEEE Journal on* 9, 8 (Oct 1991), 1318–1335. <https://doi.org/10.1109/49.105178>
- [41] Loren Schwiebert and D. N. Jayasimha. 1996. A Necessary and Sufficient Condition for Deadlock-Free Wormhole Routing. *J. Parallel and Distrib. Comput.* 32, 1 (1996), 103 – 117. <https://doi.org/10.1006/jpdc.1996.0008>
- [42] Arjun Singh, William J. Dally, Amit K. Gupta, and Brian Towles. 2003. GOAL: A Load-balanced Adaptive Routing Algorithm for Torus Networks. *SIGARCH Comput. Archit. News* 31, 2 (May 2003), 194–205. <https://doi.org/10.1145/871656.859641>

- [43] Aniruddha S. Vaidya, Anand Sivasubramaniam, and Chita R. Das. 2001. Impact of Virtual Channels and Adaptive Routing on Application Performance. In *Proceedings of the 2001 ACM SIGCPR Conference on Computer Personnel Research (SIGCPR '01)*. ACM, New York, NY, USA, 223–237. <https://doi.org/10.1145/371209.371542>
- [44] Freek Verbeek and Julien Schmaltz. 2011. On Necessary and Sufficient Conditions for Deadlock-Free Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems* 22, 12 (2011), 2022–2032. <https://doi.org/10.1109/TPDS.2011.60>
- [45] Hang-Sheng Wang, Xinping Zhu, Li-Shiuan Peh, and Sharad Malik. 2002. Orion: a power-performance simulator for interconnection networks. In *Microarchitecture, 2002. (MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*. 294–305. <https://doi.org/10.1109/MICRO.2002.1176258>
- [46] Nen-Chung Wang, Chih-Ping Chu, and Tzung-Shi Chen. 2002. A dual-hamiltonian-path-based multicasting strategy for wormhole-routed star graph interconnection networks. *J. Parallel and Distrib. Comput.* 62, 12 (2002), 1747 – 1762. [https://doi.org/10.1016/S0743-7315\(02\)00007-2](https://doi.org/10.1016/S0743-7315(02)00007-2)
- [47] Dong Wu, Bashir M. Al-Hashimi, and Marcus T. Schmitz. 2006. Improving Routing Efficiency for Network-on-chip Through Contention-aware Input Selection. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference (ASP-DAC '06)*. IEEE Press, Piscataway, NJ, USA, 36–41. <https://doi.org/10.1145/1118299.1118310>
- [48] Yi Xu, Bo Zhao, Youtao Zhang, and Jun Yang. 2010. Simple virtual channel allocation for high throughput and high frequency on-chip routers. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*. 1–11. <https://doi.org/10.1109/HPCA.2010.5416640>
- [49] Haofan Yang, Jyoti Tripathi, Natalie Enright Jerger, and Dan Gibson. 2014. Dodec: Random-Link, Low-Radix On-Chip Networks. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. 496–508. <https://doi.org/10.1109/MICRO.2014.19>