



Ville Rantala | Teijo Lehtonen | Juha Plosila

# Network on Chip Routing Algorithms

TURKU CENTRE *for* COMPUTER SCIENCE

TUCS Technical Report  
No 779, August 2006





# Network on Chip Routing Algorithms

Ville Rantala

Teijo Lehtonen

Juha Plosila

University of Turku, Department of Information Technology

Joukahaisenkatu 3-5 B, 20520 Turku, Finland

{vttran,tetale,juplos}@utu.fi

TUCS Technical Report

No 779, August 2006

## **Abstract**

Network on Chip (NoC) is a new paradigm to make the interconnections inside a System on Chip (SoC) system. In traditional solutions interconnections are realized using a bus structure. While integration increases the bus structure does not meet the needs of the new technology. Bus starts to be narrow and in the worst case it begins to block traffic. In NoC technology the bus structure is replaced with a network which is a lot similar to the Internet. Segments communicate with each other by sending packetized data over this network.

Just like a computer network, a NoC network consists of devices that use the network, routers that direct the traffic between devices and wires that connect devices to routers and routers to other routers. In the network design of the NoC the most essential things are a network topology and a routing algorithm. Routers route the packets based on the algorithm that they use. There are many kind of different algorithms for different systems to choose. Every system has its own requirements for the routing algorithm.

This report looks through the basics of networking on Network on Chip systems and presents proposed routing algorithms to be used on NoCs. In the end of the report the proposed router architectures are also presented.

**Keywords:** Network on Chip, routing algorithm, router architecture

**TUCS Laboratory**  
Distributed Systems

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Routing on NoC</b>	<b>5</b>
2.1	Network Topologies . . . . .	5
2.2	Problems on Routing . . . . .	7
2.2.1	Deadlock. . . . .	8
2.2.2	Livelock. . . . .	8
2.2.3	Starvation. . . . .	9
2.3	Network Flow Control . . . . .	9
<b>3</b>	<b>Oblivious Routing Algorithms</b>	<b>10</b>
3.1	Dimension Order Routing . . . . .	10
3.1.1	XY routing . . . . .	10
3.2	Turn Models . . . . .	11
3.3	Deterministic Routing Algorithms . . . . .	12
3.3.1	Shortest Path Routing . . . . .	13
3.3.2	Source Routing . . . . .	13
3.3.3	Destination-tag Routing . . . . .	14
3.3.4	Topology Adaptive Routing . . . . .	14
3.4	Stochastic Routing Algorithms . . . . .	14
3.4.1	Flooding Algorithms . . . . .	14
3.5	Summary . . . . .	16
<b>4</b>	<b>Adaptive Routing Algorithms</b>	<b>17</b>
4.1	Minimal Adaptive Routing . . . . .	17
4.2	Fully Adaptive Routing . . . . .	17
4.2.1	Congestion Look Ahead . . . . .	17
4.3	Turnaround Routing . . . . .	17
4.4	Other Adaptive Routing Algorithms . . . . .	18
4.5	Summary . . . . .	20
<b>5</b>	<b>Router Architectures</b>	<b>21</b>
5.1	Oblivious Routers . . . . .	21
5.1.1	Virtual Channel Router . . . . .	21
5.1.2	Xpipes . . . . .	22
5.1.3	Æthereal . . . . .	23
5.1.4	Proteo . . . . .	24
5.1.5	MANGO . . . . .	25
5.1.6	SoCBUS . . . . .	25
5.1.7	Arteris . . . . .	26
5.1.8	STNoC . . . . .	26
5.2	Adaptive Routers . . . . .	26

5.2.1	DyAD . . . . .	26
5.2.2	SPIN . . . . .	27
5.2.3	XGFT . . . . .	28
5.2.4	Nostrum . . . . .	29
5.3	Summary . . . . .	29
<b>6</b>	<b>Conclusions</b>	<b>30</b>

# 1 Introduction

*Network on Chip (NoC)* is a new paradigm for *System on Chip (SoC)* design. Increasing integration produces a situation where bus structure, which is commonly used in SoC, becomes blocked and increased capacitance poses physical problems. In NoC architecture traditional bus structure is replaced with a network which is a lot similar to the Internet. Data communications between segments of chip are packetized and transferred through the network. The network consists of wires and routers. Processors, memories and other IP-blocks (Intellectual Property) are connected to routers. A routing algorithm plays a significant role on network's operation. Routers make the routing decisions based on the routing algorithm.

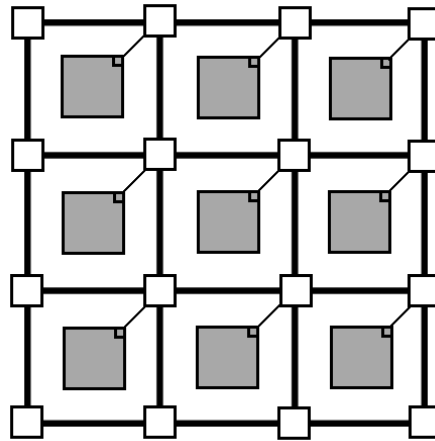


Figure 1: Network on Chip.

Different devices with different purposes have different requirements for routing algorithms. Thus there have been designed several routing algorithms with various features and purposes.

There are a couple of requirements that every Network on Chip implementation has to meet. Performance requirements are small latency, guaranteed throughput, path diversity, sufficient transfer capacity and low power consumption. Architectural requirements are scalability, generality and programmability. Fault and distraction tolerancy as well as valid operation are major on Quality of Service.

The network traffic in NoC network is divided to two types, Guaranteed Throughput (GT) and Best Effort (BE) traffics. Guaranteed Throughput is also sometimes called as Guaranteed Service (GS). An arbiter of GT traffic guarantees that some portion – for example 99% – of sent data overtakes the receiver in some time slot. GT supplier assumes that the sender complies with networks operation requirements. Guaranteed throughput works best with routing algorithm that acts like circuit switched network.

Best-effort packets are arbitrated as trustworthy as possible. Still there are no guarantees that BE packets will ever reach the receiver. Latencies can vary and in the worst case packets can be lost. Traffic in a basic packet switched network is mostly BE-traffic. [12]

The aim of this report is to review the proposed routing algorithms to be used on the Network on Chip systems. The basics of networking on NoCs and architectures of proposed routers are also presented. The report is organized as follows: The most common network topologies, routing problems and network flow control mechanisms are presented in Section 2. Oblivious and adaptive routing algorithms are discussed in Sections 3 and 4. Section 5 deals with router architectures and finally conclusions are presented in Section 6.



## 2 Routing on NoC

Routing on NoC is quite similar to routing on any network. A routing algorithm determines how the data is routed from sender to receiver.

Routing algorithms are divided into two groups, oblivious and adaptive algorithms. Oblivious algorithms are also divided into two subgroups: deterministic and stochastic algorithms. Oblivious algorithms route packets without any information about traffic amounts and conditions of the network, deterministic algorithms route packets always along a same route and stochastic routing is based on randomness.

### 2.1 Network Topologies

A network can be regular or irregular and it is non-blocking if it can manage all the requests that are offered to it. In a packet switched case this kind of network is also called as non-interfering network. Non-interfering network can deliver all the packets in guaranteed time. [12] The basic regular network topologies are listed below.

**Mesh.** A mesh-shaped network consists of  $m$  columns and  $n$  rows. The routers are situated in the intersections of two wires and the computational resources are near routers. Addresses of routers and resources can be easily defined as  $x$ - $y$ -coordinates in mesh. Regular mesh network is also called as *Manhattan Street* network.

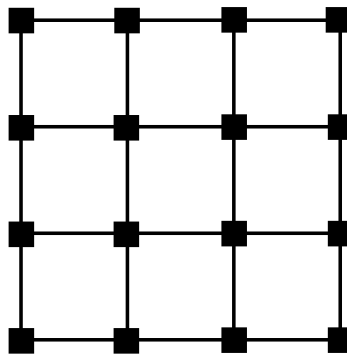


Figure 2: Mesh network.

**Torus.** A Torus network is an improved version of basic mesh network. A simple torus network is a mesh in which the heads of the columns are connected to the tails of the columns and the left sides of the rows are connected to the right sides of the rows. Torus network has better path diversity than mesh network, and it also has more minimal routes.

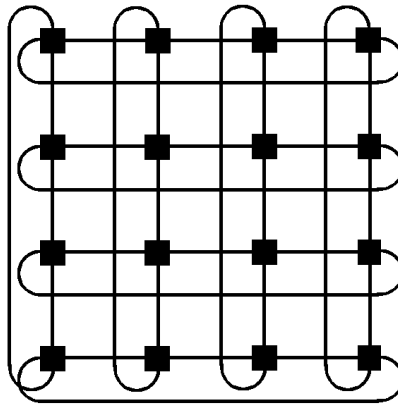


Figure 3: Torus network.

**Tree.** In a tree topology nodes are routers and leaves are computational resources. The routers above a leaf are called as leaf's ancestors and correspondingly the leafs below the ancestor are its children. In a fat tree topology each node has replicated ancestors which means that there are many alternative routes between nodes.

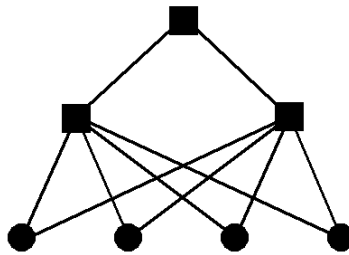


Figure 4: Fat-tree network.

**Butterfly.** A butterfly network is uni- or bidirectional and butterfly-shaped network typically uses a deterministic routing. For example a simple unidirectional butterfly network contains 8 input ports, 8 output ports and 3 router levels which each contains 4 routers. Packets arriving to the inputs on the left side of the network are routed to the correct output on the right side of the network. [12] In a bidirectional butterfly network, all the inputs and outputs are on the same side of the network. Packets coming to inputs are first routed to the other side of the network, then turned around and routed back to the correct output.

**Polygon.** The simplest polygon network is a circular network where packets travel in loop from router to other. Network becomes more diverse when chords

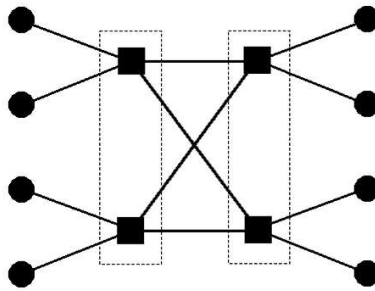


Figure 5: Butterfly network with 4 inputs, 4 outputs and 2 router stages each containing 2 routers.

are added to the circle. When there are chords only between opposite routers, the topology is called as spidergon.

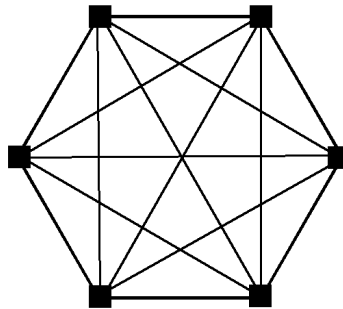


Figure 6: Polygon (hexagon) network with all potential chords.

**Star.** A star network consists of a central router in the middle of the star, and computational resources or subnetworks in the spikes of the star. The capacity requirements of the central router are quite large, because all the traffic between the spikes goes through the central router. That causes a remarkable possibility of congestion in the middle of the star.

## 2.2 Problems on Routing

Problems on oblivious routing typically arise when the network starts to block traffic. The only solution to these problems is to wait for traffic amount to reduce and try again. Deadlock, livelock and starvation are potential problems on both oblivious and adaptive routing.

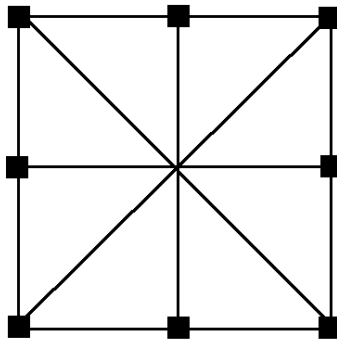


Figure 7: Spidergon network, where opposite routers are connected together.

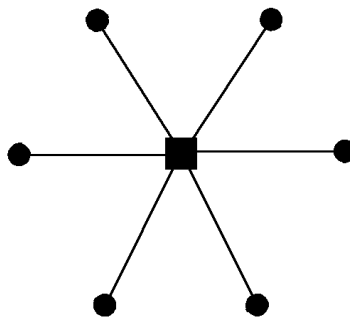


Figure 8: Star network.

### 2.2.1 Deadlock.

Routing is in deadlock when two packets are waiting each other to be routed forward. Both of the packets reserve some resources and both are waiting each other to release the resources. Routers do not release the resources before they get the new resources and so the routing is locked.

### 2.2.2 Livelock.

Livelock occurs when a packet keeps spinning around its destination without ever reaching it. This problem exists in non-minimal routing algorithms. Livelock should be cut out to guarantee packet's throughput.

There are a couple of resorts to avoid the livelock. Time to live (TTL) counter counts how long a packet has travelled in the network. When the counter reaches some predetermined value, the packet will be removed from the network. The another resort is to give packets a priority which is based on packet's age. The oldest packet always finally get the highest priority and will be routed forward. [12]

### 2.2.3 Starvation.

Using different priorities can cause a situation where some packets with lower priorities never reach their destinations. This occurs when the packets with higher priorities reserve the resources all the time. Starvation can be avoided by using a fair routing algorithm or reserving some bandwidth only for low-priority packets. [14]

## 2.3 Network Flow Control

Network flow control, also called as routing mode, determines how packets are transmitted inside a network. The mode is not directly dependent to routing algorithm. Many algorithms are designed to use some given mode, but most of them do not define which mode should be used.

**Store-and-Forward Routing.** Store-and-forward is the simplest routing mode. Packets move in one piece, and entire packet has to be stored in the router's memory before it can be forwarded to the next router. So the buffer memory has to be as large as the largest packet in the network. The latency is the combined time of receiving a packet and sending it ahead. Sending cannot be started before the whole packet is received and stored in the router's memory.

**Virtual Cut-Through Routing.** Virtual cut-through is a improved version of store-and-forward mode. A router can begin to send packet to the next router as soon as the next router gives a permission. Packet is stored in the router until the forwarding begins. Forwarding can be started before the whole packet is received and stored to router. The mode needs as much buffer memory as store-and-forward mode, but latencies are lower.

**Wormhole Routing.** In wormhole routing packets are divided to small and equal sized flits (*flow control digit* or *flow control unit*). A first flit of a packet is routed similarly as packets in the virtual cut-through routing. After first flit the route is reserved to route the remaining flits of the packet. This route is called wormhole. Wormhole mode requires less memory than the two other modes because only one flit has to be stored at once. Also the latency is smaller and a risk of deadlock is larger. The risk can be reduced by multiplexing several virtual ports to one physical port, so the possibility of traffic congestion and blocking decreases. [31]

### 3 Oblivious Routing Algorithms

Oblivious routing algorithms have no information about conditions of the network, like traffic amounts or congestions. A router makes routing decisions on the grounds of some algorithm or for example randomly. The simplest oblivious routing algorithm is a *minimal turn* routing. It routes packets using as few turns as possible.

#### 3.1 Dimension Order Routing

Dimension order routing (DOR) is a typical minimal turn algorithm. The algorithm determines to what direction packets are routed during every stage of the routing. [12]

##### 3.1.1 XY routing

XY routing is a dimension order routing which routes packets first in x- or horizontal direction to the correct column and then in y- or vertical direction to the receiver. XY routing suits well on a network using mesh or torus topology. Addresses of the routers are their xy-coordinates. XY routing never runs into deadlock or livelock. [15]

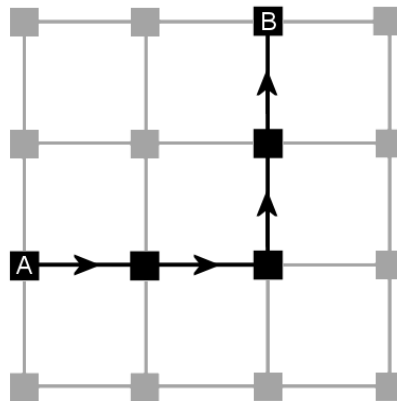


Figure 9: XY routing from router A to router B.

There are some problems in the traditional XY routing. The traffic does not extend regularly over the whole network because the algorithm causes the biggest load in the middle of the network. There is a need for algorithms which equalize the traffic load over the whole network.

**Pseudo Adaptive XY Routing.** Pseudo adaptive XY routing works in deterministic or adaptive mode depending on the state of the network. Algorithm works in deterministic mode when the network is not or only slightly congested. When

network becomes blocked, the algorithm switches to the adaptive mode and starts to search routes that are not congested.

Pseudo adaptive XY routing works on mesh network which consists of routers, wires and IP-blocks. Every router has five bidirectional ports: north, south, east, west and local. Local port connects router to its local core while the other ports are connected to neighboring routers. Each port has a small temporary storage buffer and a 2-bit status identifier called quantized load value. Identifier tells to other routers if the router is congested and cannot accept new packets.

A router assigns priorities to incoming packets when there are more than one coming simultaneously. Packets from north have the highest priority, then south, east and at last packets incoming from west have the lowest priority.

While a traditional XY routing causes network loads more in the middle of the network than to lateral areas, the pseudo adaptive algorithm divides the traffic more equally over the whole network. [15]

**Surrounding XY Routing.** Surrounding XY routing (*S-XY*) has three different routing modes. *N-XY* (*Normal XY*) mode works just like the basic XY routing. It routes packets first along x-axis and then along y-axis. Routing stays on *N-XY* mode as long as network is not blocked and routing does not meet inactive routers. *SH-XY* (*Surround horizontal XY*) mode is used when the router's left or right neighbor is deactivated. Correspondingly the third mode *SV-XY* (*Surround vertical XY*) is used when the upper or lower neighbor of the router is inactive.

The *SH-XY* mode routes packets to the correct column on the grounds of coordinates of the destination. The algorithm bypasses packets around the inactive routers along the shortest possible path. The situation is a little bit different in the *SV-XY* mode because the packets are already in the right column. Packets can be routed to left or right. Operation in *SH-XY* and *SV-XY* modes is shown in Figure 10. The routers in the *SH-XY* and *SV-XY* modes add a small identifier to the packets that tells to other routers that these packets are routed using *SH-XY* or *SV-XY* mode. Thus the other routers do not send the packets backwards.

Surrounding XY routing is used in a DyNoC. It is a method that supports communication between modules which are dynamically placed on a device. [9]

## 3.2 Turn Models

Turn model algorithms determine a turn or turns which are not allowed while routing packets through a network. Turn models are livelock-free.

**West-first Routing.** A west-first routing algorithm prevents all turns to west. So the packets going to west must be first transmitted as far to west as necessary. Routing packets to west is not possible later.

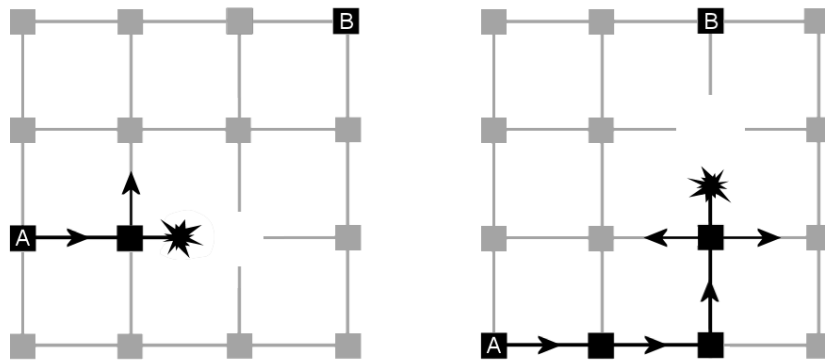


Figure 10: Surrounding XY routing in SH-XY and SV-XY modes. There are 2 optional directions in SV-XY state.

**North-last Routing.** Turns away from north are not possible in a north-last routing algorithm. Thus the packets which need to be routed to north, must be transferred there at last.

**Negative-first Routing.** Negative-first routing algorithm allows all other turns except turns from positive direction to negative direction. Packet routings to negative directions must be done before anything else. [20]

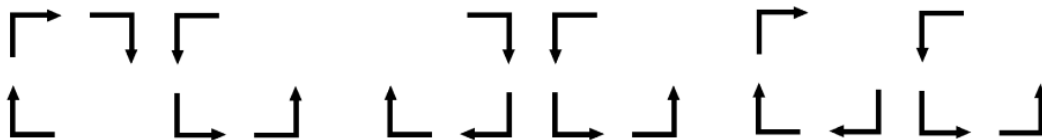


Figure 11: Allowed turns in *west-first*, *north-last* and *negative first* routing algorithms.

### 3.3 Deterministic Routing Algorithms

Deterministic routing algorithms route packets every time from a certain point A to a certain point B along a fixed path. Deterministic algorithms are used in both regular and irregular networks. In congestion free networks deterministic algorithms are reliable and have low latency. They suit well on real time systems because packets always reach the destination in correct order and so a reordering is not necessary. In the simplest case each router has a routing table that includes routes to all other routers in the network. When network structure changes, every router has to be updated.



### 3.3.1 Shortest Path Routing

A shortest path routing is the simplest deterministic routing algorithm. Packets are always routed along the shortest possible path. A distance vector routing and a link state routing are shortest path routing algorithms.

**Distance Vector Routing.** Each router has a routing table that contains information about neighbor routers and all recipients. Routers exchange routing table information with each other and this way keep their own tables up to date. Routers route packets by counting the shortest path on the grounds of their routing tables and then send packets forward. Distance vector routing is a simple method because each router does not have to know the structure of the whole network.

**Link State Routing.** Link state routing is a modification of distance vector routing. The basic idea is the same as in distance vector routing, but in link state routing each router shares its routing table with every other router in the network. Link state routing in Network on Chip systems is a little bit customized version of the traditional one. The routing tables covering the whole network are stored in router's memory already during the production stage. Routers use their routing table updating mechanisms only if there are remarkable changes in the network's structure or if some faults appear. [3]

### 3.3.2 Source Routing

In a source routing a sender makes all decisions about a routing path of a packet. The whole route is stored in a header of packet before sending, and routers along the path do the routing just like the sender has determined it. Two router architectures using source routing are presented later in this report on section 5.1.1.

A *vector routing* works basically like the source routing. In the vector routing the routing path is represented as a chain of unit vectors. Each unit vector corresponds to one hop between two routers. Routing paths do not have to be the shortest possible.

*Arbitration look ahead scheme (ALOAS)* is a faster version of source routing. The information of routing path has been supplied to routers along the path before the packets are even sent. Route information moves along a special channel that is reserved only for this purpose. [13, 23, 35]

A *contention-free routing* is a algorithm based on routing tables and time division multiplexing (TDM). Each router has a routing table that involves correct output ports and time slots to every potential sender–receiver pairs. Contention-free routing algorithm is used in Philips *Æthereal NoC* system and it is also called as a *clockwork routing*. An architecture of the *Æthereal* router using contention-free algorithm is represented on section 5.1.3. [18, 28, 29]

### 3.3.3 Destination-tag Routing

A destination-tag routing is a bit like an inversed version of the source routing. The sender stores the address of the receiver, also known as a destination-tag, to the header of the packet in the beginning of the routing. Every router makes a routing decisions independently on the grounds of the address of the receiver. The destination-tag routing is also know as a *floating vector routing*. [12, 35]

### 3.3.4 Topology Adaptive Routing

Deterministic routing algorithms can be improved by adding some adaptive features to them. A topology adaptive routing algorithm is slightly adaptive. The algorithm works like a basic deterministic algorithm but it has one feature which makes it suitable to dynamic networks. Systems administrator can update the routing tables of the routers if necessary. A corresponding algorithm is also know as an *online oblivious routing*. The cost and latency of the topology adaptive routing algorithm are near to costs and latencies of basic deterministic algorithms. A facility of topology adaptiveness is its suitability to irregular and dynamic networks.

## 3.4 Stochastic Routing Algorithms

Routing with stochastic routing algorithms is based on coincidence and an assumption that every packet sooner or later reaches its destination. Stochastic algorithms are typically simple and fault-tolerant. Throughput of data is especially good but as a drawback, stochastic algorithms are quite slow and they use plenty of network resources.

Stochastic routing algorithms determine packet's *time to live (TTL)*. It is a time how long a packet is allowed to move around in the network. After the determined time has been reached, the packet will be removed from the network.

### 3.4.1 Flooding Algorithms

The most common stochastic algorithm type is the flooding algorithms. Here are three different appliances of flooding.

**Probabilistic Flood.** The simplest stochastic routing algorithm is the probabilistic flooding algorithm. Routers send a copy of an incoming packet to all possible directions without any information about the location of packet's destination. The packet's copies diffuse over the whole network like a flood. Finally at least one of the copies will arrive to its receiver and the redundant copies will be removed.

**Directed Flood.** A directed flood routing algorithm is a improved version of probabilistic flood. It directs packets approximately to the direction where their

destination exists. The directed flood is more fault-tolerant than the probabilistic flood and uses less network resources.

**Random Walk.** A random walk algorithm sends a predetermined amount of packet's copies to the network. Every router along the routing path sends incoming packets forward through some of its output ports. The packets are directed in the same way as in the directed flood algorithm. The random walk is as fault-tolerant as the directed flood but consumes less energy and bandwidth.

Costs of each 3 algorithms are equivalent.

**Valiant's Random Algorithm.** Valiant's random algorithm is a partly stochastic routing algorithm. One main problem in the oblivious routing algorithms is that they affect an irregular load on the network. The load is especially high in the middle areas of the network. Valiant's random algorithm equalizes traffic load on networks that have a good path diversity. First the algorithm randomly picks one intermediate node and routes packets to it. Then the packets are simply routed to their destination. Routing from beginning to the intermediate node and then to the destination are done using some of oblivious algorithms.

Valiant's algorithm effectively equalizes network's load over the whole network regardless of network's topology. [12]

### 3.5 Summary

The outlines and features of the oblivious routing algorithms presented above are listed in Table 1.

Table 1: Oblivious routing algorithms.

ALGORITHM	OUTLINES	FEATURES	REFERENCES
Dimension order	routing in one dimension at a time	simple	[12]
XY	routing first in X and then in Y dimension	simple, loads network deadlock- and livelock-free	[15]
Pseudo adaptive XY	partly adaptive XY routing	livelock-free, congestion avoidance	[15]
Surrounding XY	partly adaptive XY routing	congestion avoidance	[9]
Turn model	some turns forbidden	livelock-free	[20]
Valiant's random Source	partly stochastic deterministic, sender determines the route	balances network's load simple routing	[12] [13, 35]
Destination-tag	deterministic, routers determine the route	simple sending	[12, 18, 28] [29, 35]
ALOAS	deterministic, application of source routing	fast routing	[23]
Topology adaptive	reprogrammable routing tables	suitable to dynamic networks	[6, 7]
Probabilistic flood	stochastic	cheap, consumes a lot of resources	[30]
Directed flood	stochastic	fault-tolerant, consumes a lot of resources	[30]
Random walk	stochastic	fault-tolerant	[30]

## 4 Adaptive Routing Algorithms

### 4.1 Minimal Adaptive Routing

Minimal adaptive routing algorithm always routes packets along the shortest path. The algorithm is effective when more than one minimal, or as short as possible, routes between sender and receiver exist. The algorithm uses route which is least congested. [12]

### 4.2 Fully Adaptive Routing

Fully adaptive routing algorithm uses always a route which is not congested. The algorithm does not care although the route is not the shortest path between sender and receiver. Typically an adaptive routing algorithm sets alternative congestion free routes to order of superiority. The shortest route is the best one. [12]

#### 4.2.1 Congestion Look Ahead

A congestion look ahead algorithm gets information about blocks from other routers. On the grounds of this information the routing algorithm can direct packets to bypass the congestions. [24]

### 4.3 Turnaround Routing

Turnaround routing is a routing algorithm for butterfly and fat-tree networks. Senders and receivers of packets are all on the same side of the network. Packets are first routed from sender to some random intermediate node on the other side of the network. In this node the packets are turned around and then routed to the destination on the same side of the network, where the whole routing started. The routing from the intermediate node to the definite receiver is done with the destination-tag routing (see 3.3.3 on page 14).

Routers in turnaround routing are bidirectional which means that packets can flow through router in both forward and backward directions. The algorithm is deadlock-free because packets only turn around once from a forward channel to a backward channel.

*SPIN (Scalable Programmable Interconnect Network)* is a fat-tree shaped network which uses turnaround routing algorithm. In fault-tolerant *XGFT* system (*eXtended Generalized Fat Tree*) the turnaround routing is called as *turnback routing*. The network topology in *XGFT* systems is also fat-tree. *XGFT*'s turnback routing slightly differs from the basic turnaround algorithm. While traditional turnaround routing chooses the intermediate node randomly, the *XGFT*'s turnback algorithm can choose it by itself. This is useful when the network is congested. [1, 21, 26]

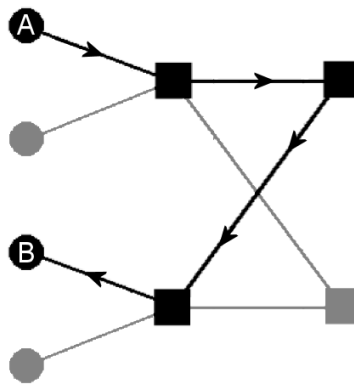


Figure 12: Turnaround routing from point A to point B in a butterfly network.

**Turn-Back-When-Possible.** Turn-back-when-possible (*TBWP*) is an algorithm for routing on tree networks. It is a little bit improved version of the turnaround routing. When turn-back channels are busy, the algorithm looks for free routing path on a higher switch level. A turn-back channel is a channel between a forward and a backward channel. It is used to change the routing direction in the network. [20]

#### 4.4 Other Adaptive Routing Algorithms

**IVAL.** IVAL (*Improved VALiant's randomized routing*) is an improved version of the oblivious Valiant's algorithm (see 3.4.1 on page 15). It is a bit similar to turn around routing. On the algorithms first stage packets are routed to a randomly chosen point between the sender and the receiver by using an oblivious dimension order routing. The second stage of the algorithm works almost equally, but this time the dimensions of the network are gone through in reversed order. Deadlocks are avoided in IVAL routing by dividing router's channels to virtual channels. Full deadlock avoidance requires a total of four virtual channels per one physical channel.

**2TURN.** 2TURN algorithm itself does not have an algorithmic description. Only algorithms possible routing paths are determined in a closed form. Routing from sender to receiver with 2TURN algorithm always consists of 2 turns that will not be U-turns or changes of direction within dimensions. Just as in the IVAL routing, a 2TURN router can avoid deadlock if all router's physical channels are divided to four virtual channels.

Locality is a routing algorithm metric which is expressed as the distance a packet travels on average. This metric largely determines the end-to-end delay of packets at low load. IVAL and 2TURN algorithms improve over Valiant's

algorithm approximately 20% and 25%. 2TURN's locality is pretty near optimal. [20, 33]

**Q-Routing.** The functionality of a Q-routing algorithm is based on the network traffic statistics. The algorithm collects information about latencies and congestions, and maintains statistics about network traffic. The Q-routing algorithm does the routing decisions based on these statistics. [25]

**Odd-Even Routing.** An odd-even routing is a adaptive algorithm used in dynamically adaptive and deterministic (DyAD) Network on Chip system (see section 5.2.1). The odd-even routing is a deadlock free turn model which prohibits turns from east to north and from east to south at tiles located in even columns and turns from north to west and south to west at tiles located in odd columns. The DyAD system uses the minimal odd-even routing which reduces energy consumption and also removes the possibility of livelock. [19]

**Slack-Time Aware.** Most of the adaptive routing algorithms do not fit in systems that require definite real-time operation. In adaptive routing the latencies can vary a lot. Packets can also flow along different paths, thus they can arrive to the receiver in wrong order. The delayed packets produce interruption for example to audio or video stream. [4]

**Hot-Potato Routing.** A hot-potato routing algorithm routes packets without temporarily storing them in routers' buffer memory. Packets are moving all the time without stopping before they reach their destination. When one packet arrives to a router, the router forwards it right away towards packet's receiver but if there are two packets going to same direction simultaneously, the router directs one of the packets to some other direction. This other packet can flow away from its destination. This occasion is called misrouting. In the worst case, packets can be misrouted far away from their destination and misrouted packets can interfere with other packets. The risk of misrouting can be decreased by waiting a little random time before sending each packet. Manufacturing costs of the hot-potato routing are quite low because the routers do not need any buffer memory to store packets during routing. [17]

## 4.5 Summary

The outlines and features of the adaptive routing algorithms presented above are listed in Table 2.

Table 2: Adaptive routing algorithms.

ALGORITHM	OUTLINES	FEATURES	REFERENCES
Minimal adaptive	shortest path routing	simple	[12]
Fully adaptive	congestion avoidance	non-minimal	[12]
Congestion look ahead	congestion avoidance	fast	[24]
Turnaround / Turnback	routing in butterfly- and tree networks	uses shortest path	[1, 21, 26]
Turn Back When Possible	routing in tree networks	uses efficiently whole network	[20]
IVAL	improved turnaround routing	uses efficiently whole network	[20]
2TURN	slightly determined	efficient	[20]
Q	statistics based routing	uses the best path	[25]
Odd-Even	turn model	deadlock free	[19]
Slack-time aware	routing for real-time applications	uses network re- sources efficiently	[4]
Hot-potato	routing without buffer memories	cheap, sometimes misrouting	[17]



## 5 Router Architectures

Many research groups in different universities and institutes have proposed router architectures for Network on Chip systems. The outlines and features of these router architectures are discussed in this section. The architectures are divided into two groups: oblivious routers and adaptive routers.

### 5.1 Oblivious Routers

#### 5.1.1 Virtual Channel Router

Virtual channel router (VCR) is a router which uses source routing algorithm (see Section 3.3.2) and wormhole network flow control (see Section 2.3) with virtual channels. It is suitable for on-chip networks with two-dimensional topologies. A traditional structure of wormhole routing with virtual channels is represented in Figure 13. This router architecture has 5 input and output ports. Four of them are connected to neighbour routers and one is for router's local core. Each input port has 4 virtual channels which are demultiplexed and buffered in FIFOs. After FIFOs the virtual channels are multiplexed again to a single channel that goes to a crossbar. Routing operations in the crossbar are controlled by an arbitration unit (AU). Arbitration unit also takes care that there are no conflicts between virtual channels and that the arbitration is fair.

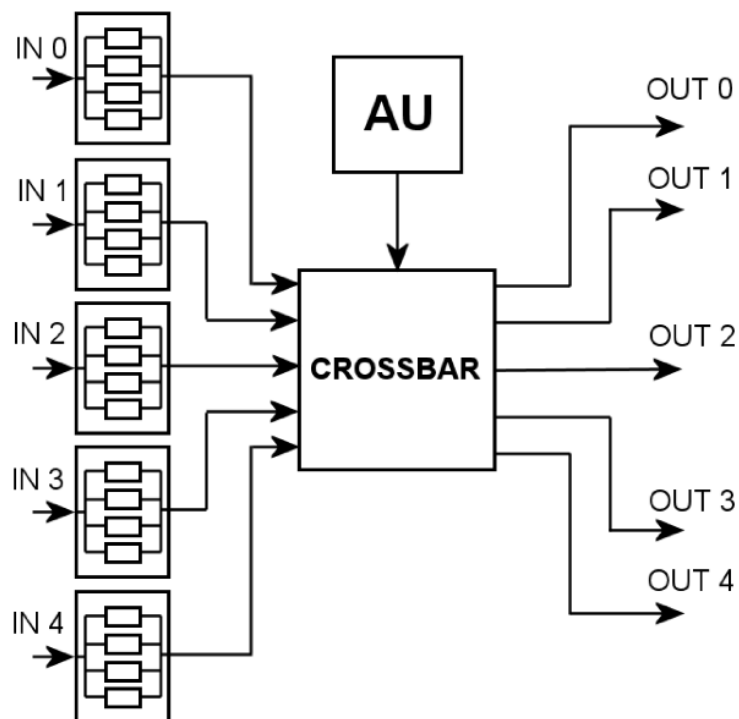


Figure 13: A virtual channel router with 5 ports and 4 virtual channels. [22]

There is also another version of virtual channel router which differs from the traditional one in that the virtual channels are not multiplexed after FIFOs in inputs. This router architecture is depicted in Figure 14. FIFOs are connected directly to the crossbar where the multiplexers for request and acknowledge signals are also integrated. In this architecture there are no conflicts at the inputs, and the arbitration unit can be replaced with small round robin arbiters (RRA) at each output port. The arbitration is deterministic and fair, and there are conflicts only at the output ports. Therefore router achieves a 100% throughput. This router suits also for transmitting a stream shaped data.

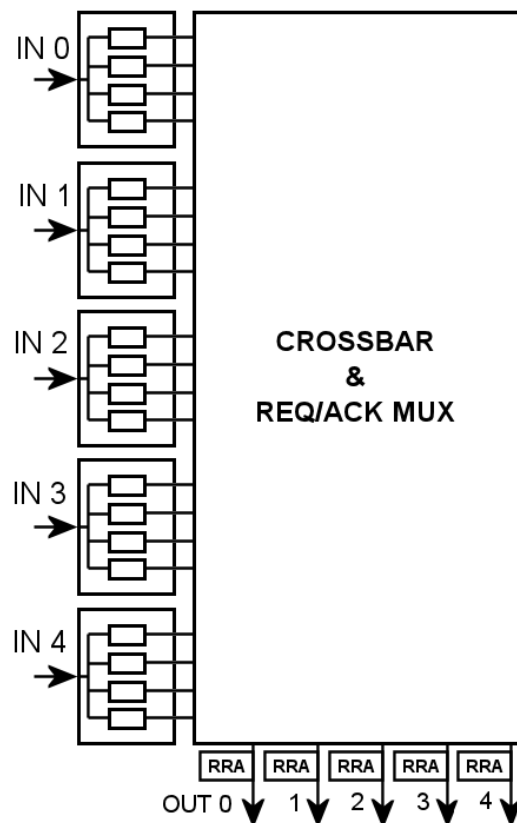


Figure 14: A virtual channel router with simplified arbitration. [22]

The cost of the latter architecture is roughly a half of the cost of the traditional one. The difference is mostly an income of the smaller arbitration unit in the latter version. The latter one is also approximately 40% faster than the traditional architecture. [22]

### 5.1.2 Xpipes

Xpipes (crosspipes or crossing pipelines) architecture uses wormhole network flow control and source routing which is in this case called the *street sign routing*. Switch structure can be kept simple because routing is deterministic and all

routing decisions are made in the beginning when a packet is send. The router architecture is a lot similar with the traditional virtual channel router architecture. Number of inputs, outputs and virtual channels as well as the network topology are design parameters to be decided by a designer. [10]

### 5.1.3 Æthereal

An Æthereal router architecture combines guaranteed throughput (GT) and best-effort (BE) routing. It uses the wormhole network flow control and the contention-free source routing algorithm. The architecture of the combined GT-BE router is depicted on Figure 15. The Æthereal uses virtual channels and shares the channels for different connections by using a time division multiplexing.

In the beginning of the routing the whole routing path is stored on the header of the packet's first flit. When the flits arrive to a router a header parsing unit extracts the first hop from the header of the first flit, moves the flits to a GT or BE FIFO and notifies the controller that there is a packet. The controller schedules flits for the next cycle. After scheduling the GT-flits, the remaining destination ports can serve the BE-flits. [16]

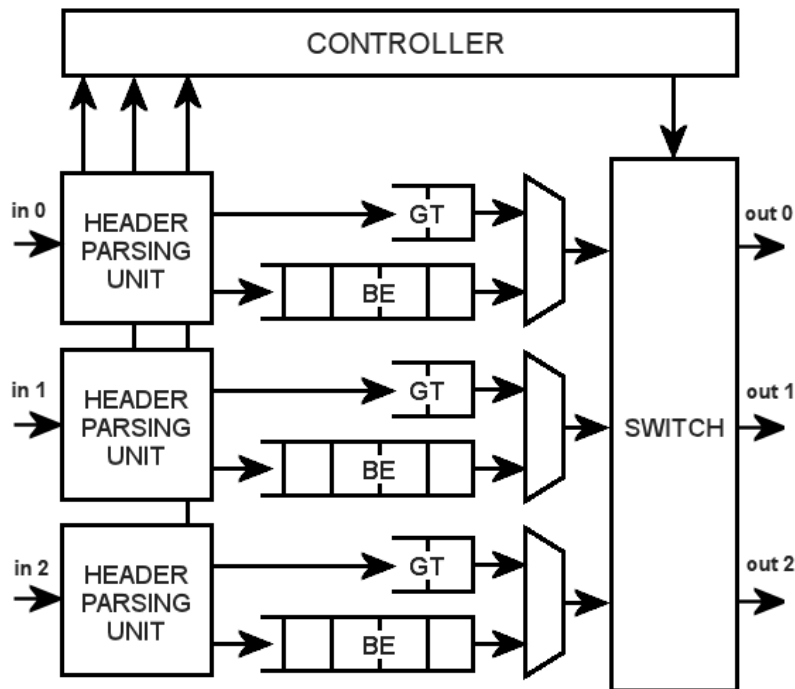


Figure 15: Æthereal router architecture. [16]

### 5.1.4 Proteo

The Proteo network consists of several sub-networks which are connected to each other with bridges. The main sub-network in the middle of the system is a ring but the topologies of the other sub-networks can be selected freely.

The layered structure of the Proteo router is depicted on Figure 16. Each layer has one input and one output port so a router with one layer is one-directional and suits only on sub-networks with simple ring topology. In more complex networks more than one layers have to be connected together.

Proteo system has two different kinds of routers, initiators and targets. The initiator routers can generate requests to the target routers while targets can only respond to these requests. The only difference between initiator and target routers is a structure of the interface. The task of the interface is to create and extract packets.

The routing on the Proteo system is destination-tag routing, where the destination address of the packet is stored on the packet's header. When a packet arrives to the input port the greeting block detects packets destination address and compares it to the address of the local core. If the addresses are equal the greeting block writes the packet to the input FIFO through the overflow checker, otherwise the packet is written to the bypass FIFO. Finally the distributor block sends packets forward from the output and bypass FIFOs. [2]

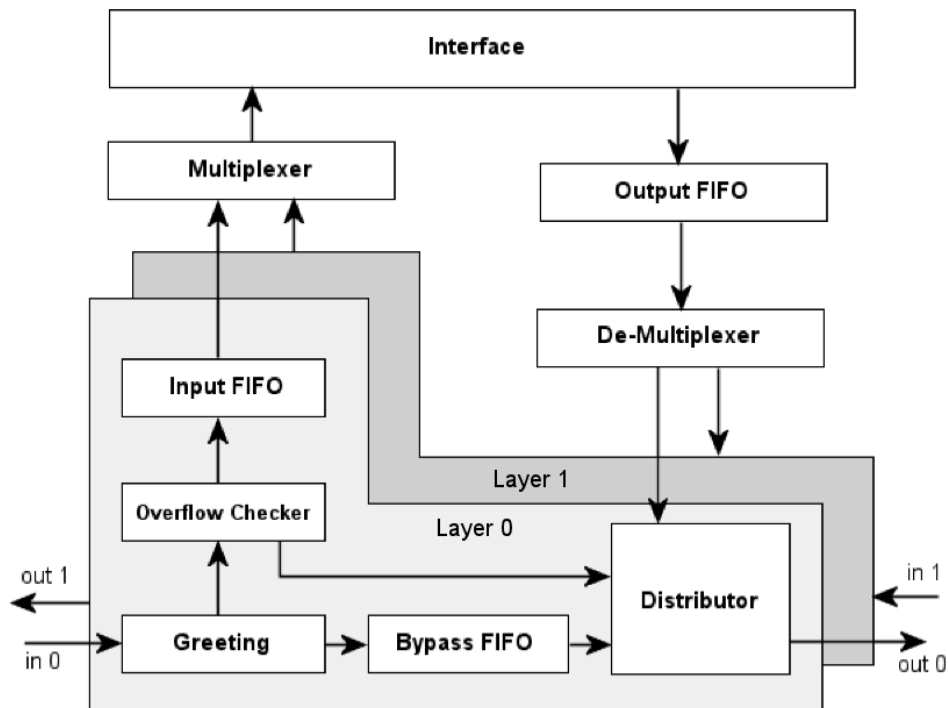


Figure 16: Two layered Proteo router. [2]

### 5.1.5 MANGO

MANGO (Message-passing Asynchronous Network on Chip providing Guaranteed services through OCP interfaces) is a clockless Network on Chip system. It uses wormhole network flow control with virtual channels and provides both guaranteed throughput (GT) and best-effort (BE) routing. Because the network is clockless the time division multiplexing cannot be used in sharing the virtual channels. Therefore some virtual channels are dedicated to BE traffic and others to GT traffic. The benefits of the clockless system are maximum possible speed and zero idle power. The MANGO router architecture (depicted in Figure 17) consists of separated GT and BE router elements, input and output ports connected to neighboring routers and local ports connected to the local IP core through network adapters which synchronize the clockless network and clocked IP core. The output port elements include output buffers and link arbiters.

The BE router routes packets using basic source routing where the routing path is stored in the header of the packet. The paths are shaped like in the XY routing. The GT connections are designed for data streams and the routing acts like a circuit switched network. In the beginning of GT routing, the GT connection is set up by programming it into the GT router via the BE router. [8]

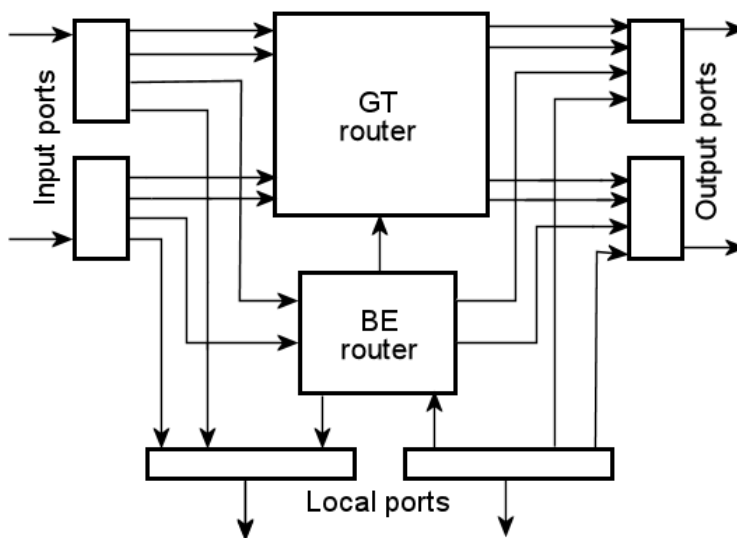


Figure 17: MANGO router architecture. [8]

### 5.1.6 SoCBUS

In contrast to most of the Network on Chip systems, the SoCBUS is based on circuit switching and store-and-forward network flow control. It uses two dimensional mesh topology. The circuit switching has some advantages over packet switching. The latency is only dependent on the distance of the sender and the

receiver, and packets always reach their destination in the same order that they were sent. The implementation of the SoCBUS is some kind of combination of the circuit and packet switching. Routing works as in circuit switching but the information is still packetized. The implementation is called as packet connected circuit (PCC).

Circuit switched routing in SoCBUS system works so that at first a request packet is routed from the sender to the receiver using destination-tag routing (see Section 3.3.3). The request packet reserves the route and then information can be transferred through it. A cancel message in the end of the routed information releases the route.

The need for buffer memories is very low in the SoCBUS system, because only the request packet has to be stored in the routers. [34]

### **5.1.7 Arteris**

Arteris NoC is the first commercial Network on Chip implementation. Most of the Arteris NoC's design parameters are user-defined so that for example network topology, routing algorithm and number of input and output ports on switches are parametrized. The Network flow control can be optimized to application needs by combining different control methods. [5]

### **5.1.8 STNoC**

STNoC is a commercial Network on Chip implementation made by STMicroelectronics. It is a simple implementation which uses wormhole network flow control, deterministic source routing and spidergon network topology. [32]

## **5.2 Adaptive Routers**

### **5.2.1 DyAD**

A dynamically adaptive and deterministic (DyAD) Network on Chip system uses dynamically both deterministic and adaptive routing algorithms to route packets. In basic situation when there are no congestions in the network the deterministic XY routing algorithm is used. Furthermore, when the network becomes congested the router switches to adaptive mode and uses the minimal odd-even routing represented in Section 4.4. Minimal version of the odd-even routing is livelock-free as well as deadlock-free which causes that the DyAD router is deadlock-free without a need for virtual channels. The network topology of DyAD is a two dimensional mesh and the wormhole network flow control is used.

The DyAD router is depicted on Figure 18. When the router receives a new header flit from some input port, the address decoder of the current input processes the destination address and sends it to the port controller. The port controller decides which output port the packet should be delivered to. Then the port

controller sends a connection request to the crossbar arbiter which controls the crossbar switch.

Each router in the DyAD network has a congestion flag, which tells that the router is congested. A router sends its flag to all its neighbor routers wherein the mode controller receives it and turns router to the adaptive mode when necessary. The advances of the DyAD are low latency in congestion free network but still good throughput in congested network. [19]

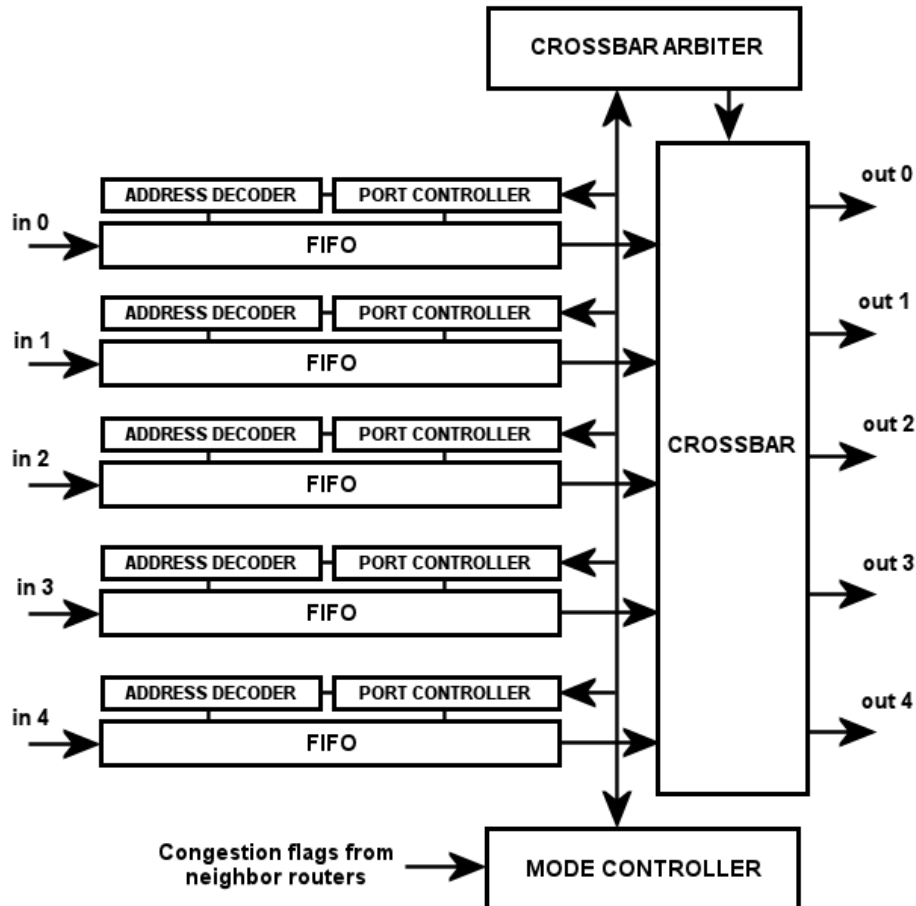


Figure 18: DyAD router. [19]

### 5.2.2 SPIN

The SPIN architecture is a scalable, packet switched, on-chip micro-network, whose network topology is fat tree and which uses wormhole network flow control. In the fat tree network the nodes are routers and leaves are terminals. The routing algorithm of the SPIN is turn around routing. The packet routing is realized as follows. First a packet flows up the tree along anyone of the available

paths. When the packet reaches a router which is a common ancestor with the destination terminal, the packet is turned around and routed to its destination along the only possible path.

The architecture of the RSPIN router, used in SPIN systems, is represented on Figure 19. There is a 4-flit buffer on each input port and two 18-flit output buffers shared between output ports. The output buffers have greater priority to use the output channels than input buffers. This reduces contention. [1]

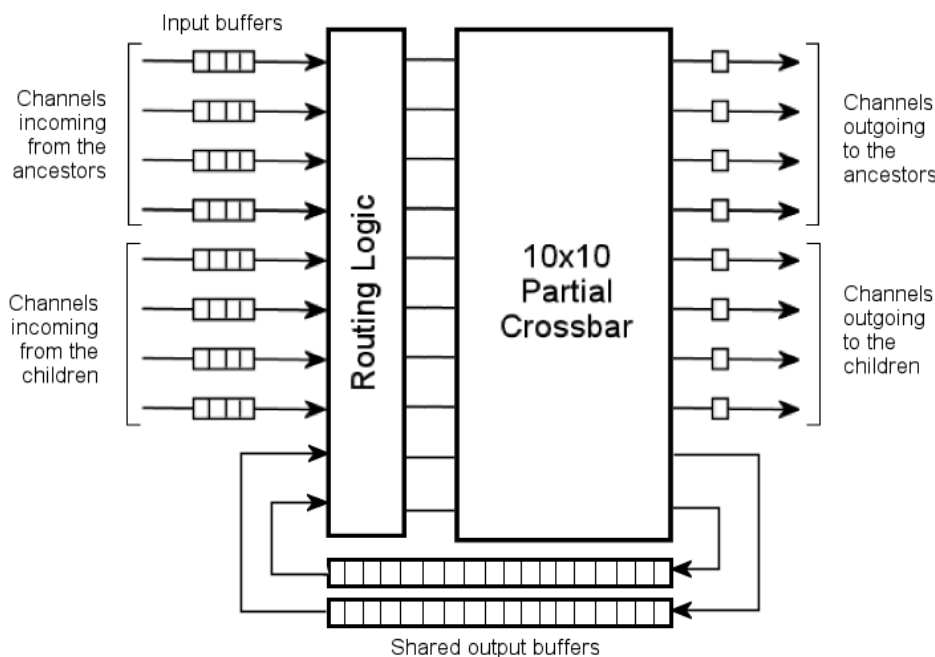


Figure 19: RSPIN router used in SPIN systems. [1]

### 5.2.3 XGFT

XGFT (eXtended Generalized Fat Tree) Network on Chip is a fault-tolerant system which is able to locate the faults and reconfigure the routers so that the packets can be routed correctly. The network is a fat tree and the wormhole network flow control is used. Besides of the traditional wormhole mechanism, there is a variant called pipelined circuit switching. If the packet's first flit is blocked, it is routed one stage backwards and routed again along some alternative path.

When there are no faults in the network, the packets are routed using adaptive turn around routing as explained above in Section 5.2.2. However, when faults are detected, the routing path is determined deterministic using source routing and so that packets are routed around faulty routers. To detect the faults there has to be some system which diagnoses the network. [21]



### 5.2.4 Nostrum

The Nostrum Network on Chip implementation is a two dimensional mesh with adaptive hot-potato routing and virtual channels. Hot-potato routing allows congestion avoidance and fault-tolerancy. There are no buffer memories or routing tables so the routers are small. [27]

### 5.3 Summary

The essential features of the router architectures discussed above are listed in Table 3. It can be noticed that some features are more common than others in these proposed router architectures. The most common network topology is mesh while fat tree topology is also used in some adaptive architectures. Wormhole network flow control as well as source routing algorithm are used in many architectures. Turn around algorithm is also used in some adaptive routers. There are only couple of architectures with other network flow control methods and routing algorithms.

Table 3: Router architectures.

ROUTER	TOPOLOGY	FLOW CTRL	ALGORITHM	SPECIAL	REF.
<i>Oblivious</i>					
VCR	2-dimensional	Wormhole	Source routing	Virtual channels	[22]
Xpipes	Any	Wormhole	Source routing	Well adaptable	[10]
Æthereal	Mesh	Wormhole	Contention free source routing	Combined GT and BE	[16]
Proteo	Ring and subnets	Wormhole	Destination-tag	Layered structure	[2]
MANGO	Mesh	Wormhole	Source routing	GT and BE traffic	[8]
SoCBUS	Mesh	Store-and-forward	Destination-tag	Circuit switching	[34]
Arteris	User-defined	User-defined	User-defined	Commercial	[5]
STNoC	Spidergon	Wormhole	Source routing	Commercial	[32]
<i>Adaptive</i>					
DyAD	Mesh	Wormhole	XY, Odd-Even	Dynamically deterministic and adaptive	[19]
SPIN	Fat tree	Wormhole	Turn around	Fault-tolerant	[1]
XGFT	Fat tree	Wormhole variant	Turn around, source routing		[21]
Nostrum	Mesh	Virtual cut-through	Hot-potato	No buffers	[27]

## 6 Conclusions

Network on Chip is a technology of future on System on Chip implementations. The NoC technology is relatively young and any of the implementations has not risen above others. There are quite few commercial applications of Network on Chip so far. However, it is expected that the NoC will be a common technology in the future.

The small size of Network on Chip circuits sets special requirements for all operations. The network technology of the Internet is very hard to straightly shrink to the NoC so the technologies should be specially adapted to the NoC. The routing algorithms presented in this report are difficult to be set in the order of superiority. Different applications need different routing algorithms. While some algorithm is suitable to one system, another algorithm works better in some other system. However, it can be generalized that in most of the cases a simple algorithm suits to simple systems while complex algorithms fit to more complex systems. Big network traffic amounts in wide complex systems need efficient traffic equalization and congestion avoidance while the most significant features in smaller systems are the low energy consumption and low latency.

Almost all proposed Network on Chip implementations are packet switched and use wormhole network flow control which is a consequence of lower latencies and smaller needs of buffer memories in contrast to other flow control methods. The most common routing algorithm is the deterministic source routing. Still there are proposed implementations using deterministic destination-tag routing and adaptive algorithms such as turn around and hot-potato routing. Furthermore the most popular network topologies are mesh and fat tree. The number of applications of the other topologies is quite few.

The most of the proposed router architectures are still deterministic. When the dimensions of the systems decrease and the systems develop towards nanoscale the need for fault-tolerant systems will be significant. Basically the adaptive implementations are more easily modified fault-tolerant than the oblivious ones. That is why the significance of adaptive implementations is expected in the future.

The Network on Chip technology develops all the time and a couple of implementations are already in commercial use.

## References

- [1] A. Adriahtenaina, H. Charlery, A. Greiner, L. Mortiez, C.A. Zeferino: *SPIN: a Scalable, Packet Switched On-chip Micro-network*. Design, Automation and Test in Europe Conference and Exhibition, 2003, p. 70–73.
- [2] M. Alho, J. Nurmi: *Implementation of interface router IP for Proteo network-on-chip*. The 6th IEEE International Workshop on Design and Diagnostics of Electronics Circuits and Systems, Poznan, Poland, 2003.
- [3] M. Ali, M. Welzl, S. Hellebrand: *A Dynamic Routing Mechanism for Network on Chip*. 23rd NORCHIP Conference, 21–22 November 2005, pages: 70–73.
- [4] D. Andreasson, S. Kumar: *Slack-Time Aware Routing in NoC Systems*. IEEE International Symposium on Circuits and Systems, 23–26 May 2005, pages: 2353–2356.
- [5] Arteris, <http://www.arteris.com/>
- [6] N. Bansal, A. Blum, S. Chawla, A. Meyerson: *Online Oblivious Routing*. Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, 2003, pages: 44–49.
- [7] T.A. Bartic, J.-Y. Mignolet, V. Nollet, T. Marescaux, D. Verkest, S. Vernalde, R. Lauwereins: *Topology adaptive network-on-chip design and implementation*. IEE Proceedings – Computers and Digital Techniques, 8 July 2005, Volume 152, Issue 4, pages: 467–472.
- [8] T. Bjerregaard, J. Sparso: *A Router Architecture for Connection-Oriented Service Guarantees in the MANGO Clockless Network-on-Chip*. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2005, Volume 2, pages: 1226–1231.
- [9] C. Bobda, A. Ahmadinia, M. Majer, J. Teich, S. Fekete, J. van der Veen: *DyNoC: A Dynamic Infrastructure for Communication in Dynamically Reconfigurable Devices*. International Conference on Field Programmable Logic and Applications, 24–26 August 2005, pages: 153–158.
- [10] M. Dall’Osso, G. Biccari, L. Giovannini, D. Bertozzi, L. Benini: *Xpipes: a Latency Insensitive Parameterized Network-on-chip Architecture For Multi-Processor SoCs*. Proceedings of the 21st International Conference on Computer Design, 13–15 October 2003, pages: 536–539.
- [11] W.J. Dally, H. Aoki: *Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels*. IEEE transactions on Parallel and Distributed Systems, 1993, Volume 4, Issue 4, pages: 466–475.

- [12] W.J. Dally, B. Towles: *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
- [13] W.J. Dally, B. Towles: *Route Packets, Not Wires: On-Chip Interconnection Networks*. Proceedings, Design Automation Conference 2001, pages: 684–689.
- [14] G. De Micheli, L. Benini: *Networks on Chips*. Morgan Kaufmann, 2006.
- [15] M. Dehyadgari, M. Nickray, A. Afzali-kusha, Z. Navabi: *Evaluation of Pseudo Adaptive XY Routing Using an Object Oriented Model for NOC*. The 17th International Conference on Microelectronics, 13–15 December 2005.
- [16] J. Dielissen, A. Radulescu, K. Goossens, E. Rijpkema: *Concepts and Implementation of the Philips Network-on-Chip*. IP-Based SOC Design, Grenoble, France, Nov 2003.
- [17] U. Feige, P. Raghavan: *Exact Analysis of Hot-Potato Routing*. 33rd Annual Symposium on Foundations of Computer Science, 24–27 October 1992, pages: 553–562.
- [18] K. Goossens, J. Dielissen, A. Radulescu: *Aetheral Network on Chip: Concepts, Architectures and Implementations*. IEEE Design & Test of Computers, 2005, Volume 22, Issue 5, pages: 414–421.
- [19] J. Hu, R. Marculescu: *DyAD – Smart Routing for Networks-on-Chip*. Proceedings, 41st Design Automation Conference, 2004, pages: 260–263.
- [20] H. Kariniemi, J. Nurmi: *Arbitration and Routing Schemes for On-chip Packet Networks*. Interconnect-Centric Design for Advanced SoC and NoC (toim: J. Nurmi, H. Tenhunen, J. Isoaho & A. Jantsch), Kluwer Academic Publishers, 2004, pages: 253–282.
- [21] H. Kariniemi, J. Nurmi: *Fault-tolerant XGFT Network-on-Chip for Multi-processor System-on-Chip Circuits*. International Conference on Field Programmable Logic and Applications, 24–26 August 2005, pages: 203–210.
- [22] N. Kavaldjiev, G.J.M. Smit, P.G. Jansen: *A Virtual Channel Router for On-chip Networks*. Proceedings, IEEE International SOC Conference, 12–15 September 2004, pages: 289–293.
- [23] K. Kim, S. J. Lee, K. Lee, H.J. Yoo: *An Arbitration Look-Ahead Scheme for Reducing End-to-End Latency in Networks on chip*. IEEE International Symposium on Circuits and Systems, 23–26 May 2005, Volume 3, pages: 2357–2360.

- [24] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, C.R. Das: *A Low Latency Router Supporting Adaptivity for On-Chip Interconnects*. Proceedings, 42. Design Automation Conference, 13–17 June 2005, pages: 559–564.
- [25] M. Majer, C. Bobda, A. Ahmadinia, J. Teich: *Packet Routing in Dynamically Changing Networks on Chip*. Proceedings, 19th IEEE International Parallel and Distributed Processing Symposium, 4–8 April 2005, page: 154b.
- [26] L.M. Ni, Y. Gui, S. Moore: *Performance Evaluation of Switch-Based Wormhole Networks*. IEEE Transactions on Parallel and Distributed Systems, 1997, Volume 8, Issue 5, pages: 462–474.
- [27] Nostrum, <http://www.imit.kth.se/info/FOFU/Nostrum/>
- [28] J. Nurmi: *Network-on-Chip: A New Paradigm for System-on-Chip Design*. Proceedings 2005 International Symposium on System-on-Chip, 15–17 November 2005, pages: 2–6.
- [29] K. Oommen, D. Harle: *Hardware Emulation of a Network on Chip Architecture Based on a Clockwork Routed Manhattan Street Network*. International Conference on Field Programmable Logic and Applications, 24–26 August 2005, pages: 727–728.
- [30] M. Pirretti, G.M. Link, R.R. Brooks, N. Vijaykrishnan, M. Kandemir, M.J. Irwin: *Fault Tolerant Algorithms for Networks-On-Chip Interconnect*. Proceedings, IEEE Computer society Annual Symposium on VLSI, 19–20 February 2004, pages: 46–51.
- [31] E. Rijpkema, K. Goossens, P. Wielage: *A Router Architecture for Networks on Silicon*. Proceedings of Progress 2001, 2nd Workshop on Embedded Systems.
- [32] STMicroelectronics. <http://www.st.com>
- [33] B. Towles, W.J. Dally, S. Boyd: *Throughput-Centric Routing Algorithm Design*. Proceedings, 15th ACM symposium on Parallel algorithms and architectures, June 2003, pages: 200–209.
- [34] D. Wiklund, D. Liu: *SoCBUS: Switched Network on Chip for Hard Real Time Embedded Systems*. Proceedings, International Parallel and Distributed Processing Symposium, 22–26 April 2003.
- [35] M. Yang, T. Li, Y. Jiang, Y. Yang: *Fault-Tolerant Routing Schemes in  $RDT(2,2,1)/\alpha$ -Based Interconnection Network for Networks-on-Chip Designs*. Proceedings, 8th International Symposium on Parallel Architectures, Algorithms and Networks, 7–9 December 2005.

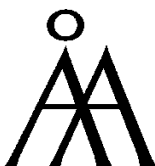
TURKU  
CENTRE *for*  
COMPUTER  
SCIENCE

Lemminkäisenkatu 14 A, 20520 Turku, Finland | [www.tucs.fi](http://www.tucs.fi)



**University of Turku**

- Department of Information Technology
- Department of Mathematics



**Åbo Akademi University**

- Department of Computer Science
- Institute for Advanced Management Systems Research



**Turku School of Economics and Business Administration**

- Institute of Information Systems Sciences

ISBN 952-12-1764-2

ISSN 1239-1891