Feasibility Analysis of Messages for On-chip Networks Using Wormhole Routing

Zhonghai Lu, Axel Jantsch and Ingo Sander
Royal Institute of Technology, Stockholm, 16440 Kista, Sweden
{zhonghai,axel,ingo}@imit.kth.se

Abstract—The feasibility of a message in a network concerns if its timing property can be satisfied without jeopardizing any messages already in the network to meet their timing properties. We present a novel feasibility analysis for real-time (RT) and nonreal-time (NT) messages in wormhole-routed networks on chip. For RT messages, we formulate a *contention tree* that captures contentions in the network. For coexisting RT and NT messages, we propose a simple *bandwidth partitioning method* that allows us to analyze their feasibility independently.

I. Introduction

Network-on-Chip (NoC) [3, 4, 10] design starts with a system specification which can be expressed as a set or sets of communicating tasks. The second step is to map these tasks onto the nodes of a NoC instance. With a mapping, application tasks running on these nodes load the network with messages, and impose timing requirements. Timely delivery of messages is essential for performance and predictability. However, routing messages in a network is inherently nondeterministic because messages experience various contention scenarios which stem from sharing buffers at routers and links between the routers. These contentions cause indeterminate delay and jitter, leading to possibly the violation of the timing constraints of the messages. It is therefore important to conduct an analysis on messages to determine their feasibility. Given a set of already scheduled messages, a message is termed *feasible* if its own timing property is satisfied irrespective of any arrival orders of the messages in the set, and it does not prevent any message in the set from meeting its timing property [2]. In general, on-chip messages can be categorized as *real-time* (RT) and *nonreal-time* (NT) messages [10]. Messages with a deterministic bound, which must be delivered predictably even under worst case scenarios, are RT messages. Messages with a probabilistic bound, which request an average response time, are NT messages.

Wormhole flow control with lanes (virtual channels) is being advocated for NoCs due to its shorter latency, greater throughput and smaller buffering requirement [3, 10]. However, few studies have been performed to analyze the message feasibility for wormhole-routed networks. For real-time messages, the lumped link model [2, 5] is a path-based model in which all the links along a message M_i's path are lumped into a single link. The message is scheduled on this link together with other competing messages. The feasibility test algorithms based on this model are efficient [2, 5]. However, due to lumping, all the competing messages must be scheduled in sequence. As a result, direct and indirect contentions are treated in the same way. Also, no concurrent use of the links on M_i's path can be taken into account. In [6], Kim et al. used a blocking dependency graph to express the contentions a message may meet and derived the message's delivery upper bound. However, this graph does not reflect the possible concurrent use of links, too.

In the paper, we present a novel feasibility analysis for both RT and NT messages on wormhole-routed networks on chip. Section II describes the communication models delivering the RT and NT messages. In Section III, we first classify messages according to the type of performance bound and timing requirements on delay or jitter. Then, for the RT messages, we formulate a contention tree that can accurately reflect contentions and link usage. Specifically, it can distinguish direct and indirect contentions and captures concurrent use of links. Finally, we use a bandwidth partitioning method to test the feasibility of RT and NT messages coexisting in the network. The experiments are described in Section IV, followed by conclusions in Section V.

II. The Communication Models

A. The Nonreal-time Communication Model

In wormhole routing, a message is divided into a number of flits (flow control units) for transmission[1]. The head flit carrying routing and sequencing information governs the route. As the head flit advances, the remaining flits follow in a pipeline fashion. The message transmission is complete when its last flit is delivered to the destination. When required resources are unavailable, the messages are blocked in place. Wormhole routing manages two types of resources: the lanes and the physical link bandwidth. In conventional wormhole routers, the shared lanes are arbitrated on First-Come-First-Serve (FCFS), and they are multiplexed over the shared link bandwidth on demand [9]. This model is fair and produces good average-case latency results. But there is no guarantee that the messages are delivered before deadline. Therefore this communication model is suitable for the delivery of NT messages. With this NT model, the average network latency T^{nt}

[1]The effect of packetization is not considered in this study.

of delivering a message with L flits is calculated by [1]:

$$T^{nt} = L/B^{nt} + HR + \omega = a + \omega \quad (1)$$

where B^{nt} is the minimum link bandwidth allocated to the message along its route; H denotes the number of hops the message passes; R is the routing delay per hop. The first two terms represent the non-contentional or base latency a, which is the lower bound on T^{nt}; ω is the average contention delay due to the message being unable to access the shared lanes and link bandwidth.

B. The Real-time Communication Model

Real-time messages must be served in such a way that the message delivery is predictable and guaranteed. Li and Mutka [7] developed a range of flow control schemes for real-time messages concerning priority mapping strategies, priority adjustment methods, and arbitration functions. In [2], based on a global priority, Preemptive Pipelined Circuit Switching for Real-Time (PPCS-RT) decouples the message delivery into two phases: path establishment and data delivery, where the path setup is preemptable. In [11], a flit-level preemption flow control is developed to resolve the priority inversion problem, i.e., a higher priority message is blocked by a lower priority message occupying shared resources. These real-time models complicate wormhole router design.

We assume a real-time (RT) message delivery model without a complicated router architecture and without a special service. All messages are globally prioritized (priority ties are resolved arbitrarily). This model arbitrates shared lanes and link bandwidth by priority. The priority, which may be assigned according to rate, deadline or laxity [5, 7], takes a small number of flits. With this RT model, assuming the same routing delay R for the head flit and other flits, the worst-case latency T^{rt} of delivering a message with L flits is given by :

$$T^{rt} = (L + L_{pri})/B^{rt} + HR + \tau = c + \tau \quad (2)$$

where B^{rt} is the minimum link bandwidth allocated to the RT message along its route; L_{pri} is the number of flits taken by the message priority. The first term counts for the transmission time of all the message flits including that occupied by the priority; the sum of the first two terms is the non-contentional latency c, which is the lower bound on T^{rt}; the last term τ is the worst-case blocking time due to contentions.

III. Feasibility Analysis

A. The Message Model and Quality Classes

We consider messages or message streams that can be characterized by four parameters $M = (S,\ p,\ D,\ j)$, where S denotes the maximum size of all the message instances; p is the message period meaning that all the inter-arrival times of the message instances are never less than p; D is the end-to-end delay constraint; j is the jitter constraint. Though the delay D is a constraint on the end-to-end communication latency, which is the sum of the latency due to the resource node T_{node} and the network T, we focus on the network latency T. The effects of T_{node} can be straightforwardly incorporated into the delay constraint resulting in a more stringent deadline.

Depending on the type of performance bound (deterministic or probabilistic) and that of timing requirement (delay or jitter), we define the Quality Class (QC) of a message, which can be viewed as an index representing the Quality of Service (QoS) requirement(s) of the message. For a probabilistic bound, we refer to constrain the bound to be an average response time. We define four quality classes as follows:

QC_1: jitter constrained, $D - j \leq T \leq D$.

QC_2: delay constrained, $T \leq D$, $j = D$.

QC_3: average jitter constrained, $D - j \leq T_{avg} \leq D$.

QC_4: average delay constrained, $T_{avg} \leq D$, $j = D$.

QC_1 and QC_2 messages are RT traffic while QC_3 and QC_4 are NT traffic. Also, QC_2 and QC_4 messages can be regarded as a special case of QC_1 and QC_3 messages when $j = D$, respectively.

B. Real-Time Messages

According to Equation (2), a feasible real-time (RT) message M_i satisfies its timing constraint:

$$\begin{array}{ll} \forall\ M_i \in QC_1 & D_i - j_i \leq c_i + \tau_i \leq D_i \\ \forall\ M_i \in QC_2 & c_i + \tau_i \leq D_i \end{array} \quad (3)$$

To estimate the worst-case latency of an RT message M_i, we must first determine all the contentions the message may meet.

In flit-buffered networks, the flits of a message M_i are pipelined along its routing path. The message advances when it receives the bandwidth of all the links along the path. The message may directly and/or indirectly contend with other messages for shared lanes and link bandwidth. M_i has a higher priority set S_i that consists of a *direct contention* set S_{D_i} and an *indirect contention* set S_{I_i}, $S_i = S_{D_i} + S_{I_i}$. S_{D_i} includes the higher priority messages that share at least one link with M_i. Messages in S_{D_i} directly contend with M_i. S_{I_i} includes the higher priority messages that do not share a link with M_i, but share at least one link with a message in S_{D_i}, and $S_{I_i} \cap S_{D_i} = \emptyset$. Messages in S_{I_i} indirectly contend with M_i. As an example, Fig. 1a shows a fraction of a network with four nodes and four messages. The messages M_1, M_2, M_3 and M_4 pass the links AB, BC, AB→BC→CD, and CD, respectively. A lower message index denotes a higher priority. The message M_1 has the highest priority, thus $S_1 = \emptyset$. For the message M_2, it directly contends with M_3, but it has a higher priority, thus $S_2 = \emptyset$. The message M_3 has a higher priority message set $S_3 = S_{D_3} = \{M_1, M_2\}$, $S_{I_3} = \emptyset$. For the message M_4, $S_{D_4} = \{M_3\}$ and $S_{I_4} = \{M_1, M_2\}$ because M_1 or M_2 may block M_3 which in turn blocks M_4.

To capture both direct and indirect contentions, we have formulated a *contention tree* defined as a directed graph G :

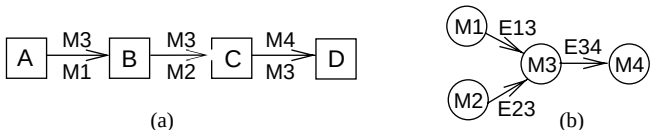

Fig. 1. Network Contentions and Contention Tree

$M \times E$. A message M_i is a node M_i in the tree, and vice versa. An edge $E_{ij}(i < j)$ directs from node M_i to node M_j, representing the direct contention between M_i and M_j. M_i is called *parent*, M_j *child*. Given a set n of RT messages, after mapping to the target network, we can build a contention tree with the following three steps:

Step 1. Sort the message set in descending priority sequence with a chosen priority assignment policy.

Step 2. Determine the routing path for each of the messages.

Step 3. Form a tree. If M_i shares at least one link with M_j where $i < j \leq n$, an edge E_{ij} is created between them. Each tree node only maintains a list of its parent nodes.

In a contention tree, a direct contention is represented by a directed edge while an indirect contention is implied by a "walk" via parent node(s). A walk is a path following directed edges in the tree. The contention tree for Fig. 1a is shown in Fig. 1b, where the three direct contentions are represented by the three edges E_{13}, E_{23} and E_{34}, and the two indirect contentions for M_4 are implied by the two walks $E_{13} \rightarrow E_{34}$ and $E_{23} \rightarrow E_{34}$ via M_4's parent node M_3. Since knowing the routing path is a priori, creating a contention tree is more suitable for deterministic routing. For adaptive routing, it is difficult to figure out the worst-case routing path.

TABLE I
MESSAGE PARAMETERS AND LATENCY BOUNDS

Message	Period p	Deadline D	Base latency c	Lat. bound
M_1	10	10	7	7
M_2	15	15	3	3
M_3	30	30	5	20
M_4	30	30	8	28

Table I shows the message parameters for Fig. 1, where the priority is assigned by rate, and deadline D equals period p. The worst-case schedules[2] for the three links are illustrated separately in Fig. 2a. The latency bounds for the four messages are also listed in Table I. We can see that all the four messages are feasible. Looking into the schedules, we can observe that (1) M_1 and M_2 are scheduled in parallel. This concurrency is in fact reflected by the *disjoint* nodes in the tree. We call two nodes *disjoint* if no single walk can pass through both nodes. For instance, M_1 and M_2 in Fig. 1b are disjoint, therefore their schedules do not interfere with each other; (2) M_3 is scheduled on the overlapped empty time slots [8, 10] and [19, 20] left after scheduling M_1 and M_2. The competed slots [1,7] and [11,18] are occupied by M_1 or M_2. This is implied in the tree where M_3 has two parents, M_1 and M_2; (3) M_4 is scheduled only after M_3 completes transmission at time 20. The indirect contentions from M_1 and M_2, which are reflected via slots [1,7] and [11,18], *propagate* via its parent node M_3. For M_3, these slots are directly competed slots. For M_4, they become indirectly competed slots. The four message schedules are individually depicted in Fig. 2b. If the concurrent use of the two links, AB by M_1 and BC by M_2, was not captured, M_3 and M_4 would be considered infeasible since M_2 would occupy the slots [8, 10] and [18, 20], levaving only three empty slots before slot 30 for M_3 and M_4.

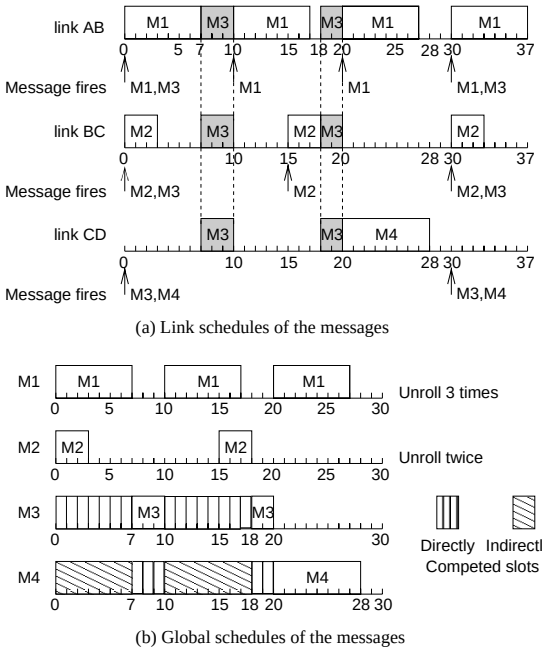

Fig. 2. Message Scheduling

In a contention tree, all levels of indirect contentions propagate via the intermediate node(s). This might be pessimistic since many of them are not likely to occur at the same time. If the number of shared lanes increases, the indirect contentions due to lane unavailability decrease. Also, a lower priority message can use the link bandwidth if a competing message with a higher priority is blocked elsewhere. To balance this pessimism, we have neglected priority inversion. As discussed in [2, 5], this problem can be alleviated by packetization.

C. Nonreal-Time Messages

According to Equation (1), a feasible nonreal-time (NT) message M_i satisfies its timing constraint:

$$\begin{array}{ll} \forall\ M_i \in QC_3 & D_i - j_i \leq a_i + \omega_i \leq D_i \\ \forall\ M_i \in QC_4 & a_i + \omega_i \leq D_i \end{array} \quad (4)$$

[2] A schedule is a timing sequence where a time slot is occupied by a message or left empty.

To analytically estimate the average contention delay ω_i is a difficult task because it is dependent on the network characteristics such as topology, routing algorithm, flow control, as well as the network communication patterns. Since this estimation is not the focus of this paper, we consider only special cases. To this end we use the closed form of contention delay [1] that Agarwal developed for random traffic k-ary d-cubes using dimension-order wormhole routing and unbounded internal buffers. For a 2D mesh network, ω_i is roughly calculated by: $\omega_i = \frac{3}{2} \cdot \frac{L_i}{B} \cdot \frac{\rho}{(1-\rho)} \cdot \frac{(H_i-1)}{H_i}$, where ρ is the network utilization calculated by $\rho = \sum_i (H_i - 1) q_i / C$, where C is the network capacity measured in the total number of network links; q_i is the probability of a network request a cycle.

Scheduling a new NT message leads to an increase in ρ. The timing constraints of the already scheduled messages must be met with the new ρ. Otherwise, the new message is infeasible.

D. Real-Time and Nonreal-Time Messages

In a network supporting both RT and NT messages, estimating the values of worst-case blocking time τ and average blocking time ω becomes more complicated due to the possible interactions while delivering both classes of messages. For example, with respect to ω, if the NT messages are allowed to use the unused bandwidth reserved by the RT messages, the RT messages may suffer from severe priority inversion problems, i.e., they may be blocked by the NT messages for an uncertain amount of time; with respect to τ, the portion of the shared resources available to the RT messages may be dynamically changing, leading to intractability. This dynamic network behavior is not in accordance with our static analysis approach. In fact, such dynamic resource sharing schemes complicate the router design; for instance, it becomes too costly for the scheduler to adjust the allocated bandwidth. Therefore we have chosen to isolate the RT and NT traffic into two disjoint virtual networks. Such a nonwork-conserving service discipline has been discussed in [12].

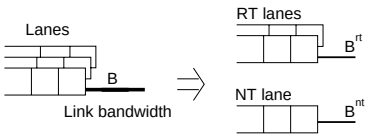

Fig. 3. Bandwidth Partitioning

Suppose the link bandwidth B is normalized to 1, then each class of traffic has a weighted portion of B, as shown in Fig. 3. Let B^{nt} and B^{rt} be the bandwidth assigned to the NT traffic and RT traffic, respectively, $B^{nt} + B^{rt} = 1$. As a result, the link bandwidth is arbitrated by weighted round robin where the weights (B^{nt} and B^{rt}) can be chosen a priori based on all types of traffic the router is designed to carry [8]. Concerning a network with uniform traffic, the same weights may be selected for all the routers. We can then apply our analysis method in Section III.B and III.C to the RT and NT traffic, respectively.

IV. EXPERIMENTS

We have implemented a feasibility test algorithm based on the contention tree for RT messages and the bandwidth partitioning scheme for coexisting RT and NT messages. Then we conducted feasibility tests on messages in a 2D 8 X 8 mesh NoC with bidirectional links (the network capacity C is $4 \times 8 \times (8-1) = 224$). The network uses wormhole flow control with dimension-order X-Y routing, which is a deterministic and deadlock-free algorithm. Lower dimension networks and deterministic routing algorithms are beneficial for NoCs in order to reduce the control complexity of the routers [4]. The purposes of our experiments are two-fold. First, we investigate how messages with a different Quality Class (QC) affect the NoC performance. Second, we examine the impact of a bandwidth partitioning on the system performance.

A message with the four parameters $(S,\ p,\ D,\ j)$ is randomly generated between a pair of nodes. The message size S including protocol overhead randomly takes a value from 32, 64, 128, and 512 in flits. For each of the message sizes, the period p takes a random value from 50λ, 100λ, 200λ, and 800λ, where $\lambda \in \{1, 2, 3\}$, respectively, and $p = D$. In this way, a longer message is likely to have a longer period. The routing delay per hop R is chosen to be 2.

The amount of traffic is *generated* given a threshold ϵ from 0.1 to 1 (normalized with the network capacity) with a step length of 0.1. For any message generated, we must ensure that the link capacity is not violated. Let the probability of a network request of an RT and an NT message M_i on any given cycle be q_i^{rt} and q_i^{nt}, respectively. With a period of p_i, $q_i^{rt} = (L_i + L_{pri,i})/p_i$ and $q_i^{nt} = L_i/p_i$. Let q_{ij} be the link bandwidth requirement of M_i on link j, $q_{ij} = q_i$. For a link j with m RT and k NT messages, the link constraint is:

$$\forall j \quad \sum_{i=1}^{m} q_{ij}^{rt} + \sum_{i=1}^{k} q_{ij}^{nt} \leq B^{rt} + B^{nt} = 1 \qquad (5)$$

If a new message generated does not lead to violate Inequality 5, the message is *offered* into the network; otherwise, it is discarded. By our traffic generation method, the *offered* traffic, which is the input of the feasibility test, is up to 62% of the generated traffic as illustrated by the dashed line in Fig. 4. Also, we treat infeasible RT and NT traffic differently. If an RT message fails the feasibility test, it will not be considered any more. In contrast, all the offered NT messages are always involved. This is because a feasibility test needs to be conducted before admitting an RT message into the network while such a test is usually not necessary for an NT message. For each ϵ, the simulation runs 50 times to steady states and reports average results of *pass ratio*, i.e. the percentage of the messages that pass the feasibility test, and of the *network link utilization* of these feasible messages. In general, the more messages that fulfill their timing constraints, the higher the performance of the system. A higher utilization may imply a lower design cost while a lower utilization may imply an over-designed network.

We designed three groups of experiments. The first two groups consider delay-constrained messages. The first (Fig. 4)

concerns only delay-constrained RT traffic (QC_2), and $B^{rt} = 1$ and $B^{nt} = 0$. An RT message with a shorter period has a higher priority. The overhead due to the priority is two flits. The second one (Fig. 5) concerns both delay-constrained RT (QC_2) and delay-constrained NT (QC_4) traffic with various values of bandwidth partitioning. The last one (Fig. 6) considers jitter-constrained traffic, i.e., QC_1 and QC_3 messages. The jitter j is set to be $0.15p$; thus the network latency of a feasible message falls in the region $[0.85p, p]$.

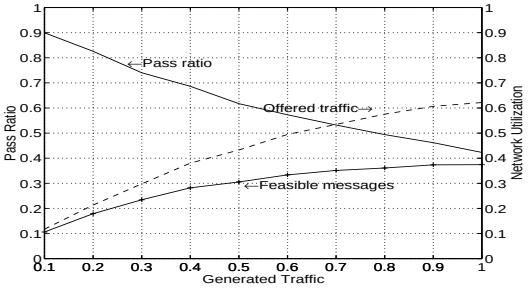

Fig. 4. Delay-constrained RT Traffic (QC_2)

In Fig. 4, as the generated traffic increases, the pass ratio decreases but the network utilization increases up to around 0.37. Closing to this point, the network is near to saturation where the network latency increases exponentially but the throughput does not improve any more [1]. Therefore the gap between the offered traffic and the feasible traffic increases rapidly. Also, the pass ratio with this uniform traffic pattern is always below 1. For a *hard* real-time system that requires 100% pass ratio, this means we need to find an application-specific mapping and our feasibility assessment can support such a mapping.

In Fig. 5, QC_2 and QC_4 messages are randomly generated; thus the number and message sizes of the RT and NT traffic have equal probability. With the value of $B^{nt} : B^{rt}$ increasing, the network tends to achieve higher pass ratio and utilization. In Fig. 6, QC_1 and QC_3 messages are also randomly generated. Comparing with Fig. 5, the corresponding pass ratio and network utilization are reduced. This is because a jitter constraint adds another condition ($D - j \leq T$) besides the deadline constraint ($T \leq D$), leading to fewer messages that pass the feasibility test.

V. Conclusion

We have presented a feasibility analysis of messages in wormhole-routed networks on chip which is a crucial step in a NoC design flow. The contention tree we formulate can accurately reflect the network contentions but relies on deterministic routing. The static bandwidth partitioning method for coexisting RT and NT messages is simple but can illustrate some non-obvious results. From the experiments conducted, we can see that the feasibility analysis is useful for performance/cost tradeoff analysis of mapping messages with different QoS requirements on a NoC.

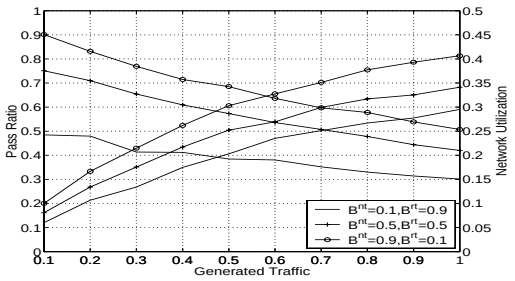

Fig. 5. Delay-constrained Traffic (QC_2-QC_4) with Bandwidth Partitioning

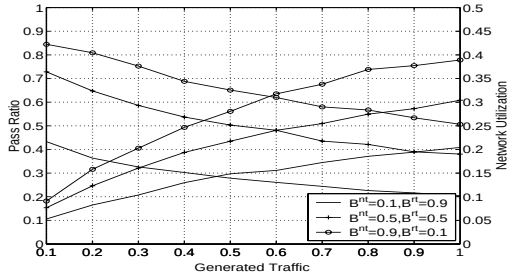

Fig. 6. Jitter-constrained Traffic (QC_1-QC_3) with Bandwidth Partitioning

Future work will investigate methods to enhance the pass ratio and/or network utilization by combining the feasibility assessment with task-to-node mappings.

References

[1] A. Agarwal. Limits on interconnection network performance. *IEEE Transactions on Parallel and Distributed Systems*, 2(4):398–412, October 1991.

[2] S. Balakrishnan and F. Özgüner. A priority-driven flow control mechanism for real-time traffic in multiprocessor networks. *IEEE Transactions on Parallel and Distributed Systems*, 9(7):664–678, July 1998.

[3] L. Benini and G. D. Micheli. Networks on chips: A new SoC paradigm. *IEEE COMPUTER*, (1):70–78, 2002.

[4] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *DAC*, 2001.

[5] S. L. Harry and F. Özgüner. Feasibility test for real-time communication using wormhole routing. *IEE Proceedings of Computers and Digital Techniques*, 144(5), 1997.

[6] B. Kim, J. Kim, S. Hong, and S. Lee. A real-time communication method for wormhole switching networks. In *Proceedings of International Conference on Parallel Processing*, pages 527–534, Aug. 1998.

[7] J.-P. Li and M. W. Mutka. Real-time virtual channel flow control. *Journal of Parallel and Distributed Computing*, 32(1):49–65, 1996.

[8] J. W. S. Liu. *Real-time Systems*. Prentice Hall, 2000.

[9] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer*, 26(2):62–76, February 1993.

[10] E. Rijpkema et al. Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip. In *DATE*, Mar. 2003.

[11] H. Song, B. Kwon, and H. Yoon. Throttle and preempt: a flow control policy for real-time traffic in wormhole networks. *Journal of Systems Architecture*, 45(8), Feb. 1999.

[12] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10), Oct. 1995.