ON THE COMMUNICATION LATENCY OF WORMHOLE ROUTED INTERCONNECTION NETWORKS

A. SHAHRABI[1], M. OULD-KHAOUA[2]

[1]*School of Computing and Mathematical Sciences*
Glasgow Caledonian University
Glasgow G4 0BA
U.K.
[2]*Department of Computing Science*
University of Glasgow
Glasgow G12 8RZ
U.K.

Abstract: Several analytical models have been proposed in the literature for wormhole-routed multicomputers. However, all these models have been discussed in the context of unicast communication and there has been comparatively little activity in the area of analytical modelling of collective communication algorithms like broadcast. To our best knowledge, this paper presents the first analytical model to predict latency of unicast and broadcast messages in wormhole-routed hypercubes with deterministic routing. Results obtained through simulation experiments show that the model exhibits a good degree of accuracy in predicting message latency under different working conditions.
Keywords: Multicomputers, Interconnection Networks, Wormhole Routing, Broadcast Operation, Performance Modelling.

1. INTRODUCTION

It is widely recognised that one of the most critical components of any multicomputer is the interconnection network used to connect the processing elements together. The hypercube has been one of the most popular such networks for many years due to desirable properties such as regularity, symmetry, recursive structure and logarithmic diameter. The Intel iPSC/2 [Nugent S.F. 1988] and SGI Origin [Laudon J. and Lenoski D. 1997] are examples of practical systems that are based on the hypercube topology.

Current routers reduce message latency by using wormhole switching (often called wormhole routing). In this switching technique, a message is divided into elementary units called flits, each composed of a few bytes for transmission and flow control. The header flit governs the route and the remaining data flits follow it in a pipelined fashion. If a channel transmits the header of a message it must transmit all the remaining flits of the same message before transmitting flits of another message. When the header is blocked, the data flits are blocked in-situ. Throughput in wormhole-routed networks can be increased by organising the flit buffers associated with a given network channel into several virtual channels [Dally W.J. 1992]. These virtual channels are allocated independently to different messages and compete with each other for the physical bandwidth. This de-coupling allows messages to bypass each other in the event of blocking, using network bandwidth that would otherwise be wasted.

The traffic distribution exhibited by parallel applications is an important factor that strongly affects network performance [Duato J., Yalamanchili S. and Ni L. 1997]. Unicast communication involves only two nodes: the source and destination. The uniform traffic pattern is a typical example of unicast communication, which has been widely studied when analysing network performance [Abraham and Padmanabhan, 1989], [Y. Boura, C.R. Das, T.M. Jacob , 1994], [Dally W.J. 1990], [Draper J.T. and Ghosh J. 1994], [Ould-Khaoua M. 1999]. However, broadcast communication, the global delivery of a single message originating from a given source to all network nodes, is important in many real-world parallel applications [Johansson S.L. and Ho C.T. 1989]. For instance, broadcast communication is often needed in scientific computations to distribute large data arrays over system nodes in order, for example, to perform various data manipulation operations. Furthermore, it is required in control operations such as global synchronisation. In the distributed shared-memory paradigm, broadcast communication is used to support the shared data

invalidation and updating procedures required for cache coherence protocols [Protic J., Tomasevic M. and Milutinovic V. 1997].

Many algorithms have been proposed for broadcast communication in wormhole-routed networks over the past few years [McKinley P. and Trefftz C. 1993], [McKinley P. et al, 1994], [Panda D., Singal S. and Kesavan R. 1999]. Among these, unicast-based broadcast algorithms have been widely used in practical systems due to their simplicity and ease of implementation [Duato J., Yalamanchili S. and Ni L. 1997], [McKinley P. et al, 1994]. These rely on the routing algorithm employed for unicast communication to route broadcast messages, and consequently do not require any changes to router hardware [Malumbres M.P., Duato J. and Torrellas J. 1996]. Several software libraries for supporting unicast-based broadcast communication have recently been developed. Broadcast communication has also been included as part of the collective communication routines in the Message Passing Interface (MPI) standard proposal [Dongarra J. *et al.* 1993]. However, when investigating a new algorithm for a collective communication operation, it is vital that its precise scope is determined and that it is evaluated with accurate modelling of the underlying routing scheme and communication mechanisms so that a clear understanding of the factors that affect its potential performance emerges. Analytical modelling offers a cost-effective and versatile tool that can help designers to assess the performance merits of broadcast algorithms to ensure successful introduction in future multicomputers.

The analytical modelling of wormhole-routed networks is well investigated, e.g. [Y. Boura, C.R. Das, Jacob T.M. 1994], [Dally W.J. 1990], [Draper J.T. and Ghosh J. 1994], [Ould-Khaoua M. 1999]. However, all these models have been discussed in the context of unicast communication and there has been comparatively little activity in the area of analytical modelling of collective communication algorithms, like broadcast. The only works considering both unicast and broadcast communication have recently been reported in [Shahrabi A. et al, 2000]. In that model, the performance of wormhole-routed hypercubes with adaptive routing has been investigated. However, despite the fact that most proposed broadcast algorithms [Lin X., McKinley P. and Ni L. 1994],[McKinley P. and Trefftz C. 1993],[McKinley P. et al, 1994] have been based on deterministic routing and that this routing approach has been suggested by the research community well before adaptive routing, to our best knowledge, no analytical model has yet been suggested for this routing scheme.

This paper presents the first analytical model to compute message latency in wormhole-routed hypercubes with deterministic routing in the presence of broadcast communication. The broadcast algorithm considered in this study is based on the well-known SBT broadcast algorithm with both unicast and broadcast messages routed according to deterministic routing. Modelling of deterministic routing implies a totally different approach from that used for adaptive routing in [Shahrabi A. et al, 2000] since the two routing algorithms impose totally different restrictions on the way messages visit network channels.

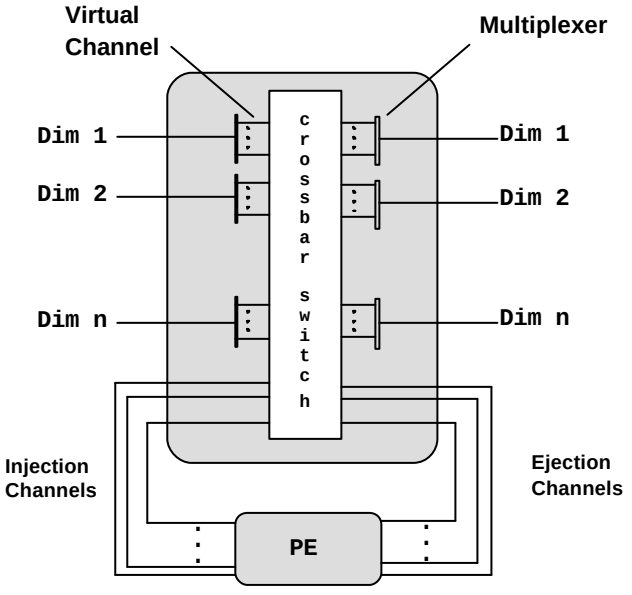

Figure 1: Node structure in the hypercube

The rest of the paper is organised as follows. Section 2 reviews some preliminary background that will be useful for the subsequent sections. Section 3 describes the analytical model while Section 4 validates it through simulation. Finally, Section 5 concludes this study.

2. PRELIMINARIES

This section briefly describes the node structure in the hypercube and the algorithm for broadcast communication.

2.1. Node Structure

The n-dimensional hypercube, has $N = 2^n$ nodes. Each node can be identified by an n-bit address $x = x_n x_{n-1} \cdots x_1$. A node with address $x = x_n x_{n-1} \cdots x_1$ is connected to the node $x' = x'_n x'_{n-1} \cdots x'_1$, $0 \le x_i, x'_i \le 1$, if and only if there exists i, $(1 \le i \le n)$, such that $x_i \ne x'_i$ and $x_j = x'_j$ $(1 \le j \le n,\ i \ne j)$.

Each node consists of a processing element (PE) and a router, as shown in Figure 1. The PE contains a processor and some local memory. A node is linked to its neighbouring nodes via *n* input and *n* output channels. Each PE is connected to the network through injection and ejection channels used by the PE to inject messages into or eject messages from the network. Messages generated at a source node to be injected into the network are placed in the local queue. Messages at the destination node are transferred to the local PE through one of the ejection channels. Each input/output channel has *V* associated virtual channels, each of which has its own flit buffers. The input and output channels are connected by a crossbar switch, which can simultaneously connect multiple inputs to multiple outputs in the absence of channel contention.

Broadcast algorithms reported in the literature have been discussed in the context of two router structures, namely the multiple-port and single-port models [Duato J., Yalamanchili S. and Ni L. 1997], [McKinley P., Tsai Y. and Robinson D.F. 1995]. The former enables copies of a broadcast message to be injected into the network through different output channels concurrently, while the latter injects them sequentially one at a time. This study focuses on the multiple-port model, but with a few simple modifications, it can be easily adapted to the single port case.

2.2. The Broadcast Algorithm

Existing broadcast algorithms are founded on either unicast-based or multidestination-based approach. Unicast-based algorithms are widely used in practice and are implemented as a sequence of unicast message exchanges as existing wormhole-routed systems support unicast communication only. Separate addressing is a simple scheme that uses the unicast-based approach [Duato J., Yalamanchili S. and Ni L. 1997], [McKinley P., Tsai Y. and Robinson D.F. 1995]. A source node in this scheme sends a copy of its message to every destination node involved in the collective communication. However, the main drawback is the waste of network bandwidth due to the excessive generated traffic. To overcome this, a tree structure is used where a source node sends the message to a subset of the destinations which they then participate recursively (forming a tree) by re-transmitting copies of the message to the remaining destination nodes [Duato J., Yalamanchili S. and Ni L. 1997], [McKinley P., Tsai Y. and Robinson D.F. 1995]. A major disadvantage of the unicast-based approach is the high increase in communication latency due to the large number of start-ups, which accounts for the

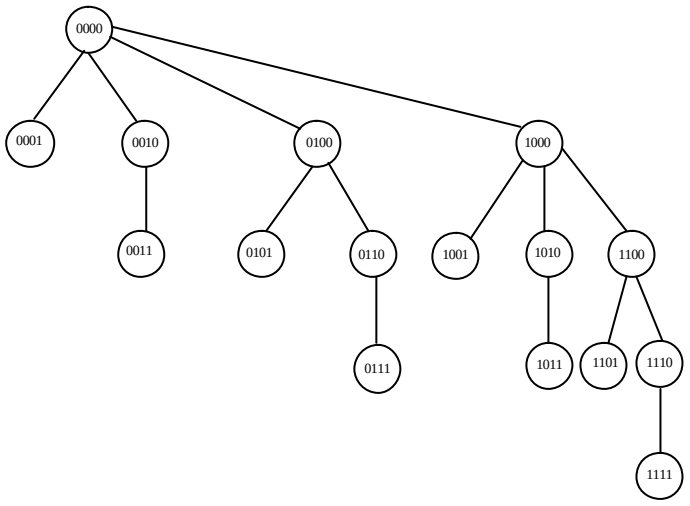

Figure 2: A broadcast spanning tree of a 4-dimensional hypercube originating from node 0.

amount of time incurred at a node when preparing a message for injection into the network.

In order to reduce the number of start-ups, Lin et al [Lin X., McKinley P. and Ni L. 1994] and Panda et al. [Panda D., Singal S. and Kesavan R. 1999] have proposed a modified router structure to support multidestination-based broadcast algorithms. An important feature of the multidestination approach is that a message can be delivered to multiple destinations with the same start-up latency as when a message is sent to a single destination. Multidestination-based algorithms rely on finding one or a few paths that can be shared by a set of destinations. Some multidestination-based broadcast algorithms use a unique path, e.g. a Hamilton path [Lin X., McKinley P. and Ni L. 1994], while others try to reduce the path length by finding a compromise between the separate addressing and unique path multidestination-based schemes. In the latter case, destination nodes are divided into subsets and a few separate copies of the broadcast message are created at the source, each of which then traverses a path covering some subset of nodes.

The multidestination -based approach suffers from several inefficiencies [Malumbres M.P., Duato J. and Torrellas J. 1996]. This is especially the case for broadcast communication where the number of destinations increases considerably. When multiple paths are used, several message copies have to be generated, each requiring a preparation time to order the destination addresses. When a unique path is used, the size of the message increases with the system size, consequently increasing both the preparation and transmission time. Moreover, the multidestination-based approach must be supported in hardware, and this has proven to be the main reason that has delayed its adoption in practical systems since most existing parallel computers support only point-to-point message transmission in hardware [Malumbres M.P., Duato J. and Torrellas J. 1996].

Our present study focuses on a unicast-based broadcast algorithm that produces a spanning binomial tree based on the concept of recursive doubling; a spanning tree is a connected graph that spans the nodes of the graph, forming a tree with no cycles. To broadcast a message, a node needs to transmit the message along a spanning tree rooted at its own location. Figure 2 shows the spanning tree in a 4-dimensional hypercube where the node 0 is the source node of the broadcast operation. Using this algorithm, the number of start-ups increases logarithmically with the number of nodes. Each node in the system will receive the broadcast message and generate new copies to send them to its own nearest neighbors. Assuming a multiple-port router model, the algorithm guarantees that every node will receive the message exactly once and in no later than n time steps.

Abraham and Padmanabhan [Abraham and Padmanabhan, 1989] have shown that when the branches of the broadcast tree are constructed in the same order (i.e., in an increasing order of network dimensions) the number of messages that cross each channel varies severely, resulting in an unbalanced traffic on network channels. To overcome this problem they have suggested assigning a different dimension as a base for every new broadcast tree. The base dimension can be selected at random or in a round-robin fashion. As has been shown in [Abraham and Padmanabhan, 1989], this improves the traffic balance in the network, and achieves higher throughput. The rest of this paper describes an analytical model for computing the broadcast latency in wormhole-routed multiport hypercubes, using the unicast-based broadcast approach that incorporates the suggestion of [Abraham and Padmanabhan, 1989]. Hereafter we will refer to this algorithm as the broadcast algorithm.

3. THE PROPOSED MODEL

This section describes an analytical model for computing the unicast and broadcast latency in the wormhole-routed hypercube. The proposed model is based on the following assumptions, which are commonly accepted in the literature [Abraham and Padmanabhan, 1989], [Y. Boura, C.R. Das, T.M. Jacob , 1994], [Ould-Khaoua M. 1999].

a) There are two types of messages in the network: "broadcast" and "unicast". A broadcast message is delivered to every node in the network using the broadcast algorithm described in Section 2-2. A unicast message is sent to other nodes in the network with equal probability. When a message is generated in a given source node, it has a finite probability b of being a broadcast message and probability $(1-b)$ of being unicast. When b = 1, all the generated traffic is broadcast, and when b = 0 the traffic pattern is purely uniform. A similar traffic model has also already been used in [Abraham and Padmanabhan, 1989].

b) Nodes generate traffic independently of each other, following a Poisson process with a mean rate of l_g messages/cycle. The mean generation rate of the broadcast messages is $l_{s_b} = b\, l_g$ and that of unicast is

$l_{s_u} = (1-b)\, l_g$.

c) Both broadcast and unicast messages are routed according to deterministic routing. In this routing approach, messages visit network dimensions in a strict order to avoid deadlocks. Let the dimensions be numbered from 1 to *n*, and messages visit higher-numbered dimensions first.

d) V ($V \geq 2$) virtual channels are used per physical channel.

e) Message length is M flits, each of which is transmitted in one cycle across the physical channel.

f) A local queue in a given source node has infinite capacity. Moreover, messages are transferred to the local PE as soon as they arrive at their destinations.

g) All messages (including unicast, broadcast or replicated messages) experience a start-up latency of Δ cycles, which accounts for the amount of time required by a node when preparing a message for injection into the network. The start-up latency is a constant value and varies from one practical system to another [Duato J., Yalamanchili S. and Ni L. 1997].

3.1. The Broadcast Latency

The broadcast latency refers to the elapsed time from when a source node sends the first copy of its broadcast message to a subset of destination nodes until the last destination in the network receives a copy. Many existing studies [Abraham and Padmanabhan, 1989], [McKinley P. et al, 1994], [Panda D., Singal S. and Kesavan R. 1999] have used the broadcast latency as a metric to assess the merits of different broadcast algorithms because of its great influence on the overall system performance.

The broadcast algorithm guarantees that each node in the network receives a copy of the broadcast message in no longer than *n* broadcast steps, corresponding to the height of the broadcast tree, as depicted in Fig. 2. The broadcast latency is composed of *n* latencies, each of which accounts for the time to send a broadcast message one step down in the tree. Let us refer to the broadcast message that crosses from one level of the broadcast tree as "one-step broadcast message" and let $\overline{L}_b$ denote the corresponding mean latency. Then, the mean broadcast latency can be written as

$$Latency = n(\overline{L}_b + \Delta) \quad (1)$$

where Δ denote the start-up latency. The mean latency of a one-step broadcast message, $\overline{L}_b$, is composed of the mean network latency, $\overline{S}_b$, i.e. the time required to make one hop in the network, and the mean waiting time seen by the message at the source node, $\overline{W}_s$, before entering the network. However, to model the effects of virtual channel multiplexing the mean latency of the one-step broadcast message has to be scaled by a factor, $\overline{V}$, representing the average degree of virtual channels multiplexing, that takes place at a given physical channel. Therefore, we can write $\overline{L}_b$ as

$$\overline{L}_b = (\overline{S}_b + \overline{W}_s)\, \overline{V} \quad (2)$$

Before describing how to determine the quantities $\overline{S}_b$, $\overline{W}_s$, and $\overline{V}$, we determine first the traffic rate on a given network channel, l_c, and then the mean service time of a channel at every individual dimension, $\overline{S}_i$ ($1 \leq i \leq n$).The appendix gives a summary of the notation used to develop the analytical model.

A. Calculation of the traffic rate on a given channel **(l_c):** Given that the destinations for unicast messages are uniformly distributed across the network and that all nodes in the network have the same probability of being a source node for a broadcast operation, messages arrive at network channels at a uniform rate. According to the broadcast algorithm, a broadcast message is replicated at various stages in the spanning tree. A replicated message is put in the local queue of the node, to be injected later across the required output channel. So, a source node generates messages with three different rates: unicast messages with a rate of $l_{s_u} = (1-b)\, l_g$, broadcast messages with a rate of $l_{s_b} = b\, l_g$, and replicated messages with a rate of l_{s_r}, which is determined as follows. Given that a source node has generated a broadcast message, the probability that a particular node in the network will replicate the broadcast message and deliver copies to its neighbouring nodes is $(2^{n-1}-1)/(2^n-1)$. Since there are (2^n-1) other nodes in the network and the generation rate of broadcast messages is $l_{s_b} = b\, l_g$, the rate of replicated messages originating from a given node is given by:

$$l_{s_r} = (2^{n-1}-1)\, l_{s_b} = (2^{n-1}-1) b\, l_g \quad (3)$$

Consider now an output channel. The traffic rate, l_c, on the channel consists of the three different

traffic rates. Thus,

$$l_c = l_{c_u} + l_{c_b} + l_{c_r} \quad (4)$$

where l_{c_u}, l_{c_b}, and l_{c_r} represent the traffic rates of the unicast, broadcast and replicated messages, respectively. These rates are computed as follows. To compute l_{c_u}, consider a generated unicast message that needs to cross *i* dimensions to reach its destination $(1 \le i \le n)$. The number of nodes that the message can reach after making *i* hops is $\binom{n}{i}$. Therefore, the probability, p_i, that a unicast message crosses *i* dimensions to reach its destination is given by

$$p_i = \frac{\binom{n}{i}}{2^n - 1} \quad (5)$$

The average number of dimensions that a unicast message crosses to reach its destination can be written as

$$\bar{d} = \sum_{i=1}^{n} i \quad p_i = \frac{n}{2}\frac{N}{N-1} \quad (6)$$

Since a router in the hypercube has *n* output channels and a node generates, on average, $l_{s_u} = (1-b)\,l_g$ unicast messages in a cycle, then the traffic rate, l_{c_u}, of unicast messages received by each channel in the network is simply [Y. Boura, C.R. Das, T.M. Jacob , 1994]

$$l_{c_u} = \frac{(1-b)\,l_g \bar{d}}{n} \quad (7)$$

A given source node generates broadcast messages with a rate $l_{s_b} = b\,l_g$. Since a copy of the broadcast message has to be sent to the *n* neighbouring nodes through the *n* output channels, the rate of broadcast traffic on a given channel is given by

$$l_{c_b} = b\,l_g \quad (8)$$

In order to compute the traffic rate, l_{c_r}, due to replicated broadcast messages we need to know the mean number of replications that a given node performs in a broadcast operation. After the source node sends its broadcast message to its *n* neighbours, each neighbour replicates the message (n-1) times, and sends, in turn, a copy to each of its neighbours. The subsequent nodes replicate the message (n-2), (n-3), …, 0 until the messag reaches all the nodes. The number of replication varies from one node to another depending the node position in the broadcast tree, as shown in Fig. 2. So, the probability that a broadcast message is replicated *i* times $(0 \le i \le n-1)$ when it reaches an intermediate node is given by

$$P_{r_i} = \frac{2^{n-i-1}}{2^n - 1} \quad (9)$$

Hence, the mean number of replication of a broadcast message in a given node can be expressed as

$$\bar{w} = \sum_{i=0}^{n-1} i \quad \frac{2^{n-i-1}}{2^n - 1} \quad (10)$$

Given that a replicated message can be sent over output channel with equal probability, the traffic rate of replicated messages on each channel is given by

$$l_{c_r} = \frac{\bar{w}}{n} l_{s_r} = \frac{\bar{w}}{n}(2^{n-1} - 1) b\, l_g \quad (11)$$

B. Calculation of the mean service time of a channel at dimension i ($\bar{S}_i$): Since every message sees a different service time when it crosses a different dimension, we should determine first the mean service time of a channel at every individual dimension. Moreover, one-step broadcast and unicast messages see different network latencies as they cross a different number of dimensions to reach their destinations. A one-step broadcast message sees a mean network latency, $\bar{S}_{b_i}$, when crossing dimension *i*, whereas a unicast message sees a mean network latency $\bar{S}_{u_i}$. As a result, the mean service time seen by an arbitrary message considering broadcast and unicast possibilities at a channel at dimension *i* with their appropriate weights, is given by

$$\bar{S}_i = \frac{l_{c_b} + l_{c_r}}{l_c}\bar{S}_{b_i} + \frac{l_{c_u}}{l_c}\bar{S}_{u_i} \quad (12)$$

In what follows, we determine the service time of a channel at dimension *i* for a boadcast and unicast messages. The mean channel service time seen by a one-step broadcast message at dimension *i*, $\bar{S}_{b_i}$, consists of two parts: one is the delay due to the actual message transmission time, and the other is due to blocking in the network. Since a one-step broadcast message makes only one hop to reach the next destination node, $\bar{S}_{b_i}$ can be written as

$$\bar{S}_{b_i} = M + B_{b_i} \quad (13)$$

where *M* is the message length and B_{b_i} is the mean blocking time seen by the message as it crosses a channel at dimension *i*. Since the one-step broadcast message makes only one hop, it can use only one $\bar{S}_{b_i}$ specific output channel to reach its destination. As a result, the message suffers blocking when all the virtual channels belonging to the required output channel are busy. Furthermore, the message sees the mean waiting time, $\bar{W}_{c_i}$, to acquire a virtual channel at dimension *i*. Let P_{v_i} denote the probability that *v*

virtual channels at dimension i are busy (P_{v_i} is calculated below). Given that a one-step broadcast message is blocked when all the V virtual channels at the required output channel are busy, the mean blocking time, B_{b_i}, can be written as

$$B_{b_i} = R_{V_i} \; W_{c_i} \tag{14}$$

To determine the mean waiting time to acquire a virtual channel at dimension i, $\overline{W}_{c_i}$, in the event of blocking, a physical channel is treated as an M/G/1 queue with a mean waiting time of [Kleinrock L. 1975]

$$\overline{W}_{c_i} = \frac{r_i \;\; \overline{S}_i (\; 1 + C_{\overline{S}_i}^2 \;)}{2 \; (\; 1 - r_i \;)} \tag{15}$$

$$r_i = l_c \;\; \overline{S}_i \tag{16}$$

$$C_{\overline{S}_i}^2 = \frac{s_{\overline{S}_i}^2}{\overline{S}_i^{\,2}} \tag{17}$$

where l_c is the traffic rate on a network channel, $\overline{S}_i$ is the mean service time of a channel at dimension i, and $s_{\overline{S}_i}^2$ is the variance of the service time distribution. While l_c and $\overline{S}_i$ are given by equations 10, and 19, respectively, to compute the other quantity, $s_{\overline{S}_i}^2$, we follow a suggestion of Draper and Ghosh [Draper J.T. and Ghosh J. 1994] to ease the development of our model while maintaining accuracy in predicting message latency. The minimum service time of a channel at the lowest dimensions is equal to the message length. Moreover, the minimum service time of a channel at other dimension is equal to the service time of a channel at the next dimension, according to the routing algorithm. Therefor, the variance of the service time distribution can be approximated as

$$s_{\overline{S}_i}^2 = \begin{cases} (\overline{S}_i - M)^2 & if \;\; i = 0 \\ (\overline{S}_i - \overline{S}_{i-1})^2 & otherwise \end{cases} \tag{18}$$

As a result, the mean waiting time becomes

$$\overline{W}_{c_i} = \begin{cases} \dfrac{l_c \;\; \overline{S}_i^{\,2} \; (\; 1 + \dfrac{(\overline{S}_i - M)^2}{\overline{S}_i^{\,2}})}{2 \; (\; 1 - l_c \;\; \overline{S}_i)} & if \;\; i = 0 \\ \dfrac{l_c \;\; \overline{S}_i^{\,2} \; (\; 1 + \dfrac{(\overline{S}_i - \overline{S}_{i-1})^2}{\overline{S}_i^{\,2}})}{2 \; (\; 1 - l_c \;\; \overline{S}_i)} & otherwise \end{cases} \tag{19}$$

Let $\Psi_{a \to d}^i$ denote the average occupation time of a channel at dimension i by a unicast message originating from node a and destined for node d. The occupation time of a channel at dimension i for a message that does not cross dimension i is zero, obviously, while for those messages that cross dimension i, it depends on the number of next dimensions still to be visited. As a result, $\Psi_{a \to d}^i$ can be written as

$$\Psi_{a \to d}^i = \begin{cases} 0 & if \;\; a_i = d_i \\ \Gamma_{a \to d}^i & otherwise \end{cases} \tag{20}$$

where $\Gamma_{a \to d}^i$ is the required time for a message, which is originated from node a and destined for node d, to cross the network from dimension i to its destination and is given by

$$\Gamma_{a \to d}^i = \begin{cases} M & if \;\; i = 0 \\ \Gamma_{a \to d}^{i-1} & if \;\; a_i = d_i \\ (1 + W_{C_i} P_{V_i}) + \Gamma_{a' \to d}^{i-1} & otherwise \end{cases} \tag{21}$$

$$(a' = a \;\; except \;\; for \;\; a_i' = d_i)$$

To calculate the mean service time of a channel at dimension i for a unicast message, $\overline{S}_{u_i}$, we consider sending a message from a specific node, say node 0, to all other nodes in the network. Averaging over the occupation times of the channel which is occupied by those unicast messages that cross dimension i yields the mean service time of a channel at dimension i for a unicast message. As only 2^{n-1} of these messages cross dimension i, according to the routing algorithm, $\overline{S}_{u_i}$ can be written as

$$\overline{S}_{u_i} = \frac{\sum_{m=1}^{2^n - 1} \Psi_{0 \to m}^i}{2^{n-1}} \tag{22}$$

The above equations reveal that there exist several inter-dependencies between the different variables of the model. For instance, equation 12 shows that $\overline{S}_i$ is a function of $\overline{S}_{b_i}$ and $\overline{S}_{u_i}$ while equations 13 and 22 show that $\overline{S}_{b_i}$ and $\overline{S}_{u_i}$ are functions of $\overline{S}$. Since obtaining closed-form expressions for such interdependencies is generally difficult, the different variables of the model are computed using iterative techniques for solving equations [Ould-Khaoua M. 1999].

C. Calculation of the mean network latency of one-step broadcast messages ($\overline{S}_b$): The mean network latency of a one-step broadcast message that crosses an arbitrary dimension, $\overline{S}_{b_i}$, can be calculated by equation 13. Averaging over the latencies of all dimensions yields the mean latency of a one-step broadcast message, $\overline{S}_b$, and is given by

$$\overline{S}_b = M + \overline{B}_b \tag{23}$$

$$\overline{B}_b = \frac{1}{n}\sum_{i=0}^{n-1} P_{V_i} \; W_{c_i} \quad (24)$$

D. Calculation of the mean waiting time at the source node ($\overline{W}_s$): The mean waiting time at the source node is calculated in a similar manner to that for a network channel (equations 15-17). By modelling the injection channel in the source node as an M/G/1 queue, the mean arrival rate and mean service time are given by the following equations

$$l_s = \frac{l_{S_u}}{n} + l_{S_b} + \frac{\overline{w}}{n} l_{S_r} \quad (25)$$

$$\overline{S}_s = \frac{l_{S_b} + l_{S_r}}{l_{S_u} + l_{S_b} + l_{S_r}} \overline{S}_b + \frac{l_{S_u}}{l_{S_u} + l_{S_b} + l_{S_r}} \overline{S}_u \quad (26)$$

where $\overline{S}_b$ can be determined using equations 23. To calculate the mean network latency of a unicast message, $\overline{S}_u$, we consider sending a message from a specific node, say node 0, to all other nodes in the network. Averaging over the network latencies of these messages yields the mean network latency of a unicast message and is given by

$$\overline{S}_u = \frac{\sum_{m=1}^{2^n-1} \Gamma^n_{0\to m}}{2^n - 1} \quad (27)$$

Approximating the variance of the service time distribution by $(\overline{S}_s - M)^2$ yields a mean waiting time at the source of

$$\overline{W}_s = \frac{l_s \overline{S}_s^2 \; (1 + \frac{(\overline{S}_s - M)^2}{\overline{S}_s^2})}{2\,(1 - l_s \overline{S}_s)} \quad (28)$$

E. Calculation of the average degree of virtual channels multiplexing ($\overline{V}$): The probability, P_{v_i}, that v adaptive virtual channels are busy in a physical channel at dimension i can be determined using a Markovian model [Dally W.J. 1992]. State p_v corresponds to v virtual channels being busy. The transition rate out of state p_k to p_{k+1} is l_c, where l_c is the traffic rate on a network channel (and is given by equation 4), while the rate out of p_k to p_{k-1} is $1/\overline{S}_i$. The transition rate out of the last state, p_V, is reduced by l_c to account for the arrival of messages while a channel is in this state. In the steady state, the model yields the following probabilities.

$$q_{v_i} = \begin{cases} 1 & v = 0 \\ q_{(v-1)_i} \, l_c \overline{S}_i & 0 < v < V \\ q_{(v-1)_i} \, \frac{l_c}{1/\overline{S}_i - l_c} & v = V \end{cases} \quad (29)$$

$$P_{v_i} = \begin{cases} \left(\sum_{j=0}^{V} q_{j_i} \right)^{-1} & v = 0 \\ P_{(v-1)_i} \, l_c \overline{S}_i & 0 < v < V \\ P_{(v-1)_i} \, \frac{l_c}{1/\overline{S}_i - l_c} & v = V \end{cases} \quad (30)$$

In virtual channel flow control, multiple virtual channels share the bandwidth of a physical channel in a time-multiplexed manner. The average degree of multiplexing of virtual channels, which takes place at the physical channel of dimension i, is given by [Dally W.J. 1992]

$$\overline{V}_i = \frac{\sum_{v=1}^{V} v^2 P_{v_i}}{\sum_{v=1}^{V} v \; P_{v_i}} \quad (31)$$

The average degree of multiplexing of a virtual channel in the network can be calculated by averaging over all dimensions and expressed as

$$\overline{V} = \frac{1}{n}\sum_{i=1}^{n} \overline{V}_i \quad (32)$$

3.2. The Unicast Latency

The mean latency of unicast messages can easily be computed since all the required information concerning these messages has already been determined. As in the case of one-step broadcast messages, the mean latency of unicast messages is composed of the mean network latency and the mean waiting time at the source before entering the network. After including the effects of virtual channel multiplexing, the mean latency of unicast messages, $\overline{L}_u$, can be calculated by

$$\overline{L}_u = (\overline{S}_u + \overline{W}_s)\overline{V} \quad (33)$$

where $\overline{S}_u$, $\overline{W}_s$ and $\overline{V}$ can be determined using equations 27, 28 and 32, respectively.

4. SIMULATION EXPERIMENTS

A set of simulation experiments was conducted to evaluate the model. The purpose of these experiments was to assess the accuracy of the analytical model results, described in section 3, for different networks.

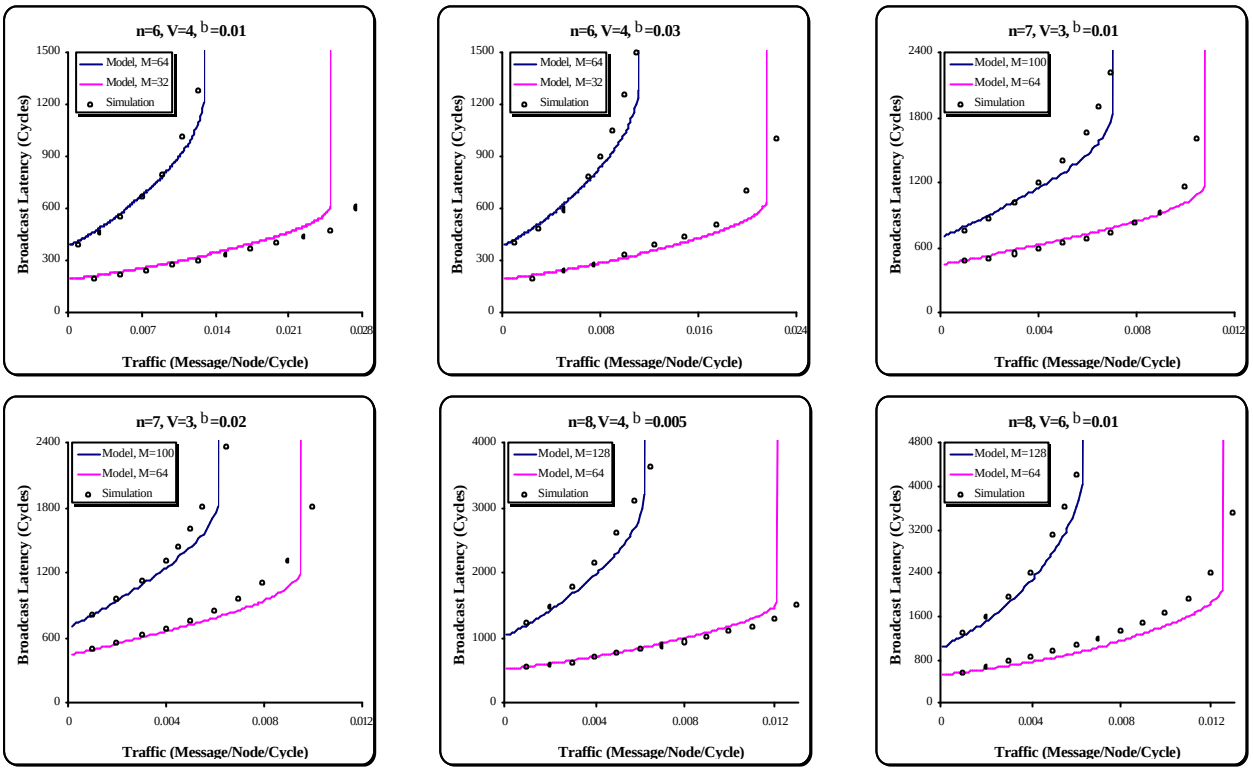

Figure 3: Validation of the broadcast message latency predicted by the model against simulation in the 6, 7 and 8-dimensional hypercubes. Message length M=32, 64, 100 and 128, broadcast portion b = 0.005, 0.01, 0.02 and 0.03 and number of virtual channels V=3, 4 and 6.

The experiment was conducted using a discrete-event simulator that performs a time-step simulation of network operations at the flit level. Each simulation experiment is run until the network reaches its steady state; that is until a further increase in simulated network cycles does not change the collected statistics appreciably. Statistics gathering was inhibited for the first 20000 messages to avoid distortions due to the start-up conditions. Extensive validation experiments have been performed for several combinations of network sizes, message lengths, different fractions of broadcast messages and virtual channels. The start-up latency, Δ, varies from one practical machine to another, and has usually been considered as a constant value independent of network traffic. For the purpose of our present study this delay factor has been fixed at $\Delta = 1$ cycle. Obviously, such a figure has no effect on the validation process, and higher values can be easily incorporated in the model. For the sake of specific illustration, latency results are presented for the following cases only:

- Network size is N=2^6, 2^7 and 2^8 nodes.
- Number of virtual channels V = 3, 4 and 6 per physical channel.
- Message length is M= 32, 64, 100 and 128 flits.
- Broadcast portion is b =0.005, 0.01, 0.02 and 0.03.

Figures 3 and 4 depict results from the mean broadcast and unicast message latency predicted by the above analytical model plotted against those provided by the simulator as a function of traffic injection in the 6-dimensional, 7-dimensional and 8-dimensional hypercubes. The horizontal axis in each figure represents the message generation rate, l_g, of each node in one cycle, while the vertical axis shows the mean unicast and broadcast message latency, respectively. The figures reveal that the simulation results closely match those predicted by the analytical model in the steady state regions (i.e. under light and moderate traffic) and even when the network starts to approach saturation. However, the discrepancies in the results near saturation are noticeable. This is due to the approximations which have been made to simplify the development of the model, such as that in equation 18 for determining the variance of service time at a network channel; this approximation greatly simplifies the model as it allows us to avoid the complex task of computing the exact distribution of the channel service time. Nevertheless, we can conclude that the model produces accurate results in the steady state regions, and its simplicity makes it a practical evaluation tool that can be used to gain insight into the behaviour of wormhole-routed

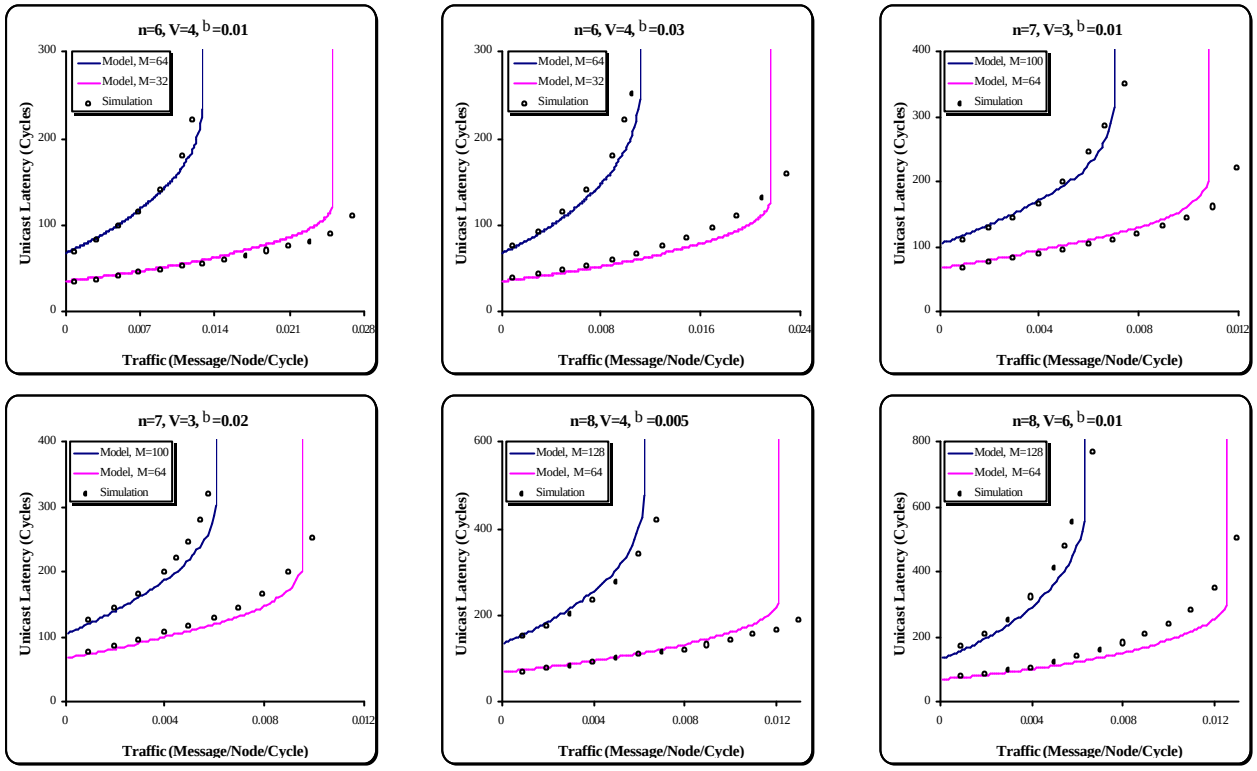

Figure 4: Validation of the unicast message latency predicted by the model against simulation in the similar situations of figure 3.

hypercubes in the presence of broadcast communication.

5. CONCLUSION

Although many broadcast algorithms have been proposed for wormhole-routed networks over the past decade, there has been little development of analytical models of these algorithms. This paper has presented a new analytical model capable of computing unicast and broadcast latency in wormhole-routed hypercubes under a number of reasonable assumptions, widely adopted in the literature. Extensive simulation experiments have shown the analytical model predicts latency with a good degree of accuracy. An obvious continuation of this work would extend the present model to other common multicomputer networks such as *k*-ary *n*-cubes and *n*-dimensional meshes. Another line of progression would be to develop new analytical models for the recently proposed multi-destination-based broadcast algorithms, such as those based on the Base Routing Conformed Path (BRCP) methodology [Panda D., Singal S. and Kesavan R. 1999].

REFERENCE

Abraham S. and Padmanabhan K. 1989, "Performance of the Direct Binary n-cube Network for Multiprocessors". IEEE Transactions on Computers, vol. 38, no. 7, Pp. 1001-1011.

Boura Y., Das C.R and Jacob T.M. 1994, "A Performance Model for Adaptive Routing in Hypercubes". In Proc. Int. Workshop on Parallel processing, Pp. 11-16.

Dally W.J. 1990, "Performance Analysis of k-ary n-cubes Interconnection Networks". IEEE Transactions on Computers, vol. 39, no. 6, Pp. 775-785.

Dally W.J. 1992, "Virtual Channel Flow Control". IEEE Transactions on Parallel & Distributed Systems, vol. 3, no. 2, Pp. 194-205.

Dongarra J. et al. 1993, "Document for a Standard Message-Passing Interface". Message Passing Interface Forum.

Draper J.T. and Ghosh J. 1994. "A Comprehensive Analytical Model for Wormhole Routing in Multicomputer Systems". Journal of Parallel & Distributed Computing, vol. 32, Pp. 202-214.

Duato J., Yalamanchili S. and Ni L. 1997, "Interconnection Networks: An Engineering Approach". IEEE Computer Society Press.

Johansson S.L. and Ho C.-T. 1989, "Optimum Broadcasting and Personalised Communication in Hypercubes". IEEE Transactions on Computers, vol. 38, no. 9, Pp. 1249-1268.

Kleinrock L. 1975, "Queueing Systems", vol. 1, John Wiley, New York.

Laudon J. and Lenoski D. 1997, "The SGI Origin: A ccNUMA Highly Scalable Server". Proc. ACM/IEEE 24th Int. Symp. Computer Architecture, Pp. 241-251.

Lin X., McKinley P. and Ni L. 1994. "Deadlock-free Multicast Wormhole Routing in 2-D Mesh Multicomputers", IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 8, Pp. 793-804.

Malumbres M.P., Duato J and Torrellas J. 1996. "An Efficient Implementation of Tree-based Multicast Routing for Distributed Shared Memory Multiprocessor", Proc. 8th IEEE Int. Symp. Parallel & Distributed Processing.

McKinley P. and Trefftz C. 1993. "Efficient Broadcast in All-port Wormhole-routed Hypercubes", Proc. Int'l Conf. on Parallel Processing, Pp. 288-291.

McKinley P., Xu H., Esfahanian A.H. and Ni L. 1994. "Unicast-based Multicast Communication in Wormhole-routed Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 12, Pp. 1252-1265.

McKinley P., Tsai Y. and Robinson D.F. 1995. "Collective Communication in Wormhole-routed Massively Parallel Computers", IEEE Computer, vol. 28, no. 12, Pp. 39-50.

Nugent S.F. 1988. "The iPSC/2 Direct-connect Communication Technology", Proc. Conf. on Hypercube Concurrent Computers & Applications, vol. 1, Pp. 51-60.

Ould-Khaoua M. 1999. "A Performance Model for Duato's Fully-adaptive Routing Algorithm in k-ary n-cubes", IEEE Transactions on Computers, vol. 42, no. 12, Pp. 1-8.

Panda D., Singal S. and Kesavan R. 1999. "Multidestination Message Passing in Wormhole k-ary n-cube Networks with Base Routing Conformed Paths", IEEE Transactions on Parallel & Distributed Systems, vol. 10, no. 1, Pp. 76-96.

Protic J., Tomasevic M. and Milutinovic V. 1997. "Distributed Shared Memory: Concepts and Systems", IEEE Computer Society, 1997.

Shahrabi A., Ould-Khaoua M. and Mackenzie L. 2000. "A Performance Model of Broadcast Communication in Wormhole-routed Hypercubes", Proc. MASCOTS'2000, IEEE Computer Society, Pp. 99-106.

Sullivan H. and Bashkow T. R. 1997. "A Large Scale Homogeneous, Fully Distributed Parallel Machine", Proc. Symp. Computer Architecture, Pp. 105-124.

APPENDIX - NOTATION

B_b	mean blocking time of a one-step broadcast message
B_{b_i}	mean blocking time of a one-step broadcast message at dimension *i*
B_0	number of leaves children of a backward tree node
B_i	number of children of a backward tree node that replicate the broadcast message *i* times
$\overline{d}$	mean distance of a unicast message
$\overline{L}_b$	mean latency of a one-step broadcast message
$\overline{L}_u$	mean latency of a unicast message
M	message length in flits
N	number of nodes in the network
N_r^i	number of nodes of the double tree that replicate the broadcast message i times
p_i	probability that a unicast message crosses *i* dimension to reach its destination
P_{r_i}	probability that a broadcast message will be replicated *i* times at a node
P_{i_v}	probability that *v* virtual channels at dimension *i* are busy
$\overline{S}_i$	mean service time of a channel at dimension *i*
$\overline{S}_s$	mean waiting time at the source
$\overline{S}_b$	mean network latency of a one-step broadcast message
$\overline{S}_{b_i}$	mean network latency of a one-step broadcast message at dimension *i*
$\overline{S}_u$	mean network latency of a unicast message
$\overline{S}_{u_i}$	mean network latency of a unicast message at dimension *i*
V	number of virtual channel per physical channel
$\overline{V}$	average degree of multiplexing of virtual channels at a given physical channel
$\overline{V}_i$	average degree of multiplexing of virtual channels at dimension *i*
$\overline{W}_{c_i}$	mean waiting time to acquire a virtual channel at dimension *i*
$G_{a\to d}^i$	required time for a message, which is originated from node a and destined for node d, to cross the network from

	dimension i to its destination
$Y_{a\to d}^{i}$	occupation time of a channel at dimension i by a message originating from node a and destined for node d
b	probability of a message to be a broadcast message
Δ	start-up latency
l_g	message generation rate
l_{s_b}	generation rate of broadcast messages
l_{s_u}	generation rate of unicast messages
l_{s_r}	rate of replicated messages
l_c	traffic rate on a network channel
l_{c_u}	traffic rate of unicast message on a channel
l_{c_b}	traffic rate of broadcast message on a channel
l_{c_r}	traffic rate of replicated message on a channel
l_s	mean arrival rate at the local queue in the source node
$\overline{w}$	mean number of replicated messages

Alireza Shahrabi received his B.Sc. and M.Sc. degree in Computer Engineering from Sharif University of Technology, Tehran, Iran, in 1991 and 1994, respectively. From 1996 till 1998, he was with the Sahand University of Technology, Tabriz, Iran. He is currently a lecturer in the School of Computing and Mathematical Sciences at Glasgow Caledonian University, UK. His current research interests are parallel computing, interconnection networks and performance modelling/ evaluation of parallel and distributed systems.

Mohamed Ould-Khaoua received his B.Sc. degree from the University of Algiers, Algeria, in 1986, and the M.App.Sci. and Ph.D. degrees in Computer Science from the University of Glasgow, U.K., in 1990 and 1994, respectively. He is currently a Lecturer in the Department of Computing Science at the University of Glasgow, U.K. He is an Associate Editor for the International Journal of Computers & Applications and International Journal of High Performance Computing & Networking. He is the Guest Editor of two special issues on systems performance evaluation in the Journal of Computation & Concurrency: Practice & Experience and IEE-Proceedings-Computers & Digital Techniques. He is the co-chair of the international workshop series on performance modeling, evaluation, and optimization of parallel and distributed systems (PMEO-PDS'02, PMEOPDS' 03, and PMEO-PDS'04). He has served on program committees of a number of international conferences. Dr. Ould-Khaoua's current research interests are performance modelling of parallel and distributed systems, and wired/wireless networks.