PERFORMANCE MODELLING OF WORMHOLE-ROUTED HYPERCUBES WITH BURSTY TRAFFIC AND FINITE BUFFERS *

DEMETRES KOUVATSOS[1], SALAM ASSI[1] AND MOHAMED OULD-KHAOUA[2]

[1] *Networks and Performance Engineering Research Group*
University of Bradford, Bradford, BD7 1DP, U.K.
E-mails: d.d.kouvatsos@scm.brad.ac.uk and s.a.assi1@bradford.ac.uk

[2] *Department of Computing Science, University of Glasgow*
Glasgow, G12 8QQ, Scotland, U.K.
E-mail: mohamed@dcs.gla.ac.uk

Abstract: An open queueing network model (QNM) is proposed for wormhole-routed hypercubes with finite buffers and deterministic routing subject to a compound Poisson arrival process (CPP) with geometrically distributed batches or, equivalently, a generalised exponential (GE) interarrival time distribution. The GE/G/1/K queue and appropriate GE-type flow formulae are adopted, as cost-effective building blocks, in a queue-by-queue decomposition of the entire network. Consequently, analytic expressions for the channel holding time, buffering delay, contention blocking and mean message latency are determined. The validity of the analytic approximations is demonstrated against results obtained through simulation experiments. Moreover, it is shown that the wormhole-routed hypercubes suffer progressive performance degradation with increasing traffic variability (burstiness).

Keywords: Wormhole routing, hypercubes, compound Poisson process (CPP), generalized exponential (GE), message latency, GE/G/1/K queue, simulation.

1 INTRODUCTION

The hypercube has been one of the most commonly employed distributed multicomputer networks due to its desirable properties, such as regularity, symmetry, low diameter and high connectivity. The iPSC/2 [Arlanskas, 1988], Cosmic Cube [Seitz, 1985] and SGI 2000 [Laudon and Lenoski, 1997] are large-scale commercial systems which are based on wormhole-routed hypercubes.

Wormhole switching has become the most widely used switching technique for modern multicomputers and distributed shared-memory multiprocessors, whose routers significantly reduce message latency (i.e., the mean amount of time from the generation of a message until it reaches the destination node) [Ni and McKinley, 1993; Dally and Seitz, 1987]. In this context, a message (or a worm) is divided into elementary units called flits, each of which is 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. However, a channel designated to transmit the header flit of a message, it should also transmit all its remaining data flits before dealing with flits of another message. A message requesting transmission is blocked if the outgoing channel is busy and resides in the network until the outgoing channel becomes available. Consequently, a blocked message may occupy only a fraction of several channels along its path and thus, deadlocks are possible unless a deadlock-free routing strategy is employed.

Most existing multicomputers use deterministic routing for deadlock avoidance [Dally and Seitz, 1987; Duato, 1997] where messages follow the same path when crossed the network from the source to the destination node. A typical example of deterministic routing is the dimension-ordered routing in the hypercube, also called e-cube routing, where messages visit dimensions in a pre-defined order.

Over the recent years, a great deal of effort has been devoted towards the development of analytic models for wormhole-routed hypercubes and tori

*This work is partly supported by the EU Network of Excellence (NoE) Euro-NGi

networks [Draper and Ghosh, 1994; Kim and Das, 1994; Sarbazi-Azad et al, 2002; Sarbazi-Azad et al, 2003]. However, most of these models employed routers with negligible flit buffers. As a consequence, these models do not adequately capture the performance behaviour of actual routers, which are often equipped with extended buffers holding more than one message flit.

Hu and Kleinrock [Hu and Kleinrock, 1997] suggested an analytic model for the study of wormhole-routed networks with finite buffers. Their approximation is subject to a Poisson message arrival process or equivalently, exponential interarrival times and arbitrary message size distributions. The justification of the Poisson assumption was based on simulation experiments that the interarrival time at each outgoing channel is close to an exponential distribution [Hu and Kleinrock, 1997] and the argument that the Poisson arrival process has a characteristic burst length that tends to be smoothed by averaging over a long enough time scale. However, even if the arrival process at the source node is assumed to be Poisson, nevertheless, the interarrival times at the outgoing channels are no longer exponential. Hence, the first two moments of the interdeparture time parameters at each channel should at least be taken into consideration. Moreover, a number of recent research studies [Crovella and Bestavros, 1997; Dinda et al, 2001; Sahuquillo et al, 2000; Min et al, 2003] have demonstrated that traffic exhibit burstiness and correlation over a wide range of time scales in a variety of networks including LANs and WANs, digitised multimedia systems, web servers, and parallel computation systems.

Using simulation system performance evaluation under bursty traffic loads is, generally, costly and time consuming. As a consequence, analytic models provide credible and cost effective alternatives to simulation models for the investigation of wormhole-routed network performance.

In this paper, an open queueing network model (QNM) is proposed for wormhole-routed hypercubes with finite buffers using deterministic routing subject to a compound Poisson arrival process (CPP) with geometrically distributed batches or, equivalently, a generalised exponential (GE) interarrival time distribution. The GE/G/1/K queue and appropriate GE-type flow formulae [Kouvatsos, 1994] are adopted, as cost-effective building blocks, in a queue-by-queue decomposition of the entire network. Consequently, analytic expressions for the channel holding time, buffering delay, contention blocking and mean message latency are determined.

The rest of the paper is organised as follows. The hypercube node structure is described in Section 2. Model assumptions and notation are presented in Section 3. The channel holding time and the finite buffer model are described in Sections 4 and 5, respectively. An analytic algorithm is summarised in Section 6 and numerical results are shown in Section 7. Conclusions are drawn in Section 8.

2 HYPERCUBE NODE STRUCTURE

An $n-$dimensional hypercube consists of $N = 2^n$ nodes, each identified by $n-$bit binary number from 0 to $2^n - 1$. Each node has exactly n neighbours. Two nodes $u = u_0 u_1 .... u_{i-1} u_i u_{i+1} ... u_{n-1}$ and $v = v_0 v_1 .... v_{i-1} v_i v_{i+1} ... v_{n-1}$, $u_i, v_i \in [0,1]$, are connected if and only if there is an i such that $u_i \neq v_i$ and $u_j = v_j$ for all $j \neq i$.

Each node consists of a processing element (PE) and router, as illustrated in Fig.1. The PE contains a processor and some local memory. The router consists of a crossbar switch with $(n+1)$ input and $(n+1)$ output physical channels. Each node is connected to its neighbouring nodes through n inputs and n output physical channels. The injection channel is used by the PE to send messages to the hypercube (via the router) and messages at the destination exit of the hypercube via the ejection channel. Each physical channel is associated with some, say V, virtual channels. Each virtual channel has its own flit queue, but shares the bandwidth of the physical channel with other virtual channels in a time-multiplexed fashion [Dally, 1992]. The router contains flit buffers for each incoming virtual channel. In an $n-$dimensional hypercube, an $(n+1)V-$way crossbar switch directs message flits from an input virtual channel to any output virtual channel. Such a switch can simultaneously connect multiple input to multiple output virtual channels.

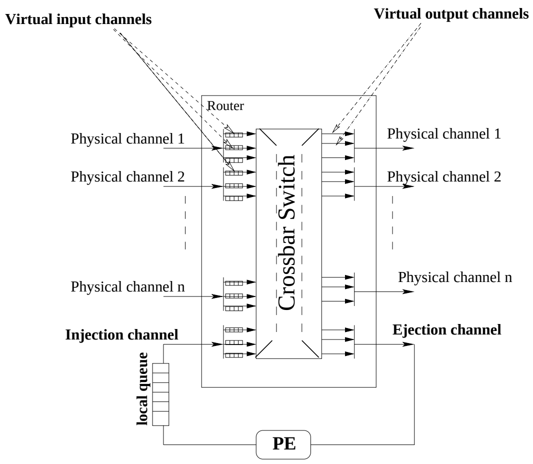


Figure 1: The hypercube node structure.

3 MODEL ASSUMPTIONS AND NOTATION

The performance modelling and analysis of the hypercube is based on the following assumptions:

- Messages are routed according to deterministic routing (i.e., messages always use the same path from source to destination by crossing the hypercube dimensions in a specific order).
- Nodes generate traffic independently of each other and which follows a GE-type interarrival time distribution with mean rate of λ_g messages/cycle and square coefficient of variation (SCV) C^2_{ag}.
- Message destinations are uniformly distributed across the network nodes. The message length is m flits, where m is a random variable whose first and second moment are known. Each flit requires one-cycle transmission time across a physical channel.
- Each physical channel is associated with only one virtual channel.
- The local queue in the source node has infinite capacity. Moreover, messages at the destination node are transferred to the local PE as soon as they arrive to their destinations.

Note that the GE-type distribution is of the form [Kouvatsos, 1994]

$$F(t) = P(X \leq t) = 1 - \tau e^{-\tau \upsilon t}, t \geq 0 \qquad (1)$$

where $\tau = 2/(1 + C^2)$, X is an interevent time random variable and $(1/\upsilon, C^2)$ are the mean and square coefficient of variation (SCV) of the interevent times, respectively.

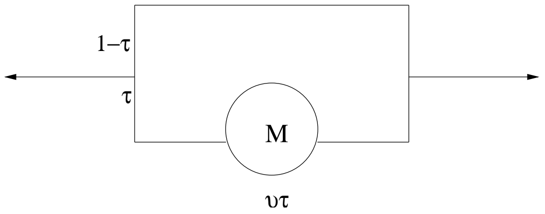


Figure 2: The GE distribution.

The GE distribution has as counting process a compound Poisson process with geometrically distributed batches with mean $1/\upsilon$. As a consequence, the GE distribution is versatile, possessing pseudo-memoryless properties which makes the solution of many GE-type queueing systems and networks analytically tractable.

The choice of the GE distribution is further motivated by the fact that measurements of actual interarrival or service times may be generally limited and so only few parameters can be computed reliably. Typically, if only the mean and variance may be relied upon, then a choice of a distribution which implies least bias (i.e., introduction of arbitrary and therefore, false assumptions) is that of GE-type distribution. Moreover, under renewality assumptions, the GE distribution is most appropriate to model simultaneous message arrivals at an input channel generated by different bursty sources with known first two moments. In this context, the burstiness of the arrival process is characterised by the SCV of the interarrival time or, equivalently, the mean size of the incoming bulk.

The GE distribution traffic may also be employed to model short range dependence (SRD) traffic with small error. For example, an SRD process may approximated by an ordinary GE distribution whose first two moments of the count distribution match the corresponding first two SRD moments. This approximation of a correlated arrival process by an uncorrelated GE traffic process may facilitate (under certain conditions) problem tractability with tolerable accuracy and, thus, the understanding of the performance behaviour of external SRD traffic in the interior of the network. It can be further argued that, for a given buffer size, the shape of the autocorrelation curve, from a certain point on wards, does not influence system behaviour. Thus, in the context of system performance evaluation, an SRD model may be used to approximate accurately long range dependence (LRD) real traffic.

Moreover, the following notation are adopted:

Δ	The buffer size in flits.
j	The number of hops to cross from the source to the destination ($1 \leq j \leq n$)
λ_g	The mean message interarrival rate at the PE.
λ_i	The message overall interarrival rate of the physical channel (or channel) i ($1 \leq i \leq j$).
C^2_{ag}	The message interarrival square of coefficient of variation (SCV) at the PE.
C^2_{ai}	The message overall interarrival SCV of channel i.
C^2_{si}	The SCV of the effective service time of channel i.
m	A random variable denoting message length in flits.
$M^*(s)$	The Laplace-Stieltjes transform (LST) of the probability density function of the message length, m.

$H_i^*(s)$ The LST of the probability density function of the contention blocking, h_i.

b_i The channel holding time (i.e., the interval from when a message first seizes channel i $(1 \leq i \leq j)$ until that message release it.

$B_i^*(s)$ The LST of the probability density function of the channel holding time, b_i.

q_i The buffer delay, which is the delay of a message head to reach the head of the input buffer of channel i, after the message has entered the buffer.

$Q_i^*(s)$ The LST of the probability density function of the buffer delay, q_i.

w_i The one-hop forwarding delay (i.e., the delay of a message head to advance to the next hop).

$W_i^*(s)$ The LST of the probability density function of the one-hop forwarding delay, w_i.

W_s The mean source waiting time (i.e., the waiting time seen by a message at the source).

T The mean message latency (i.e., the mean amount of time from the generation of a message until the last data flit reaches the local PE at the destination node).

x_i The forwarding delay for a message head to reach the position where a large enough buffer space has been accumulated to hold the entire message.

$S_i^*(s)$ The LST of the probability density function of the effective service time, s_i.

$\bar{S}$ The mean network latency (i.e., the mean time to cross the network).

P_j The probability that a given message traverses j hops from the source to the destination.

π_i The blocking probability of an arrived message is blocked at channel i.

4 Channel Holding Time

A finite buffer complicates the analytic model in two ways. Firstly, the commonly used assumption that a message reaches its destination before its tail leaves its source node, is no longer valid. It is now the case that a blocked message may occupy only a fraction of channels along its path. Secondly a buffer may hold more than one, but not an infinite number, of messages. These buffers alleviate blocking problem and their effects must be captured in the model.

A wormhole routing network differs from a virtual cut-through network [Kermani and Kleinrock, 1979] because of its channel blocking feature associated with each channel. Blocking occurs due the finite capacity of the input buffers and this results in increased the ith channel holding time, b_i $(1 \leq i \leq j)$ (which is defined as the time interval from when a message first seizes a channel until that message release it). To estimate the blocking delay at each node, it is important to first analyze how the finite buffer affects channel status and message transmission.

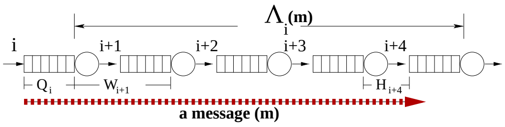


Figure 3: An illustration of the number of effective channels and various message delays.

Introducing a finite buffer on each input channel reduces the number of the so-called effective channels that a message can spread over. In other words, given a message length of m flits, b_i is only affected by a limited number of downstream channels, $\Lambda_i(m)$ and the blocking that occurs after the next $\Lambda_i(m)$ channels does not affect b_i [Hu and Kleinrock, 1997] (c.f., Fig. 3). To this end, the number of effective downstream channels for a message with m flits, at the ith channel is clearly expressed by

$$\Lambda_i(m) = \begin{cases} \lfloor \frac{m}{\Delta} \rfloor & \frac{m}{\Delta} < (j-i) \\ (j-i) & \textit{otherwise} \end{cases} \tag{2}$$

where Δ is the buffer size (in flits).

Define random variables x_i and y_i as [Hu and Kleinrock, 1997]

$$\begin{aligned} x_i &= q_i + h_{i+\Lambda_i(m)} + \textstyle\sum_{j=i+1}^{i+\Lambda_i(m)-1} w_j \\ y_i &= m + x_i \end{aligned} \tag{3}$$

The random variable, x_i, represents the forwarding delay for the message head to reach a position where enough buffer space has been accumulated to hold the entire message.

The LST of x_i is given by

$$X_i^*(s) = \begin{array}{l} \psi_i(0) + \\ \sum_{d=1}^{j-i} [\psi_i(d) Q_i^*(s) H_{i+d}^*(s) \prod_{k=i+1}^{i+d-1} W_k^*(s)] \end{array} \tag{4}$$

where $\psi_i(d)$ is the probability of the number of effective downstream channels, that $\Lambda_i(m) = d$, for a message at the ith channel and is given by

$$\psi_i(d) = \begin{cases} \mathbf{Prob}(m < (d+i)\Delta) - & \\ \mathbf{Prob}(m < d\Delta) & \frac{m}{\Delta} < j - i \\ 1 - \mathbf{Prob}(m < d\Delta) & \textit{otherwise} \end{cases} \tag{5}$$

An approximation for the LST of the channel holding time is proposed in [Hu and Kleinrock, 1997] and is determined by

$$B_i^*(s) = \begin{cases} (\bar{Y}_i + \bar{X}_i)^{-1}\left[(\bar{Y}_i - \bar{X}_i)Y_i^*(s) \right. \\ \qquad \left. +2\bar{X}_i M^*(s)\right] & \bar{M} > \bar{X}_i \\ (\bar{Y}_i + \bar{X}_i)^{-1}\left[(\bar{Y}_i - \bar{X}_i)X_i^*(s) \right. \\ \qquad \left. +2\bar{X}_i X_i^*(s)\right] & \bar{M} \leq \bar{X}_i \end{cases} \tag{6}$$

where $\bar{X}_i$, $\bar{Y}_i$ and $\bar{M}$ are the first moment of $X_i^*(s)$, $Y_i^*(s)$ and $M^*(s)$ respectively and these are the LST of x_i, y_i and m respectively.

The average channel holding time, b_i, is monotonically increasing as the forwarding delay, x_i, increases and must be as large as the message size, m.

The message different delays $W_i^*(s)$, $H_i^*(s)$ and $Q_i^*(s)$ are discussed in the following Section.

5 The GE/G/1/K Queue

In a finite buffer, flits of a message flow constantly when it is not full. However, the flow of the buffer may be interrupted due to message blocking. For the sake of accuracy and simplicity, Hu and Kleinrock [Hu and Kleinrock, 1997] treated, in the context of arbitrary queueing networks, both channel contention and the input buffer as one single queue and they defined the one-hop forwarding delay, w_i (i.e., the delay for a message to seize its output channel and reach the buffer head in the next hop) to be equivalent to the waiting time of the M/G/1/K queue. A corresponding GE-type pictorial interpretation can be seen in Fig. 4.

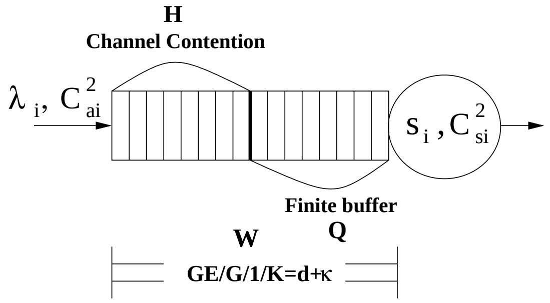


Figure 4: The forwarding delay is considered as a single queue with finite capacity.

Given κ input ports, the queue, approximately, has the capacity $\kappa + \upsilon$, where υ is the number of messages (or worms) that can be completely held in the portion of the finite buffer. However, the buffer size needs to be determined in terms of the number of flits, not the number of messages. Moreover, the messages with different sizes, is a random variable and υ is not deterministic. To simplify the analysis, an equivalent queues are used to specify how many messages can be held in the buffer and the buffer is approximated as a queue with capacity $K = d + \kappa$, (d specifies how many messages can be held in the buffer of the equivalent queue) with the probability, $\upsilon(d)$, such that,

$$\upsilon(d) = Prob(m_1 + \ldots + m_d \leq \Delta \leq m_1 + \ldots + m_{d+1}) \tag{7}$$

The LST of the one-hop forwarding delay can be estimated [Hu and Kleinrock, 1997] by

$$W_i^*(s) = \sum_{d=0}^{\infty} \upsilon(d) \Upsilon_i^*(d+\kappa, s) \tag{8}$$

where $\Upsilon_i^*(j+\kappa, s)$ is the LST of the probability density function of the waiting time of the equivalent queue with capacity K and the solution is given in [Tagagi, 1993] by

$$\Upsilon_i^*(d+\kappa, s) = \frac{\frac{p_0 \sum_{k=0}^{K-1} g_k^- [S_i^*(s)]^k}{1-\pi_i} + \sum_{k=1}^{K-1} \Pi_k^*(s) \sum_{d=0}^{K-k-1} g_d^- [S_i^*(s)]^{k+d-1}}{1-\pi_i} \tag{9}$$

where $S_i^*(s)$ is the LST of the effective service time of the GE/G/1/K queue, g_d^- is the probability that $d(d = 1, 2, ...K)$ messages are included in an arriving bulk with mean bulk size, g, and is given by

$$g_d^- = \frac{1}{g} \sum_{k=d+1}^{\infty} g_k \tag{10}$$

and

$$\sum_{k=1}^{K} \Pi_k^*(s) = \frac{1 - S_i^*(s)}{s} \sum_{k=1}^{K} \Pi_k(0) \tag{11}$$

where $\Pi_k(0) = \lambda_i g p_{k-1}$.

The blocking probability π_i and the queue length distribution, p_k, of having k messages in the GE/G/1/K queue, derived via the principle of Maximum entropy (ME) are available in [Kouvatsos, 1994].

The following subsections present a GE-type traffic analysis, the effective service time, the buffering delay, the contention blocking and the mean source waiting time.

5.1 Traffic Analysis

For a uniform traffic pattern, messages are destined to any of the $2^n - 1$ nodes in the $n-$dimensional hypercube with equal probability. A PE generates, on average, λ_g messages in a cycle, these can be transmitted to any destinations that are j hops long with probability p_j given by Eq. 13. Two types

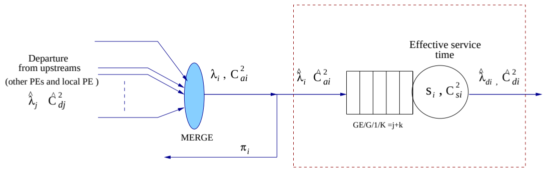

Departure
from upstreams
(other PEs and local PE)
$\hat\lambda_j$ $\hat C^2_{dj}$
MERGE
λ_i , C^2_{ai}
π_i
$\hat\lambda_i$ $\hat C^2_{ai}$
Effective service
time
s_i , C^2_{si}
GE/G/1/K =j+k
$\hat\lambda_{di}$, $\hat C^2_{di}$

Figure 5: Flow streams at channel i.

of message arrive at a given node using the input channels [Kim and Das, 1994]; one from the local PEs, λ_0 and the other is the transit message, λ_n, from other PEs, λ_n is the summation of the transit message rates of all channels, only messages with path length greater than 2 require an intermediate node and the number of intermediate channels affected by a $j-$hop message is $(j-1)$.

The total arrival rate to a node is the sum of both rates , $\hat\lambda_i = \lambda_0 + \lambda_n$, and these are distributed among the n output channels. Thus, the effective traffic rate at each input channel, $\hat\lambda_i$, offered to output channel [Kim and Das, 1994] is

$$\hat\lambda_i = \sum_{j=1}^{n} P_j \frac{\lambda_g}{n} + \sum_{j=2}^{n} (j-1) P_j \frac{\lambda_g}{n}, \qquad (12)$$

where P_j, is the probability that a given message traverses j hops and is given by

$$P_j = \frac{\binom{n}{j}}{2^n - 1}, \quad (1 \le j \le n) \qquad (13)$$

Assuming that all flow processes (i.e., merge split and departure) are renewal and the corresponding interevent-time distributions of GE-type, each queue i, can be seen as a GE/G/1/K queue with GE overall interarrival process, formed by merging of departing upstreams queues of queue i (see Fig. 5). The parameter of the effective interarrival process of each queue i can be approximated by focusing on a stable FCFS GE/G/1 queue with infinite capacity and applying the GE-type flow formulae [Kouvatsos, 1994], given by

$$\hat C^2_{ai} = -1 + \left\{ \begin{array}{l} \sum_{j=1}^{n} \frac{P_j \lambda_g (\hat C^2_{aj}+1)^{-1}}{n\hat\lambda_i} \\ + \sum_{j=2}^{n} \frac{(j-1) P_j \lambda_g (\hat C^2_{aj}+1)^{-1}}{n\hat\lambda_i} \end{array} \right\}^{-1} \qquad (14)$$

where $\hat C^2_{aj} = 1 - \frac{P_j}{n} + \frac{P_j \hat C^2_{dj}}{n}$ and the parameter $\hat C^2_{dj}$ is the SCV of the effective interdeparture time given by $\hat C^2_{dj} = \hat\rho_j(1-\hat\rho_j) + (1-\hat\rho_j)\hat C^2_{aj} + \hat\rho_j^2 C^2_{sj}$, where $\hat\rho_j = \hat\lambda_j s_j$. s_j and C^2_{sj} are the mean and SCV of the effective service time respectively. $C^2_{sj} = \frac{2}{\tau_j} - 1$ where $\tau_j = \frac{2(s_j)^2}{s_j^2}$, s_j^2 is the second moment of the effective service time.
The corresponding parameters of the overall interarrival process are clearly given by [Kouvatsos, 1994]

$$\lambda_i = \frac{\hat\lambda_i}{1-\pi_i}, \quad C^2_{ai} = \frac{\hat C^2_{ai} - \pi_i}{1-\pi_i} \qquad (15)$$

where π_i is the blocking probability of an arrived message is blocked at channel i and is given by

$$\pi_i = \sum_{k=0}^{K} \delta(k)(1-\sigma_i)^{K-k} p_k \qquad (16)$$

where

$$\delta(k) = \left\{ \begin{array}{ll} \frac{r_i}{r_i(1-\sigma_i)+\sigma_i} & k = 0 \\ & \\ 1 & k \ne 0 \end{array} \right. \qquad (17)$$

where $\sigma_i = \frac{2}{1+C^2_{ai}}$, $r_i = \frac{2}{1+C^2_{si}}$ and p_k is the queue length distribution of having k messages in the queue [Kouvatsos, 1994].

5.2 Effective Service Time

For uniform traffic pattern, the statistical characteristic of a physical channel in a given dimension are identical to those of any other dimension. However, the blocking nature of wormhole routing leads to a differentiation among channels when deterministic routing is used. In calculating the effective service time we start with the ejection channel back to the source channel.

The LST of the effective service time, s_i, of channel i can be written as

$$S_i^*(s) = B_{i-1}^*(s) H_{i-1}^*(s), \quad (1 \le i \le j) \qquad (18)$$

Equation 18 consists of the LST of the message channel holding time, b_{i-1}, and the LST of the contention blocking, h_{i-1}, of the $(i-1)$th channel.

5.3 Buffering Delay and the Contention Blocking

The LST of the buffering delay [Hu and Kleinrock, 1997], is given by $Q_i^*(s) = \frac{W_i^*(s)}{H_i^*(s)}$. The contention blocking, $H_i^*(s)$, can be approximated by a GE/G/1/K queue with K $= \kappa$ and the queue service time is exactly the channel holding time, $B_i^*(s)$. In order to find $H_i^*(s)$, a simple approximation to $B_i^*(s)$ is proposed in [Hu and Kleinrock, 1997] and is given by the probability

$$B_i^*(s) = \begin{array}{l}\mathbf{Prob}(fullB)S_i^*(s)+\\ (1-\mathbf{Prob}(fullB))\,M^*(s)\end{array} \tag{19}$$

The approximation in Eq. 19 is based on the following:

- When the buffer is full, one flit of data out of the buffer corresponds to one flit of data in. Thus, $B_i^*(s) = S_i^*(s)$.
- When there is space in the buffer, a message flows without interruption. Thus, $B_i^*(s) = M^*(s)$.

The probability that more than v messages are in the GE/G/1/K queue is estimated from the queue length distribution, p_k, and the blocking probability, π_i and is given by

$$\mathbf{Prob}\,(fullB) = \sum_{d=0}^{\infty} v(d)\left((1-\pi_d)\sum_{k=d}^{d+\kappa-1} p_k + \pi_d\right) \tag{20}$$

5.4 Mean Source Waiting Time

A $j-$hop message (i.e. a message that needs to make j ($1 \leq j \leq n$) hop to cross from its source to destination originating from a given source node sees a network latency of s_j (given by Eq. 18 when $i = j$).

Averaging over all possible values of j, ($1 \leq j \leq n$), yields the mean network latency as

$$\bar{S} = \sum_{j=1}^{n} P_j s_j \tag{21}$$

where P_j, is the probability that a given message traverses j hops given by Eq. 13.

Modelling the local queue in the source node as an GE/G/1 queue, with the mean interarrival rate λ_g and interarrival SCV of C_{ag}^2 and a service time equal to the mean network latency of Eq. 21 of mean $\bar{S}$ and SCV C_s^2, yields the mean waiting time seen by a message at source node as [Kouvatsos, 1994]

$$W_s = \frac{\bar{S}}{2}\left(1 + \frac{C_{ag}^2 + \rho C_s^2}{1-\rho}\right) - \bar{S}, \rho = \lambda_g \bar{S} \tag{22}$$

where $C_s^2 = \frac{2}{\tau} - 1$ and $\tau = \frac{2(\bar{S})^2}{\bar{S^2}}$, $\bar{S}^2$ is the second moment of the service time at the source node.

5.5 Mean Message Latency

The model computes the mean message latency, T, which is the mean amount of time from the generation of a message until the last data flit reaches the local PE at the destination node, using the formulae:

$$T = \bar{S} + W_s \tag{23}$$

where $\bar{S}$ and W_s are the mean network latency and the mean waiting time at the source node given by Eqs. 21 and 22 respectively.

6 THE ANALYTIC ALGORITHM

The steps of the analytic algorithm, based on the analysis presented in Sections 4 and 5 is described below.

```
Begin
Inputs: $n$, $\bar{M}$, $\Delta$, $\lambda_g$, $C_{ag}^2$;
Step1:  Initialisation: $C_{di}^2$, $\pi_i$ ($i = 1, ...., j$) and $s_i$,
        $C_{si}^2$ (when $i = 0$);
Step2:  Solve the system of non-linear equation
        ($\pi_i$);
        (2.1) Compute: $\lambda_i$, $C_{ai}^2$, $\pi_i$, $s_i$, $C_{si}^2$, $i =
        1, ..., j$;
        (2.2) Find new values for $\pi_i$;
        (2.3) Return to 2.1 until convergence of
        $\pi_i$;
Step3:  Find new values for ($C_{di}^2$);
Step4:  Return to step 2 until convergence of
        ($C_{di}^2$);
Step5:  For $i = 1, ...j$, evaluate in the following order $S_i^*(s)$, $W_i^*(s)$, $H_i^*(s)$,
        $Qi^*(s)$, $B_i^*(s)$;
Step6:  Find $\bar{S}$ using Eq. 21;
Step7:  Find T using Eq. 23;
End.
```

Note that only the first two moments of each distribution is required, which can be obtained by the differentiation of the LST and setting $s = 0$.

7 NUMERICAL RESULTS

The analytical model has been validated by means of a discrete-event simulator. Each simulation experiment was run for 10 batches. In each batch, the statistics are collected for 10000 messages delivered to their destinations; the first batch is ignored to avoid the warmup effects. The simulator uses the same assumptions as the analysis. The network cycle time is defined as the transmission time of a single phit (i.e., a channel word or a group of bits that can be transmitted in one go) from one router to the next. Messages are generated at each source node according to a compound Poisson process (CPP) with geometrically distributed batches and with mean interarrival rate of λ_g messages/cycle and SCV of C^2_{ag}.

A GE distribution is employed to determine the first two moments of the iterdeparture flows at each finite queue. Message length is exponentially distributed with mean $\bar{M}$ flits (i.e., a flit is group of phits). Destinations are uniformly distributed across the hypercube, and messages are routed according to deterministic routing (e-cube routing). The mean message latency, T, is calculated and plotted against the traffic generation rate, these are shown in Fig. 6 and Fig. 7. To validate the model, numerous validation experiments have been performed for several combination of message length, buffer size and interarrival SCV. Note that the relative error between simulation and analytic latency results may be captured by the following formulae:

ERROR(T) = [SIM(T)-ANAL(T)]/SIM(T)

where SIM(T) and ANAL(T) are the simulation and analytic latency results, respectively.

Figures 6(a), 6(b), 6(c) and 6(d) depict message latency results predicted by the above models plotted against those provided by the simulator for 2^8 hypercube while the results in Figs. 7(a) and 7(b) are for the 2^7 hypercube. The figures reveal that the simulation results closely match those predicted by the analytical analysis with less than a maximum of 15 percent error (c.f., Tables 1-9). Moreover, it can be observed in Figs 6(c) and 6(d) that the hypercubes with wormhole routing experience progressively performance degradation with increasing values of the SCVs of the interarrival times of traffics generated by the injection channels.

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
3000	62.03	63.32	0.021
2000	67.63	64.27	0.050
1500	72.88	72.59	0.004
1000	82.60	83.75	0.014
900	87.11	88.29	0.014
800	93.98	94.63	0.007
700	100.54	104.02	0.035
600	116.98	120.63	0.031
500	147.71	147.47	0.002
450	182.02	170.51	0.063
420	203.5	190.29	0.065

Table 1: The 8D hypercube results with buffer= 2 flits, $\bar{M} = 8$ flits and $C^2_{ag} = 5$

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
3000	34.27	35.54	0.037
2000	34.48	35.27	0.023
1000	37.29	37.87	0.016
900	38.59	38.66	0.002
800	39.26	39.74	0.012
700	39.56	41.26	0.043
600	42.01	43.56	0.037
500	44.79	47.33	0.057
400	52.65	54.42	0.034
350	57.18	60.80	0.063
300	66.88	71.56	0.070
250	95.85	92.80	0.032
230	115.18	108.09	0.062
140	1213.32	1225.60	0.010

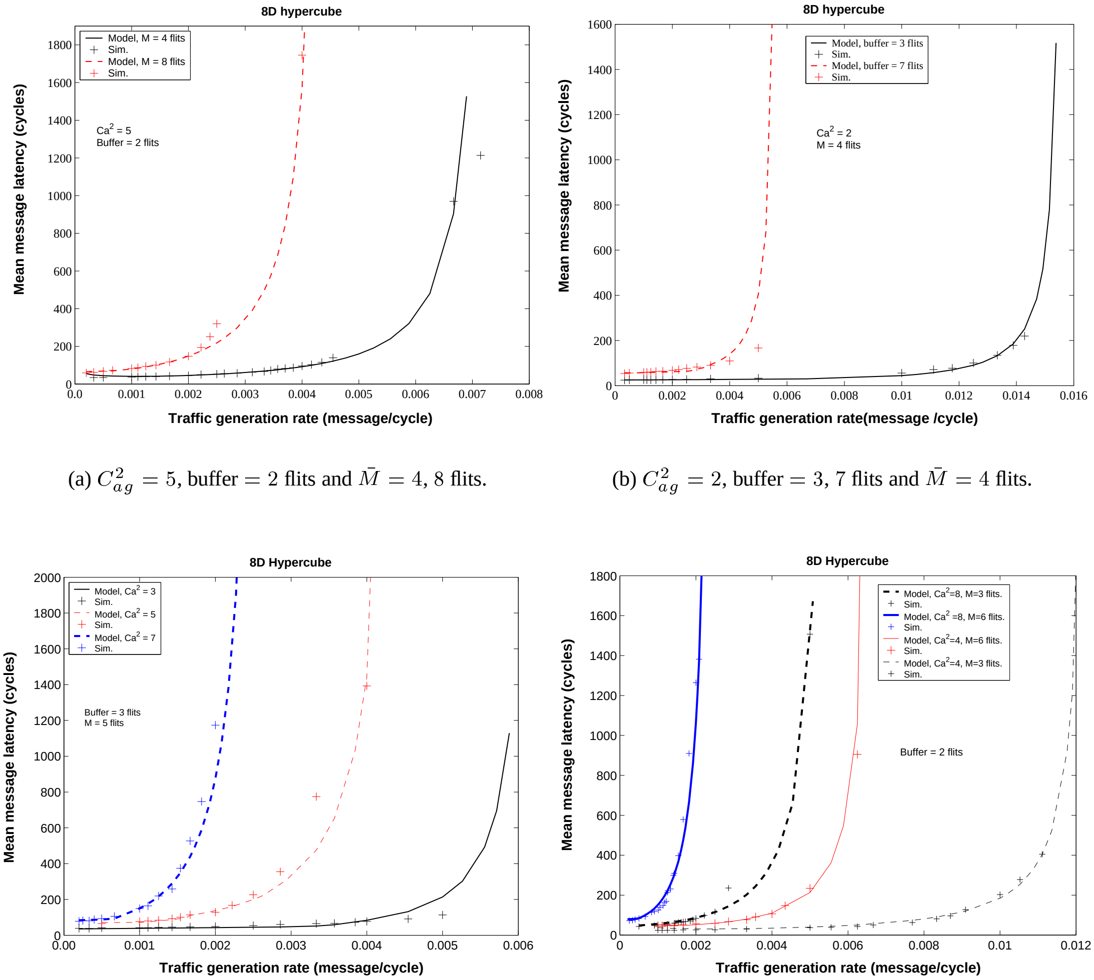
Table 2: The 8D hypercube results with buffer= 2 flits, $\bar{M} = 4$ flits and $C^2_{ag} = 5$

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
1000	23.91	22.77	0.048
900	24.01	22.87	0.047
800	24.19	23.05	0.047
700	24.20	23.32	0.036
600	25.06	23.78	0.051
500	25.56	24.54	0.040
400	26.39	25.93	0.017
300	29.90	29.71	0.006
200	36.86	39.92	0.083
100	202.32	207.71	0.027
95	277.70	268.62	0.033
90	405.29	393.66	0.029

Table 3: The 8D hypercube results with buffer= 2 flits, $\bar{M} = 3$ flits and $C^2_{ag} = 4$

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
1000	43.68	43.54	0.003
900	44.40	43.6	0.018
800	45.79	45.03	0.017
700	46.17	46.81	0.014
600	50.49	49.58	0.018
500	56.43	54.28	0.038
400	58.92	63.53	0.078
300	76.99	87.60	0.138
200	234.07	226.79	0.031
160	905.24	945.07	0.044

Table 4: The 8D hypercube results with buffer= 2 flits, $\bar{M} = 6$ flits and $C^2_{ag} = 4$



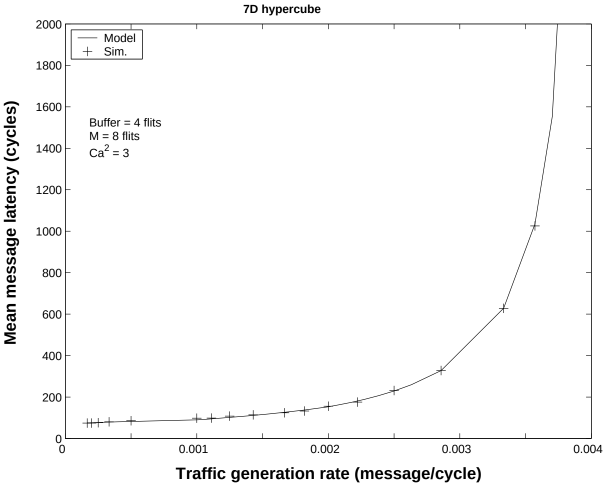
(a) $C^2_{ag} = 5$, buffer $=$ 2 flits and $\bar{M} = 4, 8$ flits.

(b) $C^2_{ag} = 2$, buffer $= 3, 7$ flits and $\bar{M} = 4$ flits.

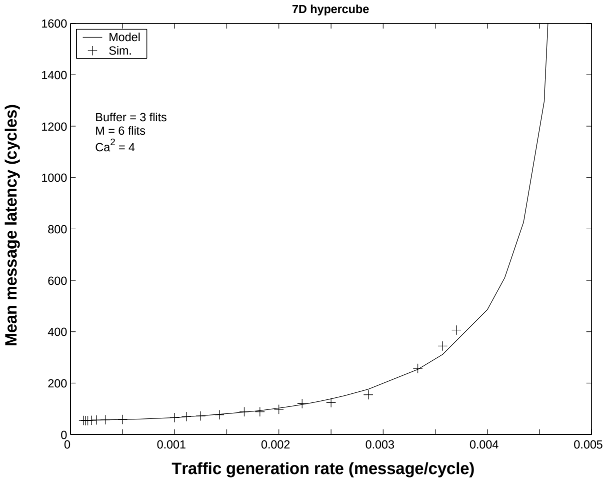
(c) $C^2_{ag} = 3, 5, 7$, buffer $=$ 3flits and $\bar{M} = 5$ flits.

(d) $C^2_{ag} = 4, 8$, buffer $=$ 2 flits and $\bar{M} = 3, 6$ flits.

Figure 6: The mean message latency predicted by the analytical model and simulation against the traffic generation rate for the 8th dimensional hypercube.



(a) $C_{ag}^2 = 3$, buffer $=4$ flits and $\bar{M} = 8$, flits.



(b) $C_{ag}^2 = 4$, buffer $=3$ flits and $\bar{M} = 6$ flits.

Figure 7: The mean message latency predicted by the analytical model and simulation against the traffic generation rate for the 7th dimensional hypercube.

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
2000	65.02	65.59	0.009
1000	76.40	77.06	0.009
900	79.67	80.44	0.010
800	83.98	85.13	0.014
700	92.36	92.02	0.004
650	100.56	103.66	0.031
600	113.61	110.47	0.028
500	128.35	131.92	0.028
450	167.16	149.91	0.103

Table 5: The 8D hypercube results with buffer= 3 flits, $\bar{M} = 5$ flits and $C^2_{ag} = 5$

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
5000	78.51	77.72	0.010
4000	81.94	78.26	0.045
3000	80.24	79.19	0.013
2000	92.87	81.11	0.127
1000	149.03	157.73	0.058
900	164.17	186.52	0.136
800	219.10	232.39	0.061
700	288.76	313.11	0.084
650	373.88	379.7	0.016
600	526.87	481.03	0.087

Table 6: The 8D hypercube results with buffer= 3 flits, $\bar{M} = 5$ flits and $C^2_{ag} = 7$

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
2000	43.17	41.85	0.031
1100	47.75	49.21	0.031
1000	51.68	51.09	0.011
900	51.72	53.60	0.036
800	54.58	57.09	0.046
700	58.83	62.2	0.057
650	63.95	65.74	0.028
600	66.70	70.29	0.054
580	67.58	72.49	0.073
520	76.88	82.18	0.069
500	84.15	85.14	0.012
450	98.62	98.23	0.004
400	116.56	117.2	0.005
200	1507.3	1431.8	0.050

Table 7: The 8D hypercube results with buffer= 2 flits, $\bar{M} = 3$ flits and $C^2_{ag} = 8$

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
6000	74.68	75.61	0.012
5000	74.70	76.39	0.023
4000	76.39	77.49	0.014
3000	80.51	79.21	0.016
2000	85.86	82.23	0.042
1000	98.04	90.24	0.080
900	98.85	95.00	0.039
800	108.33	101.92	0.059
700	114.15	111.63	0.022
600	124.09	126.47	0.019
550	132.78	137.41	0.035
500	155.96	153.24	0.017
450	175.66	181.23	0.032
400	231.51	229.26	0.010
350	328.28	327.73	0.002
300	628.00	628.46	0.001
280	1025.68	1035.00	0.009

Table 8: The 7D hypercube results with buffer= 4 flits, $\bar{M} = 8$ flits and $C^2_{ag} = 3$

$1/\lambda_g$	**SIM(T)**	**ANAL(T)**	**ERROR(T)**
8000	54.52	54.48	0.001
6000	54.09	55.01	0.017
5000	54.98	55.42	0.008
4000	56.61	56.01	0.011
3000	57.28	56.93	0.006
2000	58.66	58.55	0.002
1000	65.86	65.65	0.003
900	69.87	68.87	0.014
800	72.05	73.05	0.014
700	76.74	78.75	0.026
600	88.39	87.10	0.015
550	88.53	92.84	0.049
500	97.90	102.59	0.048
450	120.20	116.71	0.029
400	123.97	138.60	0.118
300	257.06	253.47	0.014
280	344.18	311.80	0.094
270	406.18	353.46	0.130

Table 9: The 7D hypercube results with buffer= 3 flits, $\bar{M} = 6$ flits and $C^2_{ag} = 4$

8 CONCLUSIONS

An open QNM is proposed for wormhole-routed hypercubes with finite buffers, GE-type traffic flows and deterministic routing. The GE/G/1/K queue and appropriate GE-type flow formulae are adopted, as cost-effective building blocks, in a queue-by-queue decomposition of the entire network. Consequently, analytic expressions for the channel holding time, buffering delay, contention blocking and mean message latency are determined. Simulation experiments have revealed that the approximate results from the analytic model are comparable in accuracy with those obtained through simulation. Moreover, it is shown that the wormhole routing based hypercubes networks suffer progressively performance degradation with increasing traffic variability (burstiness).

Future work will extend the above modelling approach to include more than one virtual channel and also to other major interconnection networks, such as k-ary n-cubes and meshes, and also to other routing algorithms, such as Duato's adaptive routing [Duato, 1993; Duato, 1997].

REFERENCES

Arlanskas R. 1997, " iPSC/2 system: A Second Generation Hypercube". *Proc. 3rd ACM Conf. Hypercube Concurrent Computers and Applications*, ACM Press, Pp.38–42

Crovella M.E. and Bestavros A. 1997, " Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes". *IEEE/ACM Transaction Networking*, Vol. 5(6), Pp835–846.

Dally W.J. and Seitz C.L. 1987, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks". *IEEE Transaction on Computers*, Vol. 36, Pp547–553.

Dally W.J. 1992, "Virtual-Channel Flow Control". *IEEE Transaction on Parallel and Distributed Systems*, Vol. 3(2), Pp194–205.

Dinda P.A., Garcia B. and Leung K.S. 2001, "The Measured Network Traffic of Compiler-Parallelized Programs". *In Proc. Int. Conf. Parallel Processing (ICPP'2001)*, IEEE Computer Society Press, Pp175–184.

Draper J.T. and Ghosh J. 1994, "A Comprehensive Analytical Model for Wormhole Routing in Multicomputer Systems". *Journal of Parallel and Distributed Computing*, Vol. 23, Pp202–214.

Duato J. 1993, "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks". *IEEE Transaction on Parallel and Distribution Systems*, Vol. 4(12), Pp1321–1331.

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

Hu P. and Kleinorck L. 1997, "An Analytical Model for Wormhole Routing with Finite Size Input Buffers". *In: Proceeding of the 15th International Teletraffic Congress*, University of California, Los Angeles.

Kermani P. and Kleinorck L. 1979, " Virtual Cut-Through: A New Computer Communication Switching Technique". *Computer Networks*, Vol. 3, Pp267-289.

Kim J. and Das C.R. 1994, " Hypercube Communication Delay with Wormhole Routing". *IEEE Transaction on Computers*, Vol. 43(7), Pp806–814.

Kouvatsos D.D. 1994, "Entropy Maximisation and Queueing Network Models". *Annals of Operation Research*, Vol. 48, Pp63–126.

Laudon J. and Lenoski D. 1997, "The SGI Origin: A ccNUMA Highly Scalable Server". *In Proc. ACM/IEEE 24th Int. Symposium Computer Architecture (ISCA-24)*, ACM Press. Pp241–251.

Min G., Ould-Khaoua M. and Mackenzie L.M. 2003, "On The Relative Performance Merits of Switching Techniques Under Bursty Loads". *Journal of Interconnection Networks*, Vol. 4(2), Pp179-197.

Ni L.M. and McKinley P.K. 1993, "A Survey of Wormhole Rrouting Techniques in Direct Networks". *IEEE Computer*, Vol. 26 (2), Pp62–76.

Sahuquillo J., Nachiondo T., Cano J.C., Gil J.A. and Pont A. 2000, "Self-similarity in SPLASH-2 Workloads on Shared Memory Multiprocessors Systems". *In Proc. 8th Euromicro Workshop Parallel and Distributed Processing (EURO-PDP'2000)*,IEEE Computer Society Press, Pp293–300.

Sarbazi-Azad H., Khonsari A. and Ould-Khaoua M. 2003, "Analysis of k-Ary n-Cubes with Dimension-Ordered Routing". *Future Generation Computer Systems*, Vol. 19, Pp493–502.

Sarbazi-Azad H., Khonsari A. and Ould-Khaoua M. 2002, "Performance Analysis of Deterministic Routing in Wormhole k-Ary n-Cubes with Virtual Channels". *Journal of Interconnection Networks*, Vol. 3(1,2), Pp67–83.

Seitz C. L. 1985, "The Cosmic Cube". *Communication of the ACM*, Vol. 28, Pp22–33.

Tagagi H. 1993, "Queueing Analysis: A Foundation of Performance Evaluation: Vol. 2". *North-Holland, New York, NY, U.S.A.*

BIOGRAPHIES

Demetres Kouvatsos received a BSc degree in Mathematics, Athens National University (1970), a MSc degree in Statistics, Victoria University of Manchester (1971)and a PhD degree in Computation, UMIST, University of Manchester (1974).

Professor Kouvatsos is the Head of the Performance Modelling and Engineering (PERFORM) Research Group, Department of Computing, School of Informatics. Over the years he has pioneered new and cost-effective analytic methodologies for the approximate analysis of arbitrary queueing network models (QNMs) as applied to the performance evaluation and engineering of computer telecommunication networks. His latest technical interests include traffic modelling based on batch renewal processes and advanced analytic techniques for the performance modelling, congestion control and optimisation of ad hoc wireless networks, 3G and 4G mobile switch architectures, all optical networks and the Internet.

Professor Kouvatsos was the Chairman of seven IFIP Working Conferences on the Performance Modelling and Evaluation of ATM and IP Networks (ATM and IP 93-98, 2000) and the Co-Chairman of the 3rd and 4th International Workshops on ”Queueing Networks with Finite Capacity” (1995, 2000). He also served as the Technical Track Chairman of the ”Analytical and Numerical Modelling Techniques” Conference of ESM ’98 and as a program co-chair for the IFIP Working Conference ATM and IP 01. More recently, he acted as the General Chair of HET-NETs 03, the 1st International Working Conference on the Performance Modelling and Evaluation of Heterogeneous Networks (2003) which was staged under the auspices of the EU Network of Excellence (NoE) Euro-NGi and the support of several industrial and academic organisations worldwide. Moreover, he is the co-leader of the WP.JRA.5.5 of Euro-NGi on Numerical, Simulation and Analytic Methodologies. His professional associations include memberships with the IFIP Working Group WG6.2 on Network and Inter-network Architectures and WG 6.3 on the Performance of Computer Networks. Moreover, he is a founder member of the UK Performance Engineering Specialist Group of British Computer Society (BCS). He is also an elected member of the EPSRC College and the Recipient of the IFIP Silver Core Award.

Salam Assi received a BSc degree in 1995 in Physics from Al-Najah National University in Nablus, West Bank, Palestine, and in 2000 the MSc degree in Computing with distinction from the University of Bradford, UK.

She is currently a PhD student in the area of Performance Modelling and Engineering at the University of Bradford.

Mohamed Ould-Khaoua received his BSc degree from the University of Algiers, Algeria, in 1986, and the MAppSci and PhD degrees in Computer Science from the University of Glasgow, U.K., in 1990 and 1994, respectively. He is currently a Reader in the Department of Computing Science at the University of Glasgow, U.K.

His research focuses on applying theoretical results from stochastic processes and queuing theory to the quantitative study of hardware and software architectures. Dr. Ould-Khaoua serves in the Editorial of the International Journal of Parallel, Emergent and Distributes Systems and is an Associate Editor of the International Journal of Computers and Applications and International Journal of High-performance Computing and Networking. He is the Guest Editor of eight special issues related to performance modelling and evaluation of computer systems and networks in the Journal of Computation and Concurrency: Practice and Experience, Performance Evaluation, Supercomputing, Journal of Parallel and Distributed Computing, IEE-Proceedings-Computers and Digital Techniques, International Journal of High Performance Computing and Networking, and Cluster Computing. He is the Co-Chair of the international workshop series on performance modeling, evaluation, and optimization of parallel and distributed systems (PMEO-PDS) and ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), and International Workshop on Networks for Parallel, Cluster, and Grid Systems (PEN-PCGCS). He has served on the program committees of many international conferences and workshops. Dr. Ould-Khaoua’s current research interests are performance modelling/evaluation of wired/wireless communication networks and parallel/distributed systems. He is a member of the IEEE CS.