On the Performance of Cubic Networks under Correlated Traffic Pattern

Geyong Min and Mohamed Ould-Khaoua
Department of Computing Science
University of Glasgow
Glasgow, G12 8RZ, U.K.
Email: {geyong, mohamed}@dcs.gla.ac.uk

Abstract
The efficiency of a multicomputer is critically dependent on the performance of its interconnection network. This paper presents a new analytical model for circuit-switched cubic networks in the presence of correlated traffic pattern, which is a typical scenario for multimedia applications. A message in circuit switching may need a number of connection attempts before successfully setting up a path from source to destination. The proposed model uses the approach of superposing infinite correlated traffic streams to capture the effective traffic entering the network from a source, which includes the traffic generated by the source node and the traffic due to many connection attempts. Simulation experiments reveal that the model exhibits a good degree of accuracy under various operating conditions.

1. Introduction

The cubic network (or hypercube) is one of the popular interconnection networks in existing multicomputers due to its desirable properties, such as regular structure, symmetry, low diameter and high connectivity to deal with fault-tolerance [5]. Circuit switching has been employed in practical multicomputers, e.g. Intel iPSC/2 [5]. In this switching method, only when a path has been set up from source to destination does message transmission start. To establish a path, a source node injects a header flit into the network, which reserves channels as it progresses towards its destination. When the header reaches the destination node, an acknowledgement flit is sent back to the source, signalling path establishment and the start of the message transmission phase. If the header is blocked upon reaching an intermediate node, the partial reserved path is torn down by propagating a "release" signal backward to the source.

Generally, circuit switching provides good performance for long messages. Moreover, it can provide messages with a guaranteed latency once a connection has been established, which is desirable for some multimedia applications [6]. Another advantage of circuit switching is related to the fact that adaptive routing [5], which enables messages to explore alternative paths to improve network performance, can be easily implemented with this switching method. This is because message deadlock cannot arise in circuit switching since all seized channels are released when blocking occurs.

Many network studies have revealed that the traffic generated by multimedia applications exhibits a high degree of burstiness (i.e., time-varying arrival rates) and possesses strong correlation in the number of message arrivals between adjacent time intervals [9], [20], [22]. This kind of traffic can significantly affect network performance. Several analytical models for multicomputer networks, e.g. hypercubes, with circuit switching, have been described in the literature [1], [2], [15], [21]. However, all these models have been discussed under the assumption of the Poisson input traffic. This paper proposes a new performance model for circuit-switched cubic networks in the presence of Poisson bursty traffic. The model is based on the Markov-Modulated Poisson Process (MMPP), which has been extensively used for modelling correlated traffic generated by multimedia applications because it can capture the time-varying arrival rate and the important correlation among inter-arrival times while still maintaining analytical tractability [7]–[9], [11], [19], [20], [22]. Although performance models for high-radix $k$-ary $n$-cubes under bursty traffic have been reported very recently [14], [16] the analytical model presented here is the first to use the approach of superposing infinite correlated traffic streams to capture the effective traffic entering the network from a source in circuit switching, which includes the traffic generated by the source node and the traffic due to many connection attempts. The validity of the model is demonstrated by comparing analytical results to those obtained through simulation experiments of the actual system.

The rest of the paper is organised as follows. Section 2 describes the node structure used in the analysis. Section 3 presents the derivation of the proposed analytical model. Section 4 validates the model through simulation experiments. Finally, Section 5 concludes this study.
2. Node structure in the cubic network

An n-dimensional cubic network consists of \( N = 2^n \) nodes, each identified by an n-bit binary number from 0 to \( 2^n - 1 \) [5]. Two nodes \( u = u_0u_1...u_{n-1}u_{i+1}...u_{n-1} \) and \( v = v_0v_1...v_{i-1}v_i...v_{n-1} \), \( u_i, v_i \in \{0,1\} \), are connected if and only if there is an \( i \) such that \( u_i \neq v_i \) and \( u_j = v_j \) for all \( j \neq i \). Therefore, each node has exactly \( n \) neighbours.

Each node consists of a processing element (PE) and a router, as depicted in Figure 1. The PE contains a processor and some local memory. The router has \((n+1)\) input and \((n+1)\) output channels. Each node is connected to its \( n \) neighboring nodes through \( n \) input and \( n \) output channels. The remaining channels are used by the local PE to inject/eject messages to/from the network respectively. Messages generated by the PE are transferred to the router through the injection channel. Messages at the destination are transferred to the local PE through the ejection channel. The router contains flit buffers for input virtual channels; a virtual channel shares the bandwidth of the physical channel with other virtual channels in a time-multiplexed fashion [3]. The input and output channels are connected by a \((n+1)\)\( V \)-way crossbar switch, where \( V \) is the number of virtual channels per physical channel. Such a switch can simultaneously connect multiple input to multiple output channels in the absence of channel contention.

![Node structure in the cubic network](image)

**Figure 1: Node structure in the cubic network.**

3. The analytical model

Before presenting the derivation of the analytical model, let us first describe the correlated traffic, and then outline the assumptions used in the analysis.

3.1. Modelling the correlated traffic

The MMPP\((m)\) [7] is a doubly stochastic process with an arrival rate governed by an \( m \)-state irreducible continuous time Markov chain. The main feature of the MMPP is its ability to capture the time-varying arrival rate and correlation between inter-arrival times. Moreover, it is closed under the superposition and splitting operations; the superposition and splitting of MMPPs give rise to a new MMPP [7], [17]. Such features have made the MMPP very attractive for modelling traffic generated by multimedia applications [9], [11], [19], [20], [22]. In particular, the two-state MMPP(2) has been widely used in numerous studies to model video sources and the superposition of voice sources [9], [11], [19], [22].

In this study, the traffic generated by a given node is represented by an MMPP(2) whose underlying Markov chain has two states. Each state corresponds to a different traffic arrival process. In state \( i \) (\( i=1,2 \)), the arrival traffic follows a Poisson process with rate \( \mu_i \). The transition rate out of state 1 to 2 is \( \sigma_1 \), while the rate out of state 2 to 1 is \( \sigma_2 \). The MMPP(2) is parameterised by the infinitesimal generator \( Q_s \) of the underlying Markov chain and the rate matrix \( \Lambda_s \) [7]

\[
Q_s = \begin{bmatrix}
-\sigma_1 & \sigma_1 \\
\sigma_2 & -\sigma_2
\end{bmatrix}
\]

\[
\Lambda_s = \begin{bmatrix}
\mu_1 & 0 \\
0 & \mu_2
\end{bmatrix}
\]

The traffic arrival rate of the MMPP and its covariance function, which relates to how dependent the rate at one instant of time is to the rate at another instant of time, play a major role in the method, which will be described below, for determining the traffic arrival process at network channels. The mean (Eq. 3 below), variance (Eq. 4), third central moment (Eq. 5), covariance function (Eq. 6), and the integral of the covariance function (Eq. 7), of the traffic rate are defined as

\[
\bar{\lambda}_s = \frac{\mu_1 \sigma_2 + \mu_2 \sigma_1}{\sigma_1 + \sigma_2}
\]

\[
\bar{\lambda}_s^{(2)} = \frac{\sigma_1 \sigma_2 (\mu_1 - \mu_2)^2}{(\sigma_1 + \sigma_2)^2}
\]

\[
\bar{\lambda}_s^{(3)} = \frac{\mu_1^3 \sigma_2^2 + \mu_2^3 \sigma_1^2 - (\mu_1 \sigma_2 + \mu_2 \sigma_1)^3}{(\sigma_1 + \sigma_2)^3} - \frac{3 \sigma_1 \sigma_2 (\mu_1 \sigma_2 + \mu_2 \sigma_1) (\mu_1 - \mu_2)^2}{(\sigma_1 + \sigma_2)^3}
\]
\begin{align*}
Cov(t) &= \frac{\sigma_1\sigma_2(\mu_1 - \mu_2)^2}{(\sigma_1 + \sigma_2)^2} e^{-(\sigma_1 + \sigma_2)t} \tag{6} \\
\gamma_s &= \frac{1}{A_s^{(2)}} \int_0^{\infty} Cov(t) dt = \frac{1}{\sigma_1 + \sigma_2} \tag{7}
\end{align*}

### 3.2. Assumptions of the model

The model is based on the following assumptions; while assumption (a) designates the traffic generated by the source nodes, the other assumptions have been commonly used in previous network performance studies, e.g. [2]-[4], [13]-[16] [18], [21].

- **a)** A node generates correlated traffic, which follows an independent MMPP(2), whose infinitesimal generator \( Q_s \) and rate matrix \( A_s \) are given by Eqs. 1&2.
- **b)** Message destination nodes are uniformly distributed across the network nodes.
- **c)** Message length is \( m \) flits, where \( m \) is a random variable with a Laplace-Stieltjes transform \( \mathcal{L}_m(s) \).
- **d)** The 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 at their destinations.
- **e)** Messages are routed adaptively through the network. A message always uses one of the available shortest paths to cross from source to destination. Moreover, each physical channel is divided into \( V \) virtual channels. When there are more than one available virtual channels that bring a message closer to its destination, one of them is chosen randomly.
- **f)** When a message header is blocked at an intermediate node because all the required virtual channels are busy, it experiences “connection failure”. The partial reserved path is released, and the header makes a new attempt to establish a path from the source node. Connection failures at different channels are independent of each other.

### 3.3. Derivation of the model

The mean message latency, \( \bar{T} \), is composed of the mean network latency, \( \mathcal{T} \), that is the mean time to cross the network, and the mean waiting time seen by messages at the source node, \( \bar{W}_s \). However, to model the effect of virtual channel multiplexing, the mean message latency has to be scaled by a factor, \( \bar{V} \), representing the average degree of virtual channel multiplexing, that takes place at a given physical channel. Thus, we can write \[ \bar{L} = (\mathcal{T} + \bar{W}_s)\bar{V} \tag{8} \]

Under the uniform traffic pattern, messages generated by a given node are equiprobably destined to any of the \((N-1)\) nodes in an \( n \)-dimensional cubic network. A generated message that traverses \( i \) \((1\leq i \leq n)\) channels can equiprobably reach any of the \( \binom{n}{i} \) nodes that are \( i \) hops away from its source. Thus, the probability, \( p_i \), that a generated message traverses \( i \) channels to cross the network can be written as

\begin{align*}
p_i &= \left(\frac{n}{N-1}\right)^i \quad (1 \leq i \leq n) \tag{9}
\end{align*}

Considering all the possible hops made by a message in the network yields the average number of channels, \( d \), that a typical message traverses to reach its destination as

\begin{align*}
d &= \sum_{i=1}^{n} \binom{n}{i} i \left(\frac{n}{N-1}\right)^i \quad (1 \leq i \leq n) \\
&= \frac{n}{N-1} \left(1 + \frac{1}{N-1}\right)^N - \frac{n}{N-1} \tag{10}
\end{align*}

Let \( T \) be a random variable that denotes the network latency seen by a typical message to cross the network. Given that the network latency consists of two parts: the time to set up a path and the time to transmit a message from the source to destination, \( T \) is given by

\begin{align*}
T &= C + d + m \tag{11}
\end{align*}

where \( C \) and \( m \) are random variables representing the path set-up time and the message length, respectively. Let \( F_T(x) = \text{Prob}(C + d + m \leq x) \) be the distribution function of the network latency, \( T \). Since the Laplace-Stieltjes transform of the sum of independent random variables is equal to the product of their transforms \([12]\), the Laplace-Stieltjes transform of \( T \) can be written as

\begin{align*}
\mathcal{L}_T(s) &= \int_0^{\infty} e^{-sx} dF_T(x) = \mathcal{L}_C(s)e^{-sd}\mathcal{L}_m(s) \tag{12}
\end{align*}

where \( \mathcal{L}_C(s) \) and \( \mathcal{L}_m(s) \) denote the Laplace-Stieltjes transforms of the path set-up time and the message length, respectively.

When a message header reaches a given node, this implies that it has not suffered blocking at any of the previous nodes. Using assumption (f), the probability, \( P_{F_i} \), that a header experiences a connection failure at a node that is \( i \) hops away from the source can be expressed as

\begin{align*}
P_{F_i} &= P_{b_i} \prod_{j=0}^{i-1} (1 - P_{b_j}) \tag{13}
\end{align*}

where \( P_{b_i} \) is the probability that the header suffers blocking after making \( i \) hops. Let \( P_S \) be the probability
of a successful connection, and $P_F$ be the probability of a connection failure during a single connection attempt. Since a header crosses, on average, $d$ channels to reach its destination, $P_S$ and $P_F$ are given by

$$P_S = \prod_{i=0}^{d-1} (1 - P_{b_i})$$  \hspace{1cm} (14)$$

$$P_F = 1 - P_S = 1 - \prod_{i=0}^{d-1} (1 - P_{b_i})$$  \hspace{1cm} (15)$$

In the event of a connection failure, a new attempt is made to establish a connection between the source and destination nodes. A message may need a number of, say, $r$ ($r = 1, 2, \ldots, \infty$), attempts to successfully establish a path because all the previous $(r-1)$ attempts have failed. The traffic due to the $r$-th attempt is modelled\(^1\) by an MMPP(2)\(_r\), resulting from the splitting, with the probability $P_F^{r-1}$, of the original MMPP(2)\(_s\) that characterises the traffic generated by a source node. Given that the resulting process from the splitting of an MMPP has the same underlying Markov chain as the original MMPP [17], the infinitesimal generator $Q_r$ and the rate matrix $\Lambda_r$, of the MMPP(2)\(_r\) can be written as

$$Q_r = Q_s = \begin{bmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{bmatrix}$$\hspace{1cm} (16)$$

$$\Lambda_r = P_F^{r-1} \Lambda_s = \begin{bmatrix} P_F^{r-1} \mu_1 & 0 \\ 0 & P_F^{r-1} \mu_2 \end{bmatrix}$$\hspace{1cm} (17)$$

By virtue of Eqs. 3–7, the first three moments and the integral of the covariance function of the traffic arrival rate due to the $r$-th connection attempt can be expressed as

$$\bar{\lambda}_r = P_F^{r-1} \bar{\lambda}_s$$\hspace{1cm} (18)$$

$$\bar{\lambda}_r^{(2)} = P_F^{2(r-1)} \bar{\lambda}_s^{(2)}$$\hspace{1cm} (19)$$

$$\bar{\lambda}_r^{(3)} = P_F^{3(r-1)} \bar{\lambda}_s^{(3)}$$\hspace{1cm} (20)$$

$$\gamma_r = \gamma_s$$\hspace{1cm} (21)$$

Superposing the traffic caused by all $r$, ($r = 1, 2, \ldots, \infty$), connection attempts yields the effective traffic entering the network from a given source node.

\(^1\)For the sake of clarity of the derivation of the analytical model, we always attach a subscript to the term “MMPP(2)”\(^.\) For instance, we use MMPP(2), for the traffic generated by a source node, MMPP(2)\(_s\) for the traffic due to the $r$-th connection attempt, MMPP(2)\(_s\) for the effective traffic entering the network, MMPP(2)\(_s\) for the traffic on a network channel, and MMPP(2)\(_s\) for the traffic on an injection virtual channel.

Since the superposition of MMPPs is again an MMPP [7], the effective traffic entering the network can be approximately modelled as a new MMPP(2)\(_e\). The main idea is to match the statistical characteristics of the MMPP(2)\(_s\) to those of the effective traffic [8], [9]. This can be achieved by matching the following four parameters: i) Mean arrival rate; ii) Variance of the arrival rate; iii) Third central moment; iv) Integral of the covariance function of the arrival rate.

The four characteristics of the effective traffic entering the network from a source node can be written as

$$\bar{\lambda}_e = \sum_{r=1}^{\infty} \bar{\lambda}_r = \sum_{r=1}^{\infty} P_F^{r-1} \bar{\lambda}_s = \frac{\bar{\lambda}_s}{1 - P_F}$$\hspace{1cm} (22)$$

$$\bar{\lambda}_e^{(2)} = \sum_{r=1}^{\infty} \bar{\lambda}_r^{(2)} = \sum_{r=1}^{\infty} P_F^{2(r-1)} \bar{\lambda}_s^{(2)} = \frac{\bar{\lambda}_s^{(2)}}{1 - P_F^2}$$\hspace{1cm} (23)$$

$$\bar{\lambda}_e^{(3)} = \sum_{r=1}^{\infty} \bar{\lambda}_r^{(3)} = \sum_{r=1}^{\infty} P_F^{3(r-1)} \bar{\lambda}_s^{(3)} = \frac{\bar{\lambda}_s^{(3)}}{1 - P_F^3}$$\hspace{1cm} (24)$$

$$\gamma_e = \sum_{r=1}^{\infty} \bar{\lambda}_r^{(2)} \gamma_r = \sum_{r=1}^{\infty} \bar{\lambda}_r^{(2)} \gamma_s = \gamma_s$$\hspace{1cm} (25)$$

With the above parameters expressed by Eqs. 22–25 as input parameters, the algorithm described in [8] derives the infinitesimal generator $Q_e$ and the rate matrix $\Lambda_e$ of the MMPP(2)\(_e\) that closely matches the characteristics of the traffic on a network channel. $Q_e$ and $\Lambda_e$ are found to be

$$Q_e = \begin{bmatrix} -\delta_1 & \delta_1 \\ \delta_2 & -\delta_2 \end{bmatrix}$$\hspace{1cm} (26)$$

$$\Lambda_e = \begin{bmatrix} \varphi_1 & 0 \\ 0 & \varphi_2 \end{bmatrix}$$\hspace{1cm} (27)$$

A message may encounter blocking at any of the $d$ intermediate nodes along its path. Taking into account the cases of a connection success and connection failures occurring at $d$ possible nodes gives the average number of channels, $b$, traversed by a message during a single connection attempt as

$$b = dP_S + \sum_{i=0}^{d-1} iP_{F_i}$$

$$= \prod_{i=0}^{d-1} d(1 - P_{b_i}) + \sum_{i=0}^{d-1} iP_{b_i} \prod_{j=0}^{i-1} (1 - P_{b_j})$$\hspace{1cm} (28)$$

Under the uniform traffic pattern, using adaptive routing results in a balanced traffic load on all network channels; the arrival process at the network channels
exhibits similar statistical behaviour. Examining Eqs. 10 & 28 reveals that the average number of channels, \( b \), traversed by a message during a single connection attempt is always less than \( n \) in an \( n \)-dimensional cubic network. This means that the arrival traffic at a given network channel is a fraction of the effective traffic entering into the network from a source node. This fraction, \( f \), can be estimated by

\[
f = \frac{N_b}{N_n} = \frac{b}{n}
\]

(29)

Given that the MMPP is closed under the superposition and splitting operations, we use an MMPP(2) to approximate the characteristics of the traffic arriving at a network channel. As the traffic on a channel is a fraction of the effective traffic, the MMPP(2) can be obtained by splitting the MMPP(2) characterising the effective traffic with the splitting probability \( f \). The simulation results presented below will show that this is an effective approach for making the model tractable while maintaining a good degree of accuracy. The infinitesimal generator \( \mathbf{Q}_e \) and the rate matrix \( \mathbf{\Lambda}_e \) of the MMPP(2) are given by [17]

\[
\mathbf{Q}_e = \mathbf{Q}_e = \begin{bmatrix} -\delta_1 & \delta_1 \\ \delta_2 & -\delta_2 \end{bmatrix}
\]

(30)

\[
\mathbf{\Lambda}_e = f\mathbf{\Lambda}_e = \begin{bmatrix} f\theta_1 & 0 \\ 0 & f\theta_2 \end{bmatrix}
\]

(31)

Messages generated by a source node follow an MMPP(2) and enter the network through one of the injection virtual channels with equal probability \( 1/V \). The infinitesimal generator \( \mathbf{Q}_v \) and the rate matrix \( \mathbf{\Lambda}_v \), of the resulting MMPP(2), characterising the traffic arriving at an injection virtual channel in the source node are given by

\[
\mathbf{Q}_v = \begin{bmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{bmatrix}
\]

(32)

\[
\mathbf{\Lambda}_v = \begin{bmatrix} \mu_1/V & 0 \\ 0 & \mu_2/V \end{bmatrix}
\]

(33)

To determine the mean waiting time, \( \bar{W} \), that a message experiences before entering the network, the injection virtual channel is modelled as an MMPP/G/1 queueing system. The uniformity of traffic in the network and the use of adaptive routing make the service time seen by messages in all source nodes equal to the network latency, i.e. \( T \) [18] (whose Laplace-Stieltjes transform is given by Eq. 12). Using the results of Ref. [7], The mean virtual waiting time, \( \bar{W} \), and the mean actual waiting time, \( \bar{W}_s \), can be expressed as

\[
\bar{W} = \frac{(2\rho + \lambda_2 \bar{T}^{(2)} - 2\bar{T}((1 - \rho)\mathbf{g} + \bar{T}\pi\mathbf{\Lambda}_v))(\mathbf{Q}_s + \mathbf{e}\pi)^{-1}\lambda)}{2(1 - \rho)}
\]

\[
\bar{W}_s = \frac{(2\bar{W} - \lambda_2 \bar{T}^{(2)})}{2\rho}
\]

(34)

In the above equations, \( \bar{T} \) and \( \bar{T}^{(2)} \) denote the first two moments of the message service time. These quantities are computed by differentiating \( L_T(s) \) and setting \( s = 0 \) [12]. \( \lambda_2 \) is the mean traffic rate arriving at an injection virtual channel and given by \( \lambda_2 = \pi \mathbf{d} \), where \( \pi \) is the steady-state vector of the MMPP(2), and can be written as

\[
\pi = \frac{1}{\sigma_1 + \sigma_2}(\sigma_2, \sigma_1), \quad \mathbf{d} = \frac{1}{V}(\mu_1, \mu_2).
\]

The traffic intensity, \( \rho \), is equal to \( \bar{T}\lambda_2 \), and \( \mathbf{e} = (1,1)^T \) is a unit column vector. The algorithm for calculating the matrix \( \mathbf{g} \), used in Eq. 34, has been described in [7]. Examining the algorithm reveals that the Laplace-Stieltjes transform, \( L_T(s) \), is also required to compute \( \mathbf{g} \).

As depicted in Eq. 12, in order to derive \( L_T(s) \) we need to compute the Laplace-Stieltjes transform, \( L_C(s) \), of the path set-up time. To do this, let us first derive the mean time to establish a path, \( \bar{C} \), as follows. Consider a message header that is currently at a node being \( i \) hops away from the source. Let \( P_{b_i} \), \( (0 \leq i \leq d - 1) \), be the probability that the header is blocked and \( \bar{C}_i \) denote the expected duration for the header to reach the destination from the current node. If the header succeeds in reserving the required virtual channel and advances to the next node, the residual expected duration becomes \( \bar{C}_{i+1} \). This case occurs with probability \( (1 - P_{b_i}) \). On the other hand, if the header encounters blocking and backtracks to the source node, the residual expected duration is \( \bar{C}_0 \). Given that the header requires one cycle to move from one node to the next, the above argument reveals that the expected duration \( \bar{C}_i \) satisfies the following difference equations [15]

\[
\bar{C}_i = \begin{cases} 
(1 - P_{b_i})(\bar{C}_{i+1} + 1) + P_{b_i}(\bar{C}_0 + i) \quad (0 \leq i \leq d - 1) \\
0 \quad (i = d) 
\end{cases}
\]

(35)

Solving the above equations yields \( \bar{C}_0 \). Since \( d \) cycles are required to send an acknowledgement flit back to the source node, signaling successful path establishment, the mean time to set up a path, \( \bar{C} \), can therefore be written as

\[
\bar{C} = \bar{C}_0 + d
\]

(36)
Unfortunately, finding the exact expression of the distribution function for the path set-up time from the above equations is infeasible [12]. Since we are driven by the requirement of analytic simplicity as well as the desire for versatility and practicality, we adopt the following approach. Given that any distribution function can be matched as closely as desired by a series-parallel stage-type device [12], we approximate the distribution of the path set-up time by an exponential stage since it has a simple profile and can capture the first two moments of the distribution. This method has been widely used to model the waiting time distribution at a queueing system [10], [12]. Thus, the probability density function of the path set-up time, \( f_C(x) \), and its Laplace-Stieltjes transform, \( L_C(s) \), can be expressed as
\[
f_C(x) = \alpha e^{-\beta x} \quad (\alpha > 0, \beta > 0)
\]
\[
L_C(s) = \frac{\alpha}{s + \beta}
\]
where \( \alpha \) and \( \beta \) are selected to match the first two moments, \( \overline{C} \) and \( \overline{C}^{(2)} \), of the path set-up time, and are found to be
\[
\alpha = \frac{4\overline{C}^3}{(\overline{C}^{(2)})^2}
\]
\[
\beta = 2\overline{C} / \overline{C}^{(2)}
\]
(37)
(38)
(39)
(40)
While \( \overline{C} \) is computed in Eq. 36, we need to calculate the second moment, \( \overline{C}^{(2)} \). When the header does not suffer any blocking, the minimum path set-up time is \( 2d \) (\( d \) is given by Eq. 10). Following the suggestion in Ref. [4], the variance of the path set-up time can be estimated as \( (\overline{C} - 2d)^2 \). Therefore \( \overline{C}^{(2)} \) is given by
\[
\overline{C}^{(2)} = \overline{C}^2 + (\overline{C} - 2d)^2
\]
(41)
The probability that a message header is blocked at a given node depends on its current network position. This is because the number of alternative paths that a header can take to progress changes as it advances towards the destination. When the header has made \( i \) hops, it can select any available \( (d - i)W \) virtual channels from the remaining \( (d - i) \) dimensions. If all these virtual channels are busy, blocking occurs and the header has to backtrack to the source node. Let \( P_{b_{\theta}} \), \( (\theta = 1,2) \), represent the joint probability that \( V \) virtual channels are busy and the MMPP\((2)_k \) characterising the traffic on the network channel is at state \( \theta \). The probability of blocking, \( P_{b_{j}} \), can therefore be written as
\[
P_{b_{j}} = (P_{b_{1}} + P_{b_{2}})^{d-j}
\]
(41)
The derivations of the probability, \( P_{\theta, b} \), and the average degree of virtual channel multiplexing, \( \overline{V} \) can be found in Ref. [16].

4. Validation of the model

We have developed a discrete-event simulator, operating at the flit level, in order to validate the above analytical model. The cycle time in the simulator is defined as the transmission time of a single flit to cross from one node to the next. Messages are generated at each node according to the MMPP\((2)_k \) with the infinitesimal generator \( Q_k \) and the rate matrix \( \Lambda_k \) (given by Eqs. 1&2). Message destinations are uniformly distributed across the network, and are determined using a uniform random number generator. The mean message latency (\( \overline{L} \)) is defined as the mean amount of time from the generation of a message until the last data flit reaches the local PE at the destination node. The other measures include the mean network latency (\( \overline{T} \)), that is the time taken to cross the network, and the mean waiting time (\( \overline{W} \)) at the source node, i.e. the time spent at the source node before entering the first network channel.

Numerous validation experiments have been performed for several combinations of network sizes, message lengths, numbers of virtual channels per physical channel and different MMPP\((2)_k \) input traffic. However, for the sake of specific illustration, latency results are presented for the following cases only.

- Network size is \( N=2^4 \) and \( 2^6 \) nodes.
- Number of virtual channels \( V=4, 5, 7 \) and 8.
- Message length is fixed at \( m=32 \) and 64 flits.
- Infinitesimal generator, \( Q_k \), is set as follows representing different degrees of burstiness

\[
Q_k = \begin{bmatrix}
-0.07 & 0.07 \\
0.05 & -0.05 \\
-0.4 & 0.4 \\
0.2 & -0.2
\end{bmatrix}, \quad Q_k = \begin{bmatrix}
-0.01 & 0.01 \\
0.01 & -0.01
\end{bmatrix}
\]

Figures 2–3 depict results for the mean message latency predicted by the above model plotted against those provided by the simulator as a function of the generated traffic in the 4 and 6-dimensional cubic networks, respectively. In all the figures presented below, the horizontal axis represents the traffic rate, \( \mu_1 \), at which a node injects messages into the network when the
MMPP(2), input traffic is at state 1, while the vertical axis gives the mean message latency. We have deliberately set the arrival rate, $\mu_2$, at state 2 at zero for the sake of clarity of the figures; otherwise we would have to use three-dimensional graphs to represent the results. The figures reveal that the simulation results match those predicted by the analytical model in the steady region, that is, under light, moderate traffic, and when the network enters the heavy traffic region. However, some discrepancies are apparent as the network approaches the saturation point.

This is mainly due to the approximations that have been made to ease the development of the analytical model, like the one made for determining the distribution of the path set-up time (Eq. 37). Nevertheless, it can be concluded that the proposed model produces accurate results in the steady regions, which are the regions of interests in most network evaluation studies, and its simplicity and tractability make it a practical and cost-effective evaluation tool to study the performance behaviour of cubic network networks under busy traffic.

Figure 2: Latency predicted by the model and simulation in the 4-dimensional cubic network, a) $V=4$, $\sigma_1 = 0.07$, $\sigma_2 = 0.05$, and b) $V=7$, $\sigma_1 = 0.6$, $\sigma_2 = 0.6$.

Figure 3: Latency predicted by the model and simulation in the 6-dimensional cubic network, a) $V=5$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, and b) $V=8$, $\sigma_1 = 0.01$, $\sigma_2 = 0.01$. 
5. Conclusions

Many network studies have revealed that multimedia traffic is strongly correlated. This paper has proposed a new analytical model to calculate message latency in circuit-switched cubic networks under correlated traffic. The derivation of the present analytical model exhibits more complexity than that under the traditional Poisson process. The model uses the approach of superposing infinite correlated traffic streams to capture the effective traffic entering the network from a source node in circuit switching and employs MMPP/G/1 queueing systems for calculating message waiting times. Results obtained from simulation experiments confirm that the proposed model exhibits a good degree of accuracy for various network sizes and under different operating conditions.

References