Circuit Switching with Input Queueing: An Analysis for the d-dimensional Wraparound Mesh and the Hypercube

Vishal Sharma, *Student Member, IEEE*, and Emmanouel A. Varvarigos

The authors are with the Data Transmission and Networking Lab., Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.
Email: vishal@spetses.ece.ucsb.edu, and manos@ece.ucsb.edu.

Abstract

We analyze circuit switching in a multiprocessor network, where connection requests (or sessions) arrive at each node of the network according to a Poisson process with rate λ. Each session joins the appropriate input-queue at its source node, and, upon advancing to the head of the queue, transmits a set-up packet to establish a connection. If the set-up packet is successful, it reserves the links on the path for the duration of the session, and the session is served without interruptions. Otherwise, the connection request remains queued at the source, and subsequent attempts are made to establish the circuit. We analyze the queue of connection requests at the input-buffer of a network link, and obtain analytic expressions for the stability region, the average queueing delay, the average connection time, the average waiting time, and the average total delay, which show how these parameters depend on system variables, such as network dimension and session arrival rate. The queueing analysis focuses on the input-queue of a particular link, and accounts for the interactions with queues of other links through the retrial attempts and the associated probability of success. The queueing analysis is independent of the particular network topology under consideration, as long as the probability that a session arriving at a random time successfully establishes a connection can be calculated for that network, and is also independent of the particular distribution of the session holding times. Simulations demonstrate the close agreement between the observed network behavior and that predicted by the analysis.

Keywords

Circuit switching, input queueing, interconnection networks, wraparound meshes, hypercubes, queueing delay, connection delay, stability region.

I. Introduction

Of the variety of switching and routing formats (circuit switching, packet switching, wormhole routing, and cut-through routing) used in multi-processor systems today, circuit switching combines many well-known advantages. After the circuit is setup, each message interacts only with the processors at the end nodes, thus freeing the processors at the intermediate nodes from the details of switching. Furthermore, unlike networks using worm-hole routing, networks employing circuit switching do not require that the routing algorithm be deadlock free. Also, once a connection has been established, circuit switching provides the session with an agreed upon grade of service (assuming no fault occurs), including guaranteed bandwidth and delay.

Although circuit switching (or its variations like wormhole routing) has been studied by a number of researchers in several regular topology networks under a variety of assumptions (see, for example, [GeR88], [GeR89], [ALM91], [BrS91], [CGK91], [CsS92], [ChS93], and [CCR94] for the hypercube network, [Dal90], [You92], and [GaY93] for the mesh network, [DaS87] for the torus network, and [KrS83] for banyan networks), the analysis of the steady-state throughput and delay of circuit-switched routing schemes in such networks has received relatively little attention in the literature.

One line of research in these works focuses on studying *static* circuit-switched permutation routing, that is, the problem of establishing edge-disjoint paths between the nodes of the network for any permutation mapping (see, for example, [KrS83], [ALM91], and [ChS93]). A second line of research focuses on developing efficient routing algorithms for circuit switching in regular networks and evaluating their performance (see, for example [CsS92], [You92], and [CCR94]). A third line of research involves simulating several message routing schemes in such networks and developing analytical models to explain the network behavior. For example, Grunwald and Reed [GeR88] implemented several variations of circuit switching in a 32-node hypercube, and obtained simulation results for the end-to-end total delay. They also provided an explanation of the observed behavior by using a simple model for message delay. In a later work [GeR89], they obtained recursive equations describing the total time to route a message and the expected number of retrials, for several adaptive routing algorithms that used circuit switching to establish the connection, and they used numerical methods to solve them. Their analysis assumes that the acquisition of each link on a path corresponds to an independent Bernoulli trail, and that link utilization is independent of the arrival rate of sessions. Finally, a fourth line of research, and the one closest to our approach, involves formulating analytical models for the performance of *dynamic* circuit switching in regular networks (see, for example, [CGK91] and [BrS91]). Bruno and Salvatore [BrS91] analyzed circuit switching in the hypercube by taking into account the interactions between different sessions. Their analysis also assumes, however, that the acquisition of each link on the path is independent of the number of links acquired prior to it. They also performed an average case analysis, where they considered a "typical" session with path length equal to the mean internodal distance.

In this paper, we analyze circuit switching in d-dimensional wraparound meshes and in hypercubes. We focus first on the the d-dimensional wraparound mesh and analyze it in detail, and then show how the analysis can be applied to the hypercube network. We assume that connection requests are generated at each node of the mesh according to a Poisson process with rate λ independently of other nodes, and that session destinations are uniformly distributed over the remaining nodes. In our routing scheme, a $(d!)$-sided die is tossed in order to decide which one of the $d!$ possible paths (some of them identical) a new session will use, thus ensuring that link utilization is uniform across all links of the network. Depending on the outcome of this experiment and the destination of a session, the session joins one of the $2d$ link-input queues (d link input-queues in the case of the hypercube) of the source node. Each link input-queue has infinite buffer space and uses a FIFO queueing discipline. When all sessions ahead of it in the queue have been served, the session advances to the head of the queue and a set-up packet is sent to establish a circuit. If the set-up packet is successful in establishing a connection, the links required by the circuit are reserved for the session duration, and the session is served without interruptions. If the circuit cannot be established, the server takes a vacation (the details of which we provide in Section 4), and tries again. This process is repeated until a connection for the session is finally established.

We first obtain an analytic expression for the steady-state probability $P_{success}$ that a new session arriving at a random time successfully establishes a connection. We use this expression to obtain analytical results for the average queueing delay, the average connection delay, the average waiting time, and the average total delay required to serve a connection request in a d-dimensional mesh. We also find the maximum throughput for the circuit switching scheme, that is, the maximum λ for which the average total delay is finite. The queueing analysis that we develop depends on the topology under consideration only through the steady-state probability $P_{success}$, and the number of link-input queues at a node.

In our analysis we account for the dependence between link utilization and the probability of successfully establishing a circuit. In evaluating the latter probability, we account approximately for the dependence between the acquisition of successive links on the path followed by a session. The various queueing delay parameters are functions of the probability of success $P_{success}$, and they depend on it in a simple way. Our queueing analysis is independent of the specific distribution of the session holding times (it does not, for example, require that the holding times be exponentially distributed), and results in closed form expressions for all network delay parameters of interest such as the queueing delay, the connection delay, and the total delay of a session. We also obtain asymptotic expressions for the queueing parameters of interest, as the number of nodes of the mesh tends to infinity. The close agreement between the analytically predicted values of the delay parameters and those obtained by simulations indicates that our analysis is very close to being accurate.

The remainder of the paper is organized as follows. In Section 2 we introduce some definitions, describe the network model, and give some preliminary results. In Section 3 we derive an approximate expression for the probability that a session arriving at a random time successfully establishes a connection, and compare it to simulation results. The queueing analysis of a link input-buffer is presented in Section 4, where we also compare the analytical results with those obtained by simulations. In Section 5 we specialize our results to the case of large mesh sizes, and obtain asymptotic expressions for various parameters of interest. In Section 6 we extend the analysis to the hypercube network. We compare the delays predicted analytically with those obtained by simulations. In Section 7 we conclude the paper, and in the appendices we present some auxiliary results, and we resolve some technical issues that arise throughout the paper.

II. D-DIMENSIONAL MESH COMMUNICATION MODEL AND PRELIMINARY RESULTS

In this section we describe the d-dimensional mesh network of processors, and the communication model used. We also present preliminary results that will be useful in our analysis.

The d-dimensional wraparound mesh, denoted by M_d, consists of $N = p^d$ processors arranged along the points of a d-dimensional space that have integer coordinates. Along the i^{th} dimension, obtained by fixing coordinates $x_d, \ldots, x_{i+1}, x_{i-1}, \ldots, x_1$, there are p processors with identities $(x_d, \ldots, x_i, \ldots, x_1)$, $x_i = 0, 1, \ldots, p-1$. Two processors $(x_d, \ldots, x_i, \ldots, x_1)$ and $(y_d, \ldots, y_i, \ldots, y_1)$ are connected by a (bi-directional) link if and only if for some i we have $|x_i - y_i| = 1$ and $x_j = y_j$ for all $j \neq i$. In addition to these links, wraparound links of the form

$$\Big((x_d, \ldots, x_{i+1}, 1, x_{i-1}, \ldots, x_1), (x_d, \ldots, x_{i+1}, p, x_{i-1}, \ldots, x_1)\Big)$$

are also present. A node with identity $x = (x_d, x_{d-1}, \ldots, x_1)$ is also represented by the base p number of the form $x = x_d x_{d-1} \ldots x_1$. A link connecting two nodes that differ only in the i^{th} digit is called a *link of dimension i*. The *routing tag* of a session with source node $x = (x_d, \ldots, x_1)$ and destination node $y = (y_d, \ldots, y_1)$, is defined as $(t_d, \ldots, t_1)$, where

$$t_j = \begin{cases} y_j - x_j, & \text{if } |y_j - x_j| \leq \lfloor \frac{p}{2} \rfloor; \\ y_j - x_j - p \cdot \mathrm{sgn}(y_j - x_j), & \text{if } |y_j - x_j| > \lfloor \frac{p}{2} \rfloor, \end{cases}$$

for all $j \in \{1, 2, \ldots, d\}$, and $\mathrm{sgn}(x)$ is the signum function, which is equal to $+1$ if $x \geq 0$, and equal to -1, otherwise. The network diameter is equal to dD, where $D = \lfloor p/2 \rfloor$, and the mean internodal distance $\overline{h}$ is calculated to be

$$\left(\frac{p^2 - 1}{4}\right) \frac{dp^{d-1}}{p^d - 1}$$

if p is odd, and

$$\left(\frac{p^2}{4}\right) \frac{dp^{d-1}}{p^d - 1}$$

if p is even.

Connection requests (or sessions) are generated at each node of the mesh at rate λ per unit time, and have uniformly distributed destinations. A circuit between the source node and the destination node is established by sending a set-up packet along a path determined at the source. If a set-up packet finds all links on that path available, it reserves them for a duration equal to the holding time X of the session, where X may be any random variable whose mean $\overline{X}$, and second and third moments $\overline{X^2}$ and $\overline{X^3}$ are known. We assume that the input-buffer has infinite buffer space, so that no connection requests are lost (otherwise, sessions with longer paths would be treated unfairly, since they would be dropped with higher probability). Similar assumptions are common in current literature, and have been used by researchers both for the analysis of switching schemes in multi-processor networks (see, for example, [ReF87], [ReG87], [AbP89], and [BYA89] and the references therein) and for simulations of actual networks (see, for example, [GeR88] and [GeR89]).

In our model, all incident links of a node can be used simultaneously for message transmission and reception, and messages can be transmitted simultaneously along a link in both directions. We only allow shortest paths where all links of a given dimension are traversed first, followed by all links of a different dimension and so on, until all dimensions are exhausted. In other words, during phase i, $i = 1, \ldots, d$, all links of dimension $\pi(i)$ are traversed, where π is a permutation of $\{1, 2, \ldots, d\}$. For the d-dimensional torus, there are $d!$ such paths (some of them possibly identical) for a given source-destination pair. Our routing scheme is oblivious or non-adaptive, that is, the setup packet of a session chooses an order in which the dimensions are traversed and it insists on that order as it progresses from its source to its destination. Since in a circuit switching scheme the holding time of a session is expected to be much larger than the connection setup time (otherwise, circuit switching would not be very efficient), we simplify modeling by assuming that once the source node determines that the required links are available, the session acquires them instantaneously. In other words, the circuit is not established on a link-by-link basis but reserves all required links simultaneously. Similarly, all links used by a session are released simultaneously upon its completion. This allows us to concentrate on the main features of a circuit switching scheme without having to focus on the technical details of a specific implementation, which would complicate the analysis without providing any additional insight. In the simulation results presented in Section 4, however, we examine the effect of the setup time overhead, and show that the simulations and analysis agree closely when the setup overhead is taken into account.

A network link L is at any time in one of the four states illustrated in Fig. 1. We denote by q_i, $i = 0, 1, 2, 3$, the steady-state probability that link L is in state i. Clearly, we have

$$q_0 + q_1 + q_2 + q_3 = 1. \tag{1}$$

Consider a session that chooses one of the $d!$ paths to establish a circuit to its destination. We let $I(i_d, i_{d-1}, \ldots, i_1)$ be the number of non-zero entries in $(i_d, i_{d-1}, \ldots, i_1)$. We also define

$$F_1(i_d, i_{d-1}, \ldots, i_1) = I(i_d, i_{d-1}, \ldots, i_1) - 1, \tag{2}$$

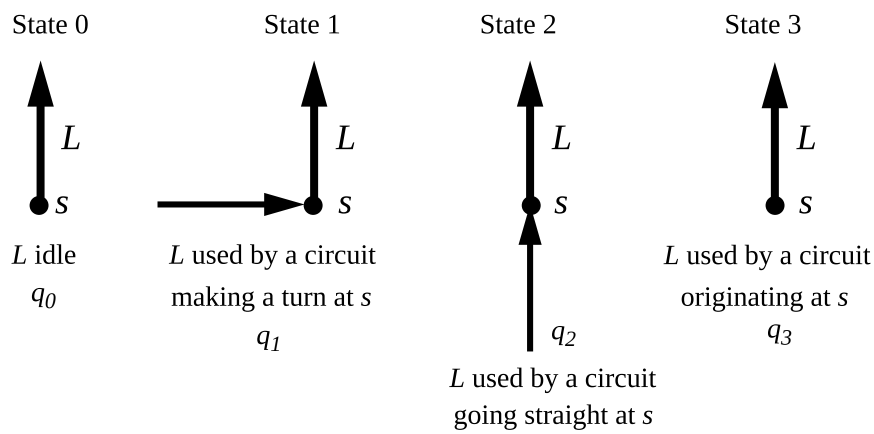

Fig. 1. The four possible states of a network link L are illustrated. State 0 corresponds to the case where L is idle. State 1 corresponds to the case where L is used by a circuit that is making a turn at node s. (Note that State 1 is made up of $2k-2$ substates, depending upon the direction from which the path makes a turn.) State 2 corresponds to the case where L is used by a circuit that is going straight at s. State 3 corresponds to the case where L is used by a circuit originating at node s.

and

$$\begin{aligned} F_2 \quad & (i_d, i_{d-1}, \ldots, i_1) \\ = \quad & |i_d| + |i_{d-1}| + \cdots + |i_1| - I(i_d, i_{d-1}, \ldots, i_1).(3) \end{aligned}$$

When a circuit for a session with routing tag $(t_d, t_{d-1}, \ldots, t_1) \neq (0, 0, \ldots, 0)$ is established, it will contain $F_1(t_d, t_{d-1}, \ldots, t_1)$ links in state 1, $F_2(t_d, t_{d-1}, \ldots, t_1)$ links in state 2, and one link in state 3.

The steady-state probability q_1 that a link is in state 1 is given by

$$q_1 = K \sum_{\substack{t_d,\ldots,t_1 \\ t_d \neq 0,\ldots,t_1 \neq 0}} F_1(t_d, t_{d-1}, \ldots, t_1), \tag{4a}$$

where K is some constant of proportionality to be derived later.
After some calculations, shown in Appendix A1, we obtain

$$q_1 = K[d(p-1)p^{d-1} - p^d + 1]. \tag{4b}$$

The probability q_2 that a link is in state 2 satisfies

$$q_2 = K \sum_{\substack{t_d,\ldots,t_1 \\ t_d \neq 0,\ldots,t_1 \neq 0}} F_2(t_d, t_{d-1}, \ldots, t_1). \tag{5a}$$

After some algebraic manipulation, shown in Appendix A1, we obtain

$$q_2 = \begin{cases} Kdp^{d-1}\frac{(p-1)(p-3)}{4}, & \text{if } p \text{ is odd;} \\ Kdp^{d-1}\frac{(p-2)^2}{4}, & \text{if } p \text{ is even.} \end{cases} \tag{5b}$$

Similarly, the probability that a given link is in state 3 is given by

$$q_3 = K \sum_{\substack{t_d,\ldots,t_1 \\ t_d\neq 0,\ldots,t_1\neq 0}} 1 = K(N-1). \tag{6}$$

To obtain Eqs. (4)-(6) we used the fact that the probability that a link is in state i is proportional to the total number of ways in which a link can be in state i. Since no sessions are dropped and the session destinations are uniformly distributed over all nodes except the source, each value of the routing tag is equally likely and contributes $F_1(t_d, t_{d-1}, \ldots, t_1)$ links to state 1, $F_2(t_d, t_{d-1}, \ldots, t_1)$ links to state 2, and one link to state 3. Therefore, the probability of a link being in state i is proportional to the sum of the corresponding one of the quantities above over all possible values of the routing tag, with the same constant of proportionality K.

Solving Eq. (1) together with Eqs. (4)-(6) with respect to K, we obtain for p odd

$$K = \frac{4(1-q_0)}{dp^{d-1}(p^2-1)}, \qquad (p \text{ odd})$$

which implies

$$q_1 = \frac{4(1-q_0)}{dp^{d-1}(p^2-1)}(d(p-1)p^{d-1} - (p^d-1)), \qquad (p \text{ odd}) \tag{7a}$$

$$q_2 = \frac{(1-q_0)(p-3)}{(p+1)}, \qquad (p \text{ odd}) \tag{7b}$$

and

$$q_3 = \frac{4(1-q_0)(p^d-1)}{dp^{d-1}(p^2-1)}. \qquad (p \text{ odd}) \tag{7c}$$

A similar calculation for p even, gives

$$K = \frac{4(1-q_0)}{dp^{d-1}p^2}, \qquad (p \text{ even})$$

which implies

$$q_1 = \frac{4(1-q_0)}{dp^{d-1}p^2}(d(p-1)p^{d-1} - (p^d-1)), \qquad (p \text{ even}) \tag{8a}$$

$$q_2 = \frac{(1-q_0)(p-2)^2}{p^2}, \qquad (p \text{ even}) \tag{8b}$$

and

$$q_3 = \frac{4(1-q_0)(p^d-1)}{dp^{d+1}}. \qquad (p \text{ even}) \tag{8c}$$

Using Little's theorem, the average number of links used is equal to the product of the average number of active circuits in the system and the mean internodal distance, that is,

$$2dp^d(1-q_0) = \begin{cases} p^d\lambda\overline{X}\left(\frac{p^2-1}{4}\right)\frac{dp^{d-1}}{p^d-1}, & \text{for } p \text{ odd;} \\ p^d\lambda\overline{X}\left(\frac{p^2}{4}\right)\frac{dp^{d-1}}{p^d-1}, & \text{for } p \text{ even,} \end{cases}$$

or

$$1-q_0 = \begin{cases} \frac{\lambda\overline{X}p^{d-1}\left(p^2-1\right)}{8\left(p^d-1\right)}, & \text{for } p \text{ odd;} \\ \frac{\lambda\overline{X}p^{d-1}p^2}{8\left(p^d-1\right)}, & \text{for } p \text{ even,} \end{cases} \tag{9}$$

where λ is the total number of connection requests generated at each node of the mesh per unit of time, and $\overline{X}$ is the average amount of time the connection is active. Substituting q_0 from Eq. (9) to the expressions for q_1, q_2, and q_3 we obtain

$$q_1 = \frac{\lambda\overline{X}(d(p-1)p^{d-1}-(p^d-1))}{2d(p^d-1)},$$

$$q_2 = \begin{cases} \frac{\lambda\overline{X}p^{d-1}(p-1)(p-3)}{8(p^d-1)}, & \text{for } p \text{ odd,} \\ \frac{\lambda\overline{X}p^{d-1}(p-2)^2}{8(p^d-1)}, & \text{for } p \text{ even,} \end{cases}$$

and

$$q_3 = \frac{\lambda\overline{X}}{2d}.$$

III. Probability of successfully establishing a circuit connection

The probability that a new session arriving at a random time successfully establishes a circuit connection from its source node to its destination node depends on its routing tag $(t_d, t_{d-1}, \ldots, t_1)$. We denote by $P_{success}(t_d, t_{d-1}, \ldots, t_1)$ the probability that a new session will succeed in establishing a connection in its first trial (subsequent trials are more complicated), given that it has a routing tag $(t_d, t_{d-1}, \ldots, t_1)$. In this section, we find an approximate expression for $P_{success}(t_d, t_{d-1}, \ldots, t_1)$, and from it an expression for $P_{success}$, the unconditional probability that an arriving session successfully establishes a circuit.

As explained in Section 2, the path followed by a session with routing tag $(t_d, t_{d-1}, \ldots, t_1)$ will make $F_1(t_d, t_{d-1}, \ldots, t_1)$ turns along the way, and will go straight for a total of $F_2(t_d, t_{d-1}, \ldots, t_1)$ steps. The first link of the path is available with probability q_0. At each subsequent step at which the path does not make a turn, the probability that the next link L is available given that the previous link was available is approximately equal to

$$\Pr(L \text{ available} \mid L \text{ not in state } 2) = \frac{q_0}{1-q_2} \stackrel{\text{def}}{=} \alpha. \tag{10}$$

The probability that the first link L after a turn is available is approximately equal to

$$\begin{aligned} \Pr & \quad (L \text{ available} \mid L \text{ not in a given substate of state } 1) \\ = & \quad \frac{q_0}{1-\frac{q_1}{(2d-2)}} \stackrel{\text{def}}{=} \beta.(11) \end{aligned}$$

To see that note that state 1 is composed of $2d-2$ substates depending on the direction from which the path makes a turn at node s into link L (see Fig. 1). (Since the path is making a turn from some other dimension we know that link L cannot be in the corresponding substate of state 1.) Since each of these substates is equally likely, the probability of a link being in one of these substates is $\frac{q_1}{2d-2}$.

The (conditional) probability of successfully establishing a connection is then given by

$$P_{success}(t_d, t_{d-1}, \ldots, t_1) = q_0\, \alpha^{F_2(t_d, t_{d-1}, \ldots, t_1)} \beta^{F_1(t_d, t_{d-1}, \ldots, t_1)},$$

where $F_1(t_d, t_{d-1}, \ldots, t_1)$ and $F_2(t_d, t_{d-1}, \ldots, t_1)$ are defined in Eqs. (2) and (3), respectively.

The (unconditional) probability of successfully establishing a connection is given by

$$P_{success} = \sum_{t_d, t_{d-1}, \ldots, t_1} P_{success}(t_d, t_{d-1}, \ldots, t_1) \cdot \Pr(\text{routing tag } = (t_d, t_{d-1}, \ldots, t_1)). \tag{12}$$

When the destinations are uniformly distributed, Eq. (12) simplifies to

$$P_{success} = \frac{1}{p^d - 1} \sum_{t_d, t_{d-1}, \ldots, t_1} P_{success}(t_d, t_{d-1}, \ldots, t_1).$$

After some calculations, given in Appendix A2, the above expression yields

$$P_{success} = \frac{q_0}{\beta(p^d-1)} \left[\left(1 + 2\beta \left(\frac{1-\alpha^D}{1-\alpha} \right) \right)^d - 1 \right] \tag{13}$$

for p odd, and

$$P_{success} = \frac{q_0}{\beta(p^d-1)} \times \left[\left(1+\beta\left(\frac{1-\alpha^D}{1-\alpha}\right)+\beta\left(\frac{1-\alpha^{D-1}}{1-\alpha}\right)\right)^d - 1\right] \quad (14)$$

for p even, where α and β are given by Eqs. (10) and (11), respectively. Observe that since in our model all arriving sessions are queued, the loading of the network does not depend on the distribution of the resources used by successful connections. As a result, while the probability of success $P_{success}$ depends on the session arrival rate λ, the arrival rate itself is independent of the probability of success, so that a closed-form solution can be obtained for $P_{success}$, without the need for a fixed-point iteration. By contrast, models in which connections are dropped either after a time out or due to buffer overflow [CGK91], require a fixed-point iteration to determine the network load.

Figure 2 illustrates the analytical and simulation results obtained for the probability of success as a function of the arrival rate λ for various mesh sizes and dimension $d = 2$. As is evident from the figure, the agreement between the simulations and the analytically obtained expressions is very good.

IV. Queuing analysis at link input-buffer

In this section we describe in more detail the node model, and present the analysis of the input-buffer queue.

A. Node Model

Each node has $2d$ input-buffers, one for each outgoing link of the node. We denote by $\mathcal{B}_L^s$ the input-buffer of link L of node s. Each input-buffer is used for storing new sessions that are generated at the node and are going to start transmission over that link. Sessions that are transiting through node s via link L do not occupy any space in $\mathcal{B}_L^s$ (in fact, since circuit switching is assumed, there is no queueing of such sessions because resources have been reserved in advance). We assume a FIFO queueing discipline, and infinite buffer space for each link input-buffer. The queue of a link input-buffer of a node is interacting with corresponding queues of other nodes and with input-buffer queues of other links of the same node. This is because a session's attempt to establish a connection will be influenced

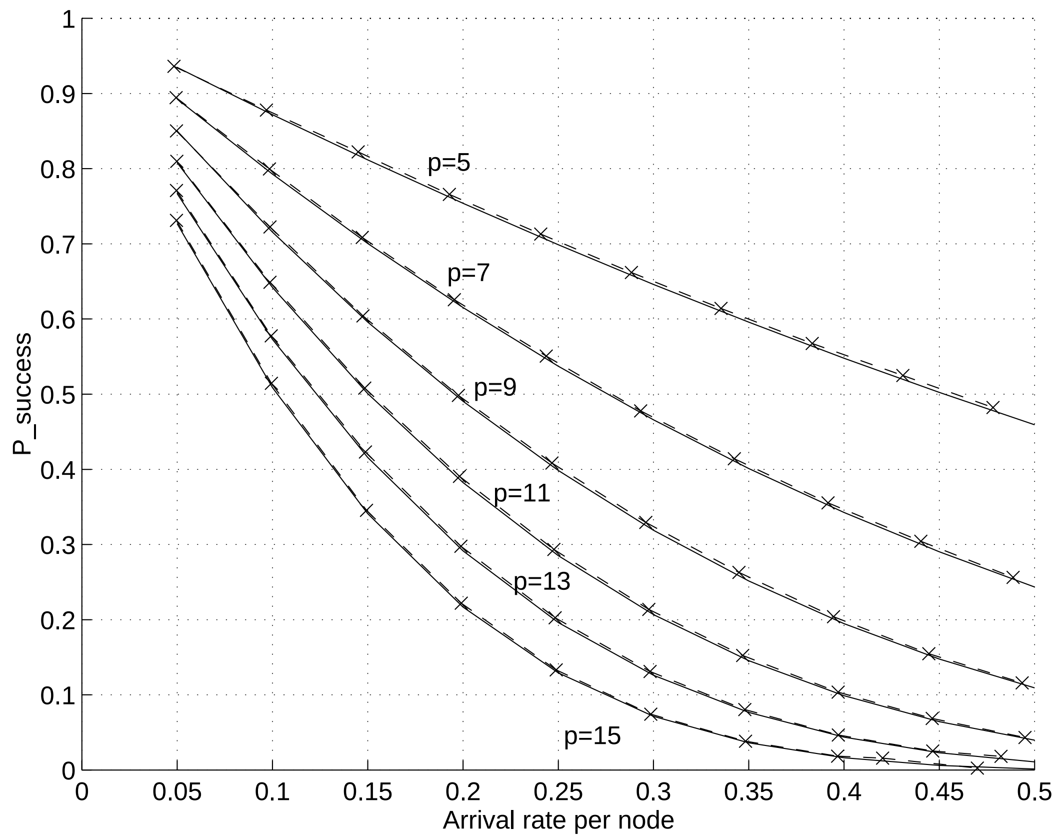

Fig. 2. Analytical and simulation results obtained for the probability of success as a function of the arrival rate λ of sessions, for $d = 2$. All curves correspond to exponentially distributed session holding times, with $\overline{X} = 1$. The analytical formula is plotted with a solid line (+) and the simulation results are plotted with a dashed line ($\times$). The curves correspond to different values of p, with the larger mesh sizes corresponding to the lower probability of success curves. The curves obtained from the analytical results are plotted for λ ranging from 0.05 to 0.5, while the curves obtained from the simulation results are plotted for the values of λ actually obtained in the simulations.

by sessions originating at originating at other nodes and by sessions originating at other links of the same node. Our analysis will take into account the activity of the other queues in the network through the retrial attempts that a message must make before establishing a connection.

The holding time of session i is denoted by X_i and can be any random variable whose mean $\overline{X}$, and second and third moments $\overline{X^2}$ and $\overline{X^3}$ are known. A session i, arriving at an outgoing link L, joins the queue in the corresponding input-buffer $\mathcal{B}_L^s$. The delay incurred by the session consists of several components. Session i will first have to wait in the queue

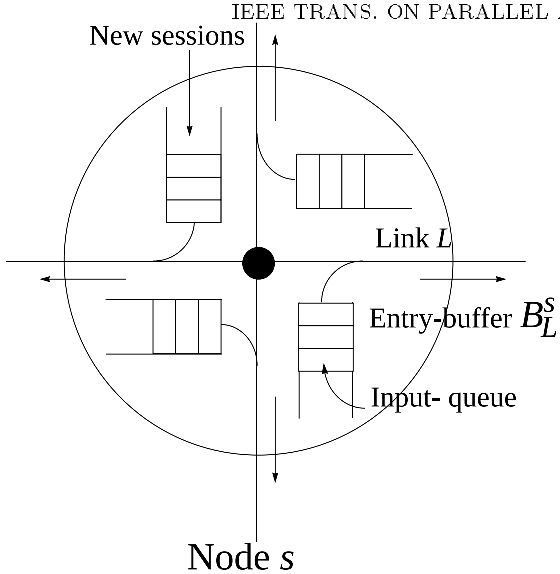

Fig. 3. This figure illustrates the node model assumed. Each node has $2d$ link input-buffers. New sessions generated at the node join the buffers $\mathcal{B}_L^s$ of their respective outgoing links L. Transit traffic through the node does not use the input-buffers.

for the time it takes for the the session currently at the head of the queue to finish, and depart from the system. This time is called *residual time*, and is denoted by R_i. If session i arrives at an empty queue, R_i is equal to zero. When session i reaches the head of the queue, it will have to wait for the time it takes until the connection to its destination is established, since more than one trials may be required to establish the connection. This time is called *connection delay*, and is denoted by C_i. The details of the connection phase will be described shortly. Session i will also have to wait for the time $\sum_{j=i-N_i}^{i-1} (X_j + C_j)$ it takes to serve the N_i sessions found in queue upon its arrival. The time between the arrival of the session at a node and the time it reaches the head of the link input-queue is called the *queueing delay* and is given by

$$Q_i = R_i + \sum_{j=i-N_i}^{i-1} (X_j + C_j) \,. \tag{15}$$

The total *waiting time* of session i at the queue includes its own connection delay, and is

$$W_i = Q_i + C_i. \tag{16}$$

The *total delay* of a session is defined as the time that elapses between the arrival of the session and the time it is completed. Therefore, the total delay includes both waiting time and holding time and is given by

$$T_i = X_i + W_i. \tag{17}$$

During the connection phase of a session currently at the head of the queue, a set-up packet is sent to the destination to establish the circuit connection. We will use the convention that if the new session is blocked due to other active sessions, currently using one or more links required by the session, its failure to establish a connection is "charged" randomly to one of these active sessions. The link queue then takes an *obligatory vacation* for a duration equal to the remaining holding time of the session to which the failure was charged. (Recall that session durations are recorded by the nodes. Thus, the node where a setup packet is blocked knows the remaining holding time of the session that blocks the setup packet, and sends this information back to the source via the setup packet. The "charging" of a failure randomly to one of the sessions using links required by the new session is then merely an analytical convenience.) Once this session is completed, the link queue is informed and it takes an *artificial vacation* of duration V, where V is an independent random variable with first and second moments equal to $\overline{V}$ and $\overline{V^2}$, respectively. The reason for introducing the artificial vacation is the following. In general, there can be multiple sessions whose failure to establish a connection has been charged to a given active session. When the active session is completed, the multiple sessions blocked by it contend for the links of the network in an attempt to establish their respective circuits. Sending these sessions on an artificial vacation randomizes the times at which these sessions retry, providing in this way an effective way to resolve conflicts. When the artificial vacation is over, a set-up packet is sent again to establish a circuit. If it is unsuccessful, the input-buffer queue repeats the cycle of taking an obligatory vacation followed by an artificial vacation until the circuit is finally established.

B. Queuing analysis

For the purpose of the subsequent analysis, we will make the following approximating assumption:

Approximating Assumption A1: When the input-buffer queue returns from an artificial vacation, it finds the network in steady-state.

We let P_h be the steady-state probability that a session at the *head* of a link input-buffer successfully establishes a circuit. An implication of assumption A1 is that the probability of successfully establishing a circuit during *any* trial (following an artificial vacation) is equal to P_h. In addition to that, assumption A1 implies that the connection times C_j of the sessions found in queue by a new session i are identically distributed for all j, and are independent of the number of sessions N_i found in the queue by session i.
In what follows, we derive expressions for the probability P_h, connection delay, residual time, and queueing delay at a particular link input-buffer.

B.1 Calculation of the steady-state probability P_h

The session that advances to the head of the link input-queue first takes an artificial vacation. When this session returns from the artificial vacation and makes its first attempt to establish a connection, the first link along the session's path is available not with probability q_0 as derived in Section 3 (for a random time instant), but with a somewhat larger probability. This is because this probability is now conditional on the fact that the first link along its path is not being used by a session starting at this node, which means that the first link is not in state 3. Thus, the probability that the first link along its path is available is now given by

$$\Pr(L \text{ available} \mid L \text{ not in state } 3) = \frac{q_0}{1 - q_3}.$$

Since all other probabilities remain the same as before, the probability of success for a session at the head of a queue becomes

$$P_h = \frac{P_{success}}{1 - q_3}, \tag{18}$$

where $P_{success}$ is given by Eqs. (13) and (14) for p even and p odd, respectively.

B.2 Calculation of the mean connection delay $\overline{C}$

We let k_i be the number of unsuccessful tries made by session i before successfully establishing the circuit. The connection delay is given by

$$C_i = \sum_{j=1}^{k_i} \left(X_j^{\,i,r} + V_j^i \right) + V_0^i, \tag{19}$$

where $X_j^{i,r}$ is the remaining holding time of the circuit due to which the j^{th} trial failed (that is, the duration of the obligatory vacation described in Subsection 4.1), V_j^i is the duration of the j^{th} artificial vacation, and V_0^i is the duration of the zeroth artificial vacation, which every session takes immediately after advancing to the head of the link input-queue. (We remind the reader that when the j^{th} trial fails due to its being blocked by more than one ongoing circuit, we randomly charge this failure to only one such circuit.)

Based on assumption A1, the probability that k_i unsuccessful tries are made before the circuit for session i is finally established is given by

$$\Pr(k_i \text{ retries before success}) = (1 - P_h)^{k_i} P_h. \tag{20}$$

Modulo our approximating assumption A1, the number of retries is independent of $X_j^{i,r}$, the remaining holding time of the ongoing circuit at the j^{th} retry, and is also independent of V_j^i, the duration of the j^{th} artificial vacation. Since at each trial a session trying to establish a connection waits for the session that blocked it to finish, it impinges on a different session at each trial, and the remaining holding times $X_j^{i,r}$ at the j^{th} retry are also i.i.d. Therefore, if $f_X(x)$ be the distribution of the session holding time X, $f_Y(y) = xf_X(x)/\overline{X}$ is the distribution as seen by an arriving setup packet, and the distribution of the residual time $X_j^{i,r}$ can be calculated to be (see, for instance, [Kle75] pp. 170-173) $f_{X_j^{i,r}}(R) = (1 - F_X(R))/\overline{X}$, where $F_X(R)$ is the cumulative distribution function of X.

Thus, we have

$$\begin{aligned}
E[C_i] &= E\Big[\sum_{j=1}^{k_i} \left(X_j^{i,r} + V_j^i\right) + V_0^i\Big] \\
&= E\Big[E\Big[\sum_{j=1}^{k_i} \left(X_j^{i,r} + V_j^i\right) \mid k_i\Big]\Big] + E[V_0^i] \\
&= E[k_i]E\Big[X^{i,r} + V^i\Big] + E[V^i].
\end{aligned}$$

Taking the limit as $i \to \infty$, and assuming that the system reaches steady-state, we obtain

$$\overline{C} = \overline{k}\Big(\frac{\overline{X^2}}{2\overline{X}} + \overline{V}\Big) + \overline{V}, \tag{21}$$

where $\overline{C} = \lim_{i\to\infty} E[C_i]$, $\overline{k} = \lim_{i\to\infty} E[k_i]$, $\overline{X^2}/2\overline{X} = \lim_{i\to\infty} E[X^{i,r}]$, and $\overline{V} = \lim_{i\to\infty} E[V^i]$ are the mean connection time, number of retries, residual waiting time, and artificial vacation time, respectively. The mean number of retries is calculated in Appendix A3 to be equal to

$$\overline{k} = \frac{1 - P_h}{P_h}.$$

Therefore, Eq. (21) can be rewritten as

$$\overline{C} = \frac{(1 - P_h)}{P_h} \frac{\overline{X^2}}{2\overline{X}} + \frac{\overline{V}}{P_h}. \tag{22}$$

B.3 Calculation of the mean queueing delay Q

Taking expectations in Eq. (15) we obtain

$$E[Q_i] = E[R_i] + E\Big[\sum_{j=i-N_i}^{i-1} (X_j + C_j)\Big]$$

The random variables N_i and $X_{i-1}, \ldots, X_{i-N_i}$ are clearly independent, since the number of sessions N_i found in queue by session i is independent of their holding times. The random variables $C_{i-1}, \ldots, C_{i-N_i}$ are also independent of N_i because of assumption A1. Therefore, we have

$$\begin{aligned} E[Q_i] &= E[R_i] + E\Big[\sum_{j=i-N_i}^{i-1} E[X_j + C_j \mid N_i]\Big], \\ &= E[R_i] + E[X + C]E[N_i], \\ &= E[R_i] + \left(\overline{X} + \overline{C}\right) E[N_i]. \end{aligned}$$

Taking the limit as $i \to \infty$ we get

$$Q = R + N_Q \left(\overline{X} + \overline{C}\right), \tag{23}$$

where $Q = \lim_{i\to\infty} E[Q_i]$ is the mean queueing delay, $N_Q = \lim_{i\to\infty} E[N_i]$ is the mean number of sessions in queue, and $R = \lim_{i\to\infty} E[R_i]$ is the mean residual time.

The arrival rate of connection requests at the input-buffer queue of a link is equal to $\frac{\lambda}{2d}$, where λ is the corresponding arrival rate per node. Using Little's Theorem, we have

$$N_Q = \frac{\lambda}{2d} Q. \tag{24}$$

Letting

$$\rho = \frac{\lambda}{2d}\left(\overline{X} + \overline{C}\right), \tag{25}$$

and substituting Eqs. (24) and (25) into Eq. (23), we obtain

$$Q = \frac{R}{1-\rho}. \tag{26}$$

The mean residual time R is calculated in Appendix A4, and works out to

$$R = \frac{\lambda}{4d}\overline{(X+C)^2}. \tag{27}$$

Substituting the expression for $\overline{C^2}$ (derived in Appendix A5) in the bracketed expression in Eq. (27) and simplifying, we obtain

$$\begin{aligned}\overline{(X+C)^2} &= \frac{\left(\overline{X^2}+\overline{V^2}+2\overline{X}\,\overline{V}\right)}{P_h} + 2\frac{(1-P_h)}{P_h}\overline{V}\left(\overline{V}+\frac{\overline{X^2}}{\overline{X}}\right) \\ &\quad + \frac{(1-P_h)^2}{P_h^2}\left(\frac{\overline{X^2}}{2\overline{X}}\right)^2 + \frac{(1-P_h)}{P_h}\left(\frac{\overline{X^3}}{3\overline{X}}\right). \end{aligned} \tag{28}$$

Substituting Eqs. (28) and (27) in Eq. (26), we get that the mean queueing delay is given by

$$\begin{aligned} Q &= \frac{\lambda\left(\overline{X^2}+\overline{V^2}+2\overline{X}\,\overline{V}\right)}{4d(1-\rho)P_h} + 2\frac{\lambda(1-P_h)}{4d(1-\rho)P_h}\overline{V}\left(\overline{V}+\frac{\overline{X^2}}{\overline{X}}\right) \\ &\quad + \frac{\lambda(1-P_h)^2}{4d(1-\rho)P_h^2}\left(\frac{\overline{X^2}}{2\overline{X}}\right)^2 + \frac{\lambda(1-P_h)}{4d(1-\rho)P_h}\left(\frac{\overline{X^3}}{3\overline{X}}\right). \end{aligned} \tag{29}$$

The average waiting time of a session W is given as $W = Q + \overline{C}$, where $\overline{C}$ is given by Eq. (22). Finally, the average total delay $T = W + \overline{X}$ is given by

$$\begin{aligned} T &= \frac{\lambda\left(\overline{X^2}+\overline{V^2}+2\overline{X}\,\overline{V}\right)}{4d(1-\rho)P_h} + 2\frac{\lambda(1-P_h)}{4d(1-\rho)P_h}\overline{V}\left(\overline{V}+\frac{\overline{X^2}}{\overline{X}}\right) \\ &\quad + \frac{\lambda(1-P_h)^2}{4d(1-\rho)P_h^2}\left(\frac{\overline{X^2}}{2\overline{X}}\right)^2 + \frac{\lambda(1-P_h)}{4d(1-\rho)P_h}\left(\frac{\overline{X^3}}{3\overline{X}}\right) \\ &\quad + \frac{1}{P_h}\left(\overline{V}+\frac{\overline{X^2}}{\overline{X}}\right) + \left(\frac{2\overline{X}^2-\overline{X^2}}{2\overline{X}}\right) \end{aligned} \tag{30}$$

We simulated both the analytical model and a model of the physical system in which we accounted for the overhead incurred by the setup packets during the reservation phase. The simulation of the analytical model (where the setup phase is instantaneous) is necessary in order to confirm the quality of the approximating assumption A1. Figure 4 illustrates analytical and simulation results for the probability of success P_h at the head of a queue and the delay parameters Q and C, versus the arrival rate λ per node, for $d = 2$ and

$p = 9$, when the setup phase is instantaneous (simulation of the analytical model), and the session holding times X and artificial vacation durations V are both exponentially distributed with parameters $\overline{X} = 1.0$ and $\overline{V} = 0.5$, respectively. As is evident from Fig. 4, the agreement between the analysis and simulation results is very good.

In the simulations of the physical system model, we also take into account the effect of the overhead due to reservations, which has been assumed to be negligible in our analysis (where the setup phase was assumed to be instantaneous). In the physical system, each time that a session attempts to make a reservation, a setup packet travels through the network, one hop at a time, to reserve the links on the session's path, thus interfering with other set-up packets and with previously established circuits. If the setup packet is blocked enroute, it retraces its path tearing down a partially established circuit (also in steps), freeing the reserved links, and bringing back with it information on when the session should try next. If the setup packet makes it successfully to the destination, it returns to the source with a confirmed reservation of the session's path, at which time the session begins transmission. Therefore, if h be the number of hops on a session's path and Δ the time taken by the setup packet to advance one hop (as a fraction of the average session holding time $\overline{X}$), the average setup overhead for a successful connection is equal to $2\Delta\overline{h}$, where $\overline{h}$ is the mean internodal distance, as specified in Section 2.

Figure 5 shows the simulation results for the queueing delay Q and the connection delay C, for $d = 2$ and $p = 9$, when the setup overhead is equal to zero (presented as a reference), and for average setup overhead equal to 1% and 2% of the average session holding time. We note that a setup overhead of 2% represents a relatively high overhead in a circuit switching context. As is evident from the simulations, the values of the delay parameters obtained via simulations are close to the values for 0% overhead when the setup delay is a small percentage, 2% or less, of the session holding time (as would be the case for a circuit switched network).

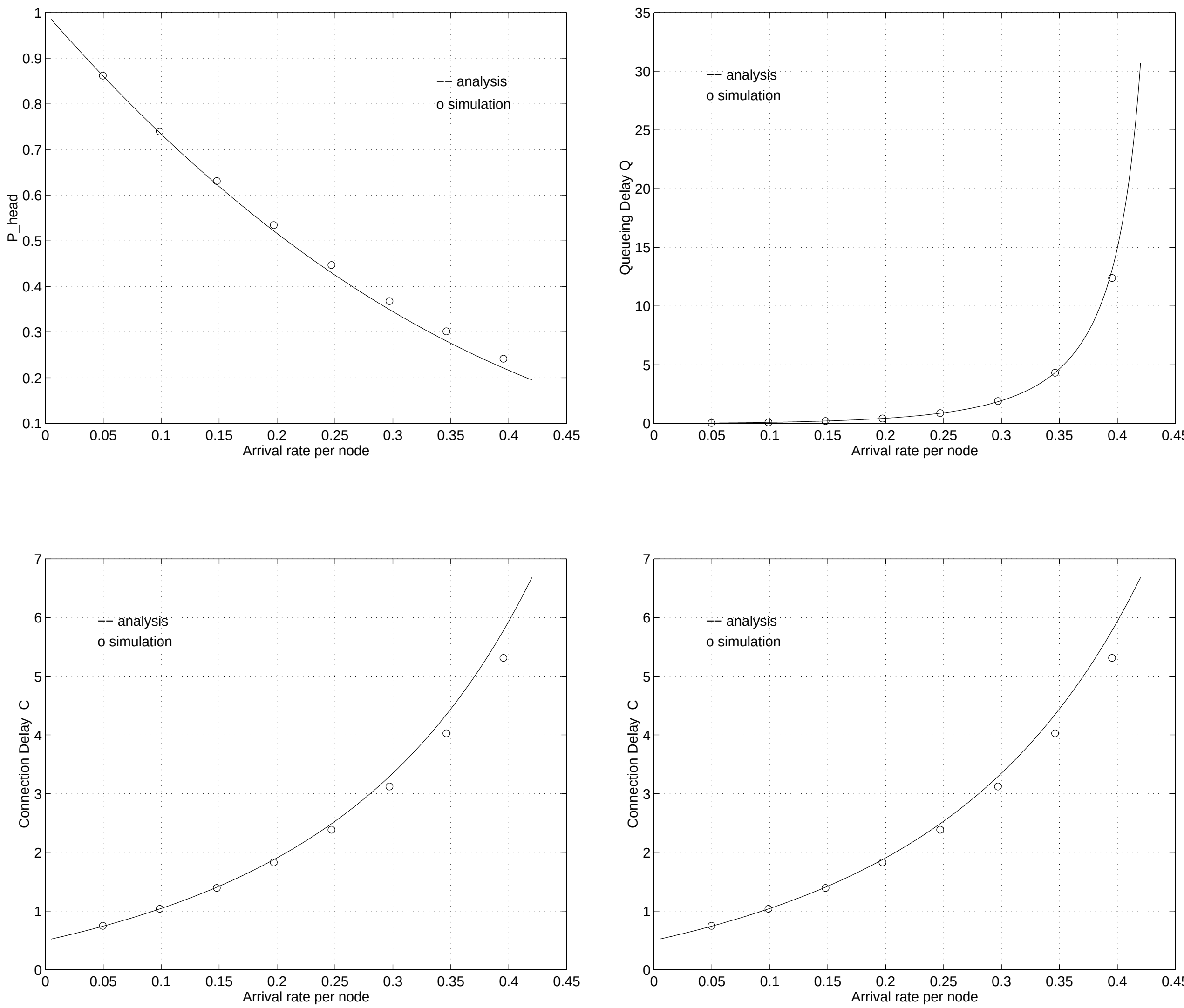

Fig. 4. The probability of success P_h at the head of the queue and the various delay parameters versus the arrival rate per node λ, for $d = 2$ and $p = 9$. The plots show the analytically predicted values of the queueing delay Q, the residual service time R, and the connection delay C, and the corresponding values obtained through simulations when the setup overhead is equal to zero. All calculations and simulations have been done assuming that session holding times and artificial vacations are exponentially distributed with means $\overline{X} = 1$ and $\overline{V} = 0.5$, respectively.

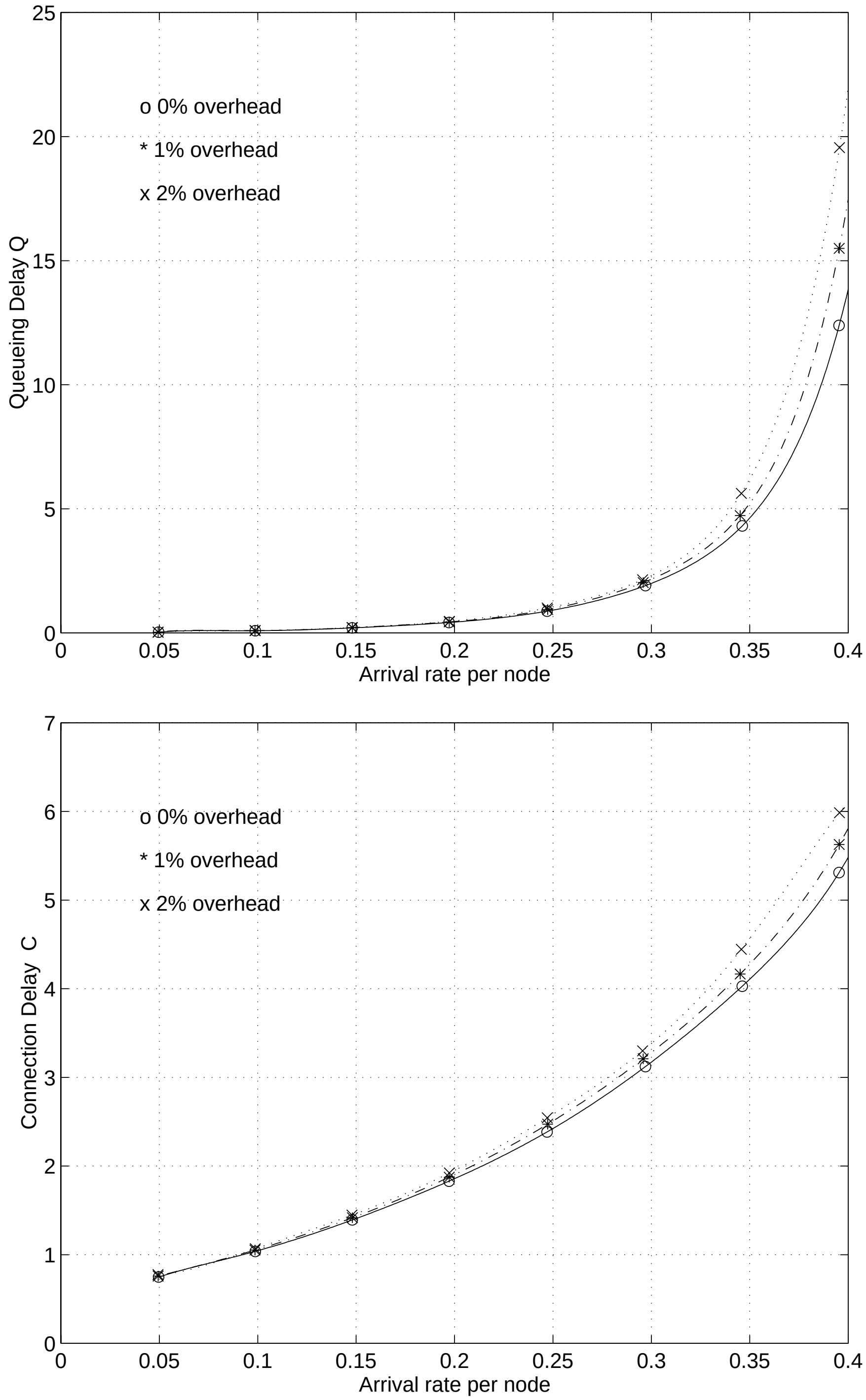

Fig. 5. Simulation results for the queueing delay Q and the connection delay C for $d = 2$ and $p = 9$, when the setup overhead is accounted for and is equal to 0%, 1% and 2% of the mean session holding time. All simulations have been done assuming that session holding times and artificial vacations are exponentially distributed with means $\overline{X} = 1$ and $\overline{V} = 0.5$, respectively.

B.4 Region of stability of the link input-queue

A necessary (but not sufficient) condition for stability is obtained by noting that $1-q_0 < 1$, which gives

$$\lambda < \frac{8}{\overline{X}} \frac{(p^d - 1)}{p^{d-1}(p^2 - 1)}, \quad \text{for } p \text{ odd}, \tag{31a}$$

and

$$\lambda < \frac{8}{\overline{X}} \frac{(p^d - 1)}{p^{d+1}}, \quad \text{for } p \text{ even}. \tag{31b}$$

This condition is necessary for the stability for any circuit switched scheme, independently of the routing scheme and the other details of its implementation. For our link input-buffer queueing system to be stable, the following stronger condition has to hold:

$$\rho = \frac{\lambda}{2d}(\overline{X} + \overline{C}) < 1,$$

or

$$\lambda < \frac{2d}{\overline{X} + \overline{C}}. \tag{32}$$

Equation (32) imposes a stronger condition on λ and provides a condition that is both necessary and sufficient for our scheme to be stable. Substituting for $\overline{C}$ from Eq. (22) (with $\overline{X^r}$ replaced by $\overline{X}$), we get after some algebraic manipulation that

$$\lambda < \frac{2dP_h(\lambda, \overline{X})}{\overline{X} + \overline{V}}, \tag{33}$$

where the dependence of P_h on λ and $\overline{X}$ has been made explicit.

Although, Eq. (33) cannot be solved explicitly for λ, the stability region is easily obtained graphically, by plotting $P_h(\lambda, \overline{X})$ and $\frac{\lambda}{2d}(\overline{X} + \overline{V})$ on the same graph and identifying the region where Eq. (33) holds. The calculation of the stability region is illustrated in Fig. 6. We note that although the retrials made by a session to establish a connection are reminiscent of retrails in an ALOHA system, bistability problems do not arise in our analysis, because there is no penalty for trying when the setup phase is instantaneous.

V. Asymptotic results for large meshes

In this section we obtain asymptotic expressions for the probability of success $P_{success}$ and the various queueing delay parameters of interest, when the size of the mesh tends

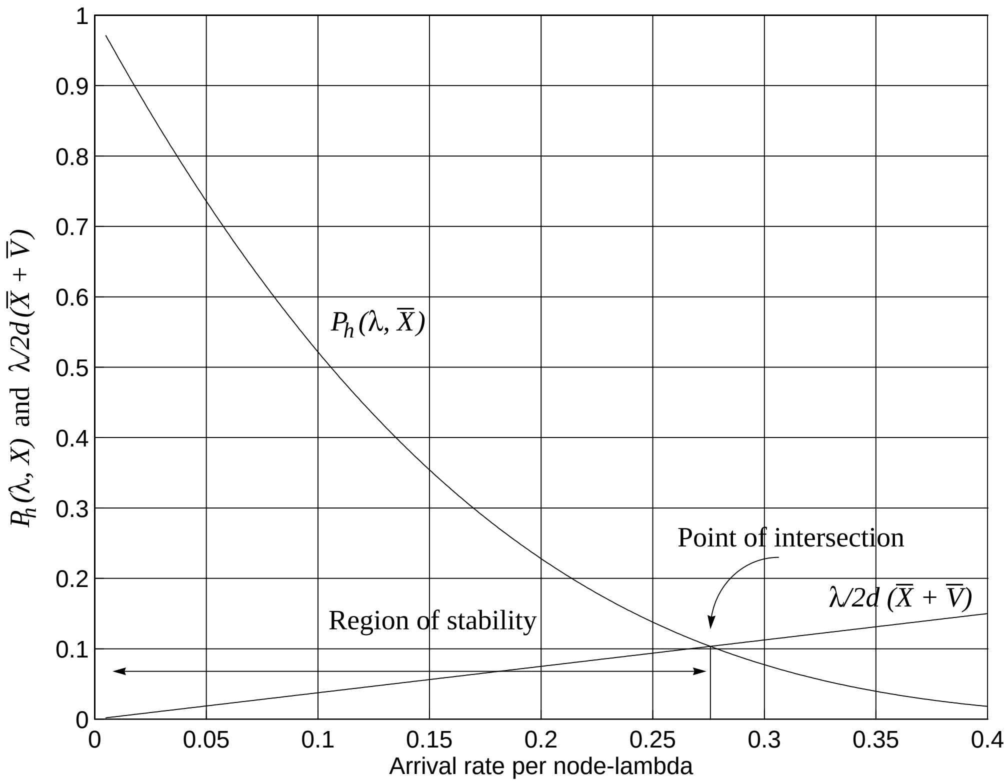

Fig. 6. Illustrates the graphical calculation of the stability region of a link input-queue. The curve for $P_{success}(\lambda, \overline{X})$ and the straight line $\frac{\lambda}{2d}(\overline{X}+\overline{V})$ are both plotted on the same graph. The stability region is obtained by simply identifying the region where Eq. (33) holds. In the above figure, the marked region (the range of λ values from 0 upto the point of intersection of the two curves) represents the region of stability, that is, those values of λ where the link queue is stable. In the above example, $p = 15$ with $\overline{X} = 1.0$ and $\overline{V} = 0.5$, respectively.

to infinity. In particular, if we keep the probability q_0 that a link is idle constant, while allowing the mesh size p to tend to infinity, we find (see Appendix A6) that

$$P_{success}(\infty) = q_0^{2d}\frac{(1 - e^{-2\frac{(1-q_0)}{q_0}})^d}{2^d(1-q_0)^d}. \tag{34}$$

Note that $P_{success}(\infty)$ decreases exponentially with the mesh dimension d. The asymptotic values of all queueing delay parameters can now be obtained simply by replacing P_h with $P_{success}(\infty)$ in the all expressions derived in Section 4, since $P_h = \frac{P_{success}}{1-q_3}$, and $q_3 \to 0$ as $p \to \infty$. For a given value of q_0 the corresponding value of λ is obtained from the expression for q_0 derived in Section 2 (Eq. (9)). Figure 8 shows the plot of $P_{success}(\infty)$ versus the link utilization $1 - q_0$, for various dimensions d ranging from 2 to 6.

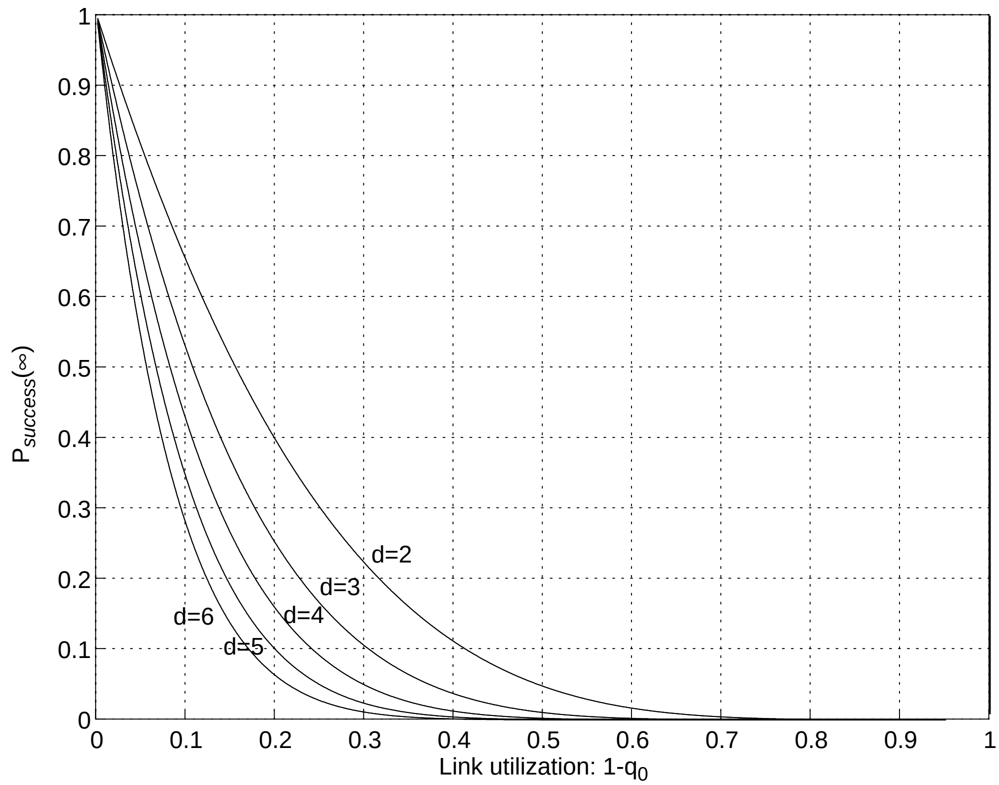

Fig. 7. Illustrates how the asymptotic probability of success $P_{success}(\infty)$ varies with link utilization $1-q_0$, for various dimensions d. The plots are obtained for d varying from 2 to 6, as shown above. Note that the same link utilization does *not* correspond to the same traffic arrival rate λ here. In fact, for the same value of the utilization $1-q_0$, the arrival rate λ will be smaller for a mesh of higher dimension d.

VI. Circuit switching in the hypercube

Beginning with this section, we turn our attention to circuit switching in the hypercube network. Chlamtac, Ganz, and Kienzle [CGK91], have analyzed circuit switching in a hypercube network of processors with the assumptions that each node has only a single transmission buffer and that arrivals of new sessions only occur when the input buffer is empty. New sessions are discarded if there is already a session waiting to be served. The analysis in [CGK91] does not yield closed form expressions for the delay parameters, and requires a fixed point iteration to obtain the steady-state probabilities of the Markov model used.

In our model, each hypercube node is a cross-bar switch and has d link entry-buffers, one for each outgoing link. The routing tag of a session is a d-bit string, which is equal to

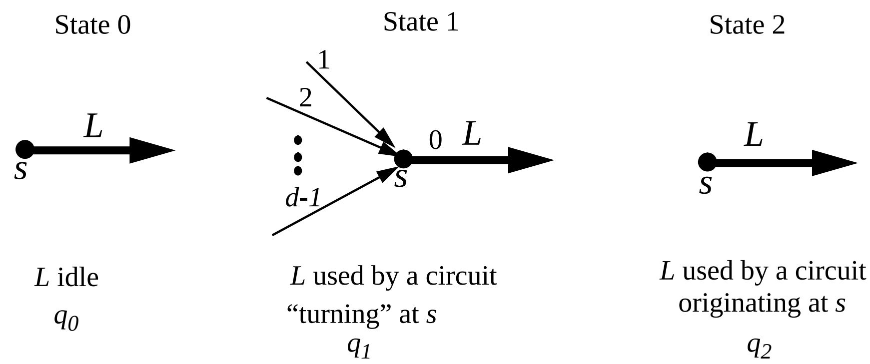

Fig. 8. The three possible states of a network link L in a hypercube with *cross-bar* switches are illustrated. State 0 corresponds to the case where L is idle. State 1 corresponds to the case where L is used by a continuing circuit at node s. Note that State 1 is composed of $d-1$ substates, depending on the dimension of the link through which the circuit makes a turn into node s. State 2 corresponds to the case where L is used by a circuit originating at node s.

$s \oplus t$, where s is the source node, t is the destination node, and $\oplus$ is the bitwise exclusive OR operation. We assume that the setup packet of a session chooses a random order in which the bits of its routing tag are corrected and that it insists on that order as it progresses from its source to its destination (therefore, the routing scheme is non-adaptive or oblivious). The link input-buffer at which an arriving session is placed depends on the order in which the bits of its routing tag will be corrected.

Any network link L can be at any time in one of three states illustrated in Fig. 9. As before, we denote by q_i, $i = 0, 1, 2$, the steady-state probability that link L is in state i. Clearly, we have

$$q_0 + q_1 + q_2 = 1. \tag{35}$$

When an outgoing link at a node s is used by a continuing circuit, there are $d-1$ other input links from which the circuit may have come (see Fig. 8). As a result State 2 is composed of $d-1$ substates, depending on the dimension of the incoming link from which the circuit turns into link L.

When a circuit with routing tag $(t_{d-1}t_{d-2}\ldots t_0) \neq (00\ldots0)$ that has k ones and $d-k$ zeros is established, it will contain one link in state 2 and $k-1$ links in state 1. The steady-state probability q_1 that a link is in state 1 is given by

$$q_1 = K\Big[\sum_{k=1}^{d}(k-1)\binom{d}{k}\Big] = K[(d-2)2^{d-1}+1], \tag{36}$$

where K is some constant of proportionality to be derived shortly.
Similarly, the probability q_2 that a given link is in state 2 is given by

$$q_2 = K(2^d - 1) = K(N-1). \tag{37}$$

Solving Eqs. (35)–(37) with respect to K, we obtain

$$K = \frac{1-q_0}{d2^{d-1}}. \tag{38}$$

Using Little's theorem, we get, after some algebraic manipulation, that

$$1 - q_0 = \frac{\lambda 2^{d-1}\overline{X}}{2^d - 1}, \tag{39}$$

where λ is the session arrival-rate per node, and $\overline{X}$ is the mean session holding time. Substituting Eqs. (38) and (39) in the expression for q_1 and q_2 we get

$$q_1 = \frac{\lambda\overline{X}}{d(2^d-1)}[(d-2)2^{d-1}+1], \tag{40}$$

and

$$q_2 = \frac{\lambda\overline{X}}{d}. \tag{41}$$

To calculate the probability of success, we proceed as in Section 3. The first link of the path is available with probability q_0. At each subsequent step the probability that the $i+1^{\text{th}}$ link, L, on the path is available given that the i^{th} link ($i > 0$), $L-1$, was available is equal to

$$\begin{aligned} Pr & \quad (L \text{ available} \mid L \text{ not in a given substate of state 1}) \\ = & \quad \frac{q_0}{1-\frac{q_1}{d-1}} \stackrel{\text{def}}{=} \alpha.(42) \end{aligned}$$

To see this, note that state 1 is composed of $d-1$ substates depending on the direction from which the path makes a turn at node s into link L (see Fig. 8). Since each of the substates is equally likely, the probability of the link being in a given substate is equal to $q_1/(d-1)$.

The (conditional) probability of successfully establishing a connection is therefore given by

$$P_{success}(t_{d-1}t_{d-2}\ldots t_0) = q_0\,\alpha^{k-1},$$

and the probability of successfully establishing a connection averaged over all routing tags (equivalently, over all source-destination pairs) is given by

$$\begin{aligned} P_{success} &= \frac{1}{2^d-1}\sum_{t_{d-1}t_{d-2}\ldots t_0} P_{success}(t_{d-1}t_{d-2}\ldots t_0) \\ &= \frac{q_0}{\alpha(2^d-1)}\Big[(1+\alpha)^d - 1\Big], (43) \end{aligned}$$

where α is given by Eq. (42). Figure 9 illustrates the analytical and simulation results obtained for the probability of success as a function of the arrival rate λ for several hypercube networks, with dimensions ranging from $d = 3$ to $d = 9$.

As in Subsection 4.2.1, the steady-state probability P_h that a session at the head of a link input-buffer successfully establishes a connection is now given by

$$P_h = \frac{P_{success}}{1-q_2}.$$

Consequently, the connection delay $\overline{C}$, the queueing delay Q, and the total delay T, can now be obtained directly from Eqs. (22), (29), and (30), simply by using the appropriate value of P_h as calculated above, and by noting that the arrival rate of connection requests at the input-buffer queue of a hypercube link is equal to $\frac{\lambda}{d}$. Figure 10 illustrates the probability of success at the head of the queue P_h and the various delay parameters for a hypercube of dimension $d = 8$, when the setup phase is instantaneous. As explained in Section 4, for the hypercube network also we performed simulations both for the analytical model and for a model of a physical system, where the reservation overhead incurred due to the setup packets is explicitly accounted for. Figure 11 illustrates the simulation results obtained for the queueing delay Q and the connection delay C for $d = 8$, when the average setup overhead is equal to 0%, 1% and 2% of the average session holding time.

VII. Conclusions

We have analyzed circuit-switching in a d-dimensional wraparound mesh, which is an important interconnection topology for multiprocessor computers. Our analysis uses few

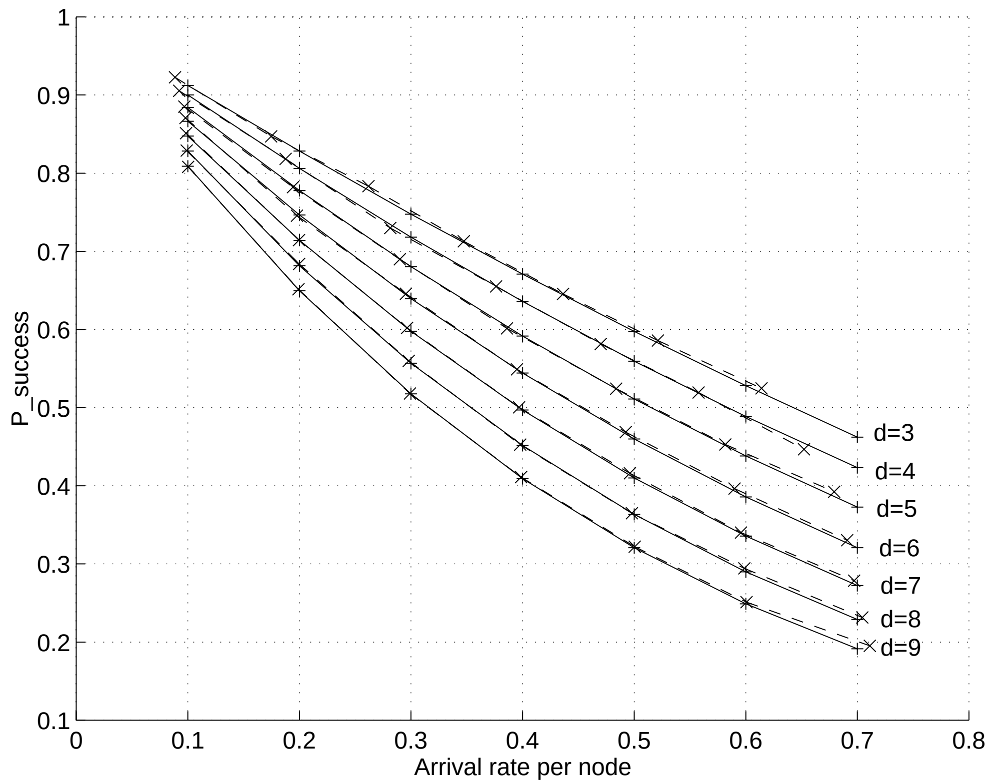

Fig. 9. Analytical and simulation results obtained for the probability of success as a function of the arrival rate λ of sessions, for hypercubes of dimensions ranging from $d = 3$ to $d = 9$. All curves correspond to exponentially distributed session holding times, with $\overline{X} = 1$. The analytical formula is plotted with a solid line (+) and the simulation results are plotted with a dashed line ($\times$). The curves obtained from the analytical results are plotted for λ ranging from 0.1 to 0.7, while the curves obtained from the simulation results are plotted for the values of λ actually obtained in the simulations.

approximating assumptions, and resulted in closed form expressions for most queueing parameters of interest. Since the queueing analysis depends on the network topology only through the probability of success $P_{success}$, we showed how it can be extended to other regular topologies such as the 2^d-node hypercube. Although analytic models cannot always reflect all the details of a real system, they provide an important framework for predicting network performance. A possible extension of this work would be to examine the case when each network link has capacity k and can carry multiple (up to k) circuits. In that case we expect the throughput to improve by more than a factor of k. A further direction of research would be to consider multiple virtual channels per link, where the effective bandwidth of the session, rather than the session rate, could be the parameter

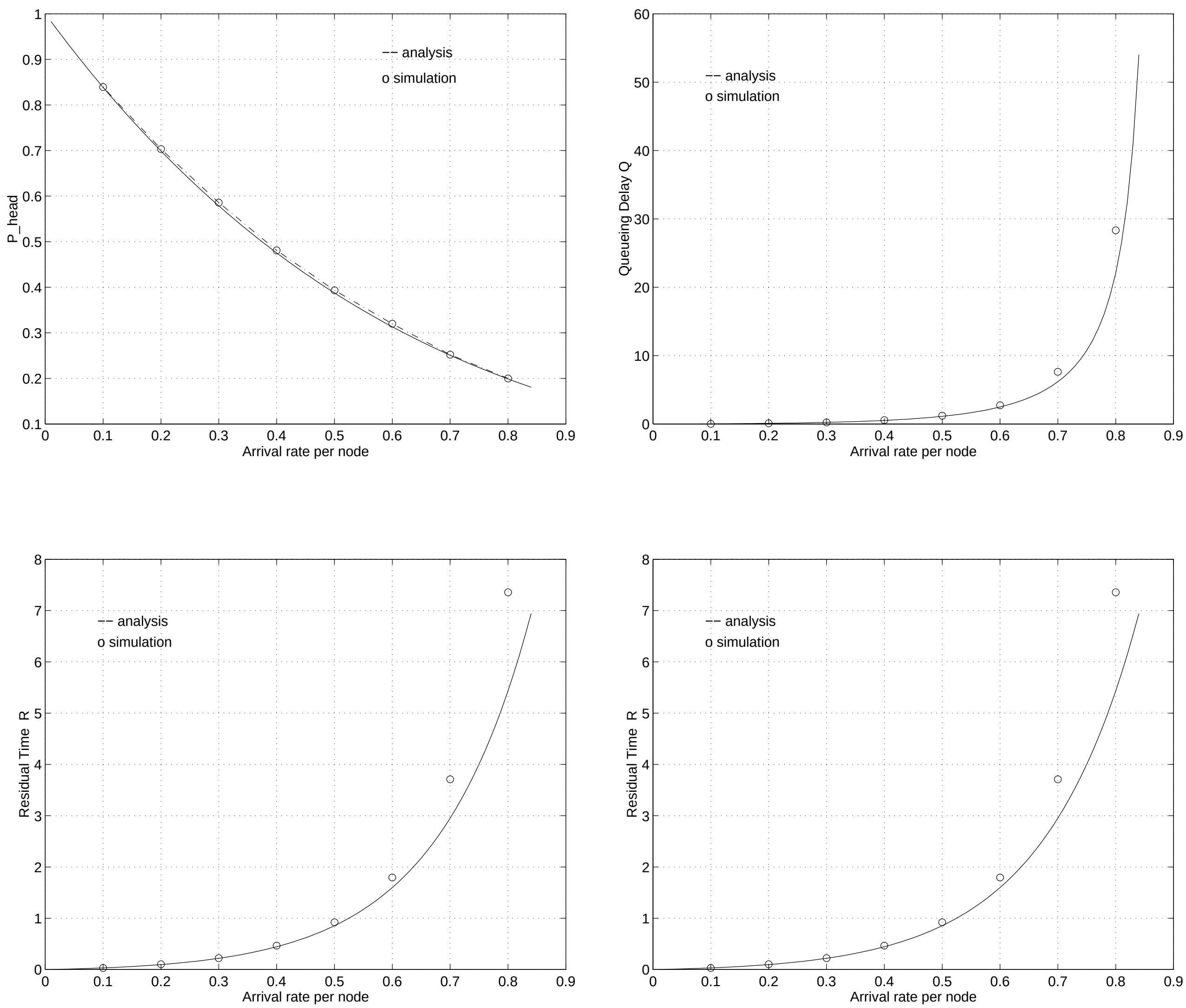

Fig. 10. The probability of success P_h at the head of the queue and the various delay parameters versus the arrival rate per node λ, for a hypercube of dimension $d = 8$. The plots show the analytically predicted values of the three components of session delay: the queueing delay Q, the connection delay C, and the residual service time R, and the corresponding values obtained through simulations when the setup overhead is equal to zero. All calculations and simulations have been done assuming that session holding times and artificial vacations are exponentially distributed with means $\overline{X} = 1$ and $\overline{V} = 0.5$, respectively.

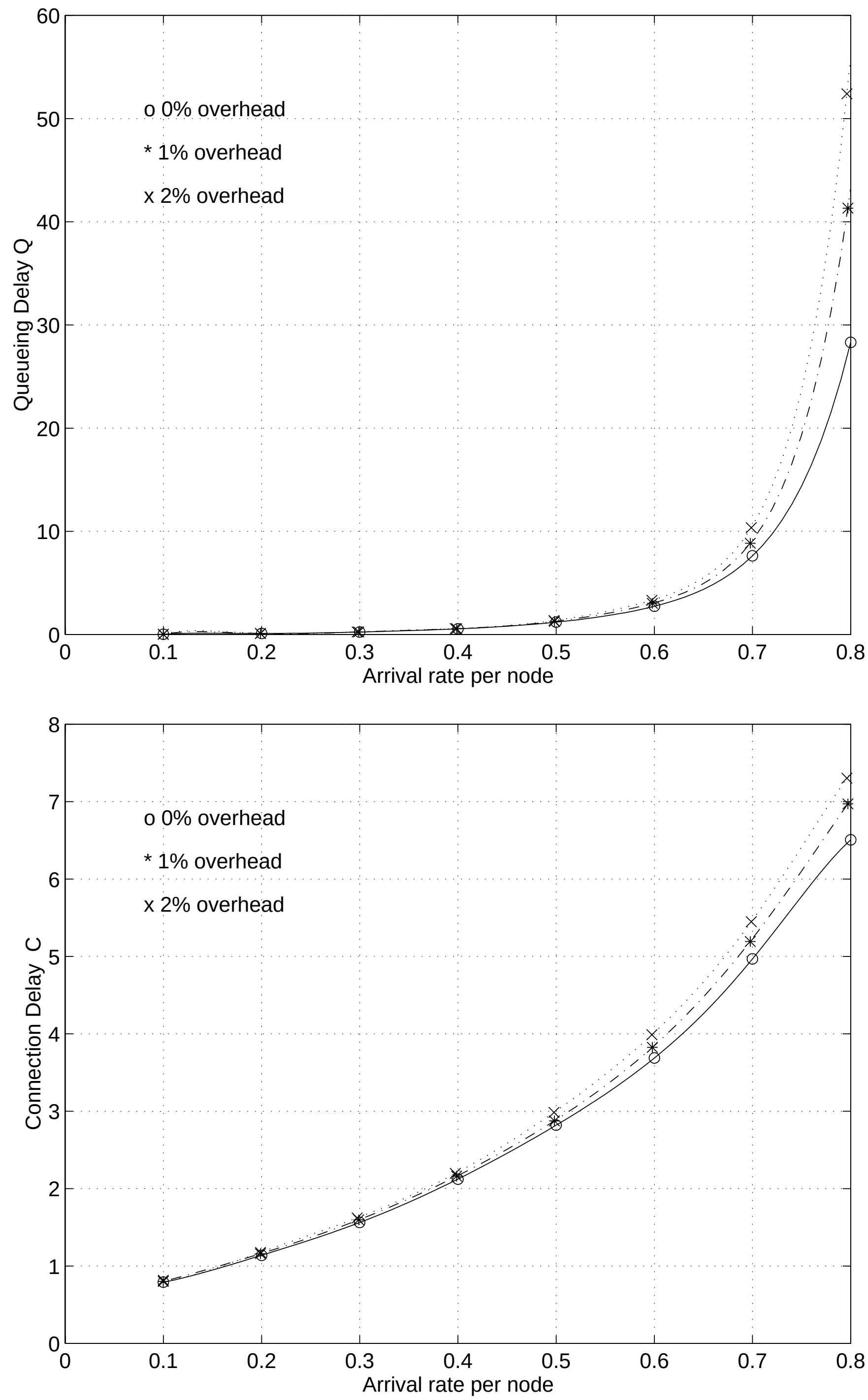

Fig. 11. Simulation results for the queueing delay Q and the connection delay C for a hypercube of dimension $d = 8$, when the setup overhead is accounted for and is equal to 0%, 1% and 2% of the mean session holding time. All simulations have been done assuming that session holding times and artificial vacations are exponentially distributed with means $\overline{X} = 1$ and $\overline{V} = 0.5$, respectively.

used. Finally, our analysis was done for oblivious routing, an analysis of adaptive routing methods would be a potentially interesting area for research.

VIII. Appendix

This section contains derivations of some results that were only stated in the main body of the paper.

A. *Derivation of the probabilities* $q_i, i = 1, 2, 3$

In this appendix we evaluate the probabilities $q_i, i = 1, 2, 3$. We treat separately the cases p odd and p even.

(a) Case p odd :

Expanding the summations in Eq. (4a) over the t_k's we get

$$\begin{aligned} q_1 &= K\Big[\sum_{\substack{t_d=-D \\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_1=-D \\ t_1\neq 0}}^{D} (d-1) + \binom{d}{1} \sum_{\substack{t_d=-D \\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_2=-D \\ t_2\neq 0}}^{D} (d-2) \\ &\qquad + \cdots + \binom{d}{d-2} \sum_{\substack{t_d=-D \\ t_d\neq 0}}^{D} \sum_{\substack{t_{d-1}=-D \\ t_{d-1}\neq 0}}^{D} 1\Big] \\ &= K\Big[2^d \sum_{t_d=1}^{D} \cdots \sum_{t_1=1}^{D} (d-1) + \binom{d}{1} 2^{d-1} \sum_{t_d=1}^{D} \cdots \sum_{t_2=1}^{D} (d-2) \\ &\qquad + \cdots + \binom{d}{d-2} 2^2 \sum_{t_d=1}^{D} \sum_{t_{d-1}=1}^{D} 1\Big] \\ &= K[(d-1)(2D)^d + \binom{d}{1}(2D)^{d-1} + \cdots + \binom{d}{d-2}(2D)^2] \\ &= K[d(2D)(2D+1)^{d-1} - (2D+1) + 1] \\ &= K[d(p-1)p^{d-1} - (p)^d + 1], (A.1) \end{aligned}$$

where we used the fact the $D = \lfloor \frac{p}{2} \rfloor$.

Writing the summation in Eq. (5a) explicitly we obtain

$$\begin{aligned} q_2 &= K\Big[\sum_{\substack{t_d=-D \\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_1=-D \\ t_1\neq 0}}^{D} (|t_d| + |t_{d-1}| + \cdots + |t_1| - d) \\ &\quad + \binom{d}{1} \sum_{\substack{t_d=-D \\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_2=-D \\ t_2\neq 0}}^{D} (|t_d| + |t_{d-1}| + \cdots + |t_2| - (d-1)) \end{aligned}$$

$$
\begin{aligned}
& + \cdots + \binom{d}{d-1} \sum_{\substack{t_d=-D \\ t_d \neq 0}}^{D} (t_d - 1) \Big] \\
= \quad & K \Big[2^d \sum_{t_d=1}^{D} \cdots \sum_{t_1=1}^{D} (t_d + t_{d-1} + \cdots + t_1 - d) \\
& + \binom{d}{1} 2^{d-1} \sum_{t_d=1}^{D} \cdots \sum_{t_2=1}^{D} (t_d + t_{d-1} + \cdots + t_2 - (d-1)) \\
& + \cdots + \binom{d}{d-1} 2 \sum_{t_d=1}^{D} 1 \Big] . (A.2)
\end{aligned}
$$

After some algebra, Eq. (A.2) reduces to

$$
\begin{aligned}
q_2 \quad = \quad & K \Big(\frac{D-1}{2} \Big) \\
\times \quad & \Big[d(2D)^d + (d-1) \binom{d}{1} (2D)^{d-1} + \cdots + \binom{d}{d-1} (2D) \Big] \\
= \quad & K d D (D-1)(2D+1)^{d-1} \\
= \quad & K p^{d-1} \frac{(p-1)(p-3)}{4} . (A.3)
\end{aligned}
$$

(b) Case p even :

Expanding the summation in Eq. (4a) over the t_k's, we get

$$
\begin{aligned}
q_1 \quad = \quad & K \Big[\sum_{\substack{t_d=-(D-1) \\ t_d \neq 0}}^{D} \cdots \sum_{\substack{t_1=-(D-1) \\ t_1 \neq 0}}^{D} (d-1) + \\
& \binom{d}{1} \sum_{\substack{t_d=-(D-1) \\ t_d \neq 0}}^{D} \cdots \sum_{\substack{t_2=-(D-1) \\ t_d \neq 0}}^{D} (d-2) \\
& + \cdots + \binom{d}{d-2} \sum_{\substack{t_d=-(D-1) \\ t_d \neq 0}}^{D} \sum_{\substack{t_{d-1}=-(D-1) \\ t_{d-1} \neq 0}}^{D} 1 \Big] . (A.4)
\end{aligned}
$$

Noting that

$$
\sum_{\substack{t_k=-(D-1) \\ t_k \neq 0}}^{D} 1 = 2D - 1 = p - 1,
$$

we can simplify Eq. (A.4) as

$$
\begin{aligned}
= \quad & K \Big[(d-1)(2D-1)^d + \binom{d}{1} (d-2)(2D-1)^{d-1} \\
& + \cdots + \binom{d}{d-2} (2D-1)^2 \Big]
\end{aligned}
$$

$$
\begin{aligned}
&= K[d(2D-1)(2D)^{d-1} - (2D)^d + 1] \\
&= K[d(p-1)p^{d-1} - p^d + 1], (A.5)
\end{aligned}
$$

where we used the fact that $D = \lfloor \frac{p}{2} \rfloor$.

Writing the summation in Eq. (5a) explicitly we obtain

$$
\begin{aligned}
q_2 \;=\; K\Big[& \sum_{\substack{t_d=-(D-1)\\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_1=-(D-1)\\ t_1\neq 0}}^{D} (|t_d| + |t_{d-1}| + \cdots + |t_1| - d) \\
& + \binom{d}{1} \sum_{\substack{t_d=-(D-1)\\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_2=-(D-1)\\ t_2\neq 0}}^{D} (|t_d| + |t_{d-1}| + \cdots + |t_2| - (d-1)) \\
& + \cdots + \binom{d}{d-1} \sum_{\substack{t_d=-(D-1)\\ t_d\neq 0}}^{D} (|t_d| - 1)\Big].
\end{aligned}
$$

Since

$$
\sum_{\substack{t_k=-(D-1)\\ t_k\neq 0}}^{D} (|t_k| - 1) = (D-1)^2,
$$

the expression for q_2 can be simplified to

$$
\begin{aligned}
q_2 \;&=\; K\Big[d(2D-1)(D-1)^2 \\
&\quad + \binom{d}{1}(d-1)(2D-1)^{d-2}(D-1)^2 \\
&\quad + \cdots + \binom{d}{d-1}(D-1)^2 \Big] \\
&= K(D-1)^2[d(2D)^{d-1}] = Kdp^{d-1}\frac{(p-2)^2}{4}.(A.6)
\end{aligned}
$$

B. Derivation of $P_{success}$

To derive the probability of success for p odd, we begin by writing Eq. (12) in the following expanded form

$$
\begin{aligned}
P_{success} \;=\; & \frac{q_0}{(p^d - 1)} \\
& \times \Big[\sum_{\substack{t_d=-D\\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_1=-D\\ t_1\neq 0}}^{D} \beta^{d-1} \alpha^{|t_d|+|t_{d-1}|+\cdots+|t_1|-d} \\
& + \binom{d}{1} \sum_{\substack{t_d=-D\\ t_d\neq 0}}^{D} \cdots \sum_{\substack{t_2=-D\\ t_2\neq 0}}^{D} \beta^{d-2} \alpha^{|t_d|+|t_{d-1}|+\cdots+|t_2|-(d-1)}
\end{aligned}
$$

$$+\cdots+\binom{d}{d-1}\sum_{\substack{t_d=-D\\t_d\neq 0}}^{D}\alpha^{|t_d|-1}\Big],$$

which simplifies to

$$\begin{aligned}P_{success} &= \frac{q_0}{(p^d-1)}\times \\ &\Big[2^d\beta^{d-1}\sum_{t_d=1}^{D}\cdots\sum_{t_1=1}^{D}\alpha^{t_d+t_{d-1}+\cdots+t_1-d} \\ &+\binom{d}{1}2^{d-1}\beta^{d-2}\sum_{t_d=1}^{D}\cdots\sum_{t_2=1}^{D}\alpha^{t_d+t_{d-1}+\cdots+t_2-(d-1)} \\ &+\cdots+\binom{d}{d-1}2\sum_{t_d=1}^{D}\alpha^{t_d-1}\Big].(B.1)\end{aligned}$$

Noting that

$$\sum_{t_k=1}^{D}\alpha^{t_k-1}=\frac{1-\alpha^D}{1-\alpha},$$

after some simplification, Eq. (B.1) gives

$$\begin{aligned}P_{success} &= \frac{q_0}{(p^d-1)}\frac{1}{\beta} \\ &\times\Big[\left(2\beta\left(\frac{1-\alpha^D}{1-\alpha}\right)\right)^d+\binom{d}{1}\left(2\beta\left(\frac{1-\alpha^D}{1-\alpha}\right)\right)^{d-1} \\ &+\cdots+\binom{d}{d-1}\left(2\beta\left(\frac{1-\alpha^D}{1-\alpha}\right)\right)\Big],\end{aligned}$$

which using the binomial expansion formula, can be rewritten to yield Eq. (13).

For the case p even, we again start with Eq. (12) and rewrite it in its expanded form to get

$$\begin{aligned}P_{success} &= \frac{q_0}{(p^d-1)}\Big[\sum_{\substack{t_d=-(D-1)\\t_d\neq 0}}^{D}\cdots\sum_{\substack{t_1=-(D-1)\\t_1\neq 0}}^{D}\beta^{d-1}\alpha^{|t_d|+|t_{d-1}|+\cdots+|t_1|-d} \\ &+\binom{d}{1}\sum_{\substack{t_d=-(D-1)\\t_d\neq 0}}^{D}\cdots\sum_{\substack{t_2=-(D-1)\\t_2\neq 0}}^{D}\beta^{d-2}\alpha^{|t_d|+|t_{d-1}|+\cdots+|t_2|-(d-1)} \\ &+\cdots+\binom{d}{d-1}\sum_{\substack{t_d=-(D-1)\\t_d\neq 0}}^{D}\alpha^{|t_d|-1}\Big].(B.3)\end{aligned}$$

Since we have

$$\sum_{\substack{t_k=-(D-1)\\t_k\neq 0}}^{D}\alpha^{|t_k|-1} = \alpha^{D-1}+2\sum_{t_k=1}^{D-1}\alpha^{|t_k|-1},$$

$$= \frac{1-\alpha^{D-1}}{1-\alpha} + \frac{1-\alpha^{D}}{1-\alpha} \stackrel{\text{def}}{=} \gamma,$$

Eq. (B.3) can be simplified to give

$$\begin{aligned} P_{success} &= \frac{q_0}{(p^d-1)}\left[\beta^{d-1}\gamma^d + \binom{d}{1}\beta^{d-2}\gamma^{d-1} + \cdots + \binom{d}{d-1}\beta\gamma\right] \\ &= \frac{q_0}{(p^d-1)}\frac{1}{\beta}\left[(1+\beta\gamma)^d - 1\right], \end{aligned}$$

which on substituting γ immediately yields Eq. (14).

C. Derivation of moments of retrial attempts

We derive the first and second moments of the number of retrial attempts k_i of session i. Since the probability of k_i retrials is $(1-P_h)^{k_i}P_h$ these moments are obtained simply by noting the identities

$$\sum_{j=0}^{\infty}(1-x)^j = \frac{1}{x}, \quad \sum_{j=0}^{\infty} j(1-x)^j = \frac{1-x}{x^2},$$

and

$$\sum_{j=0}^{\infty} j^2(1-x)^j = \frac{(1-x)(2-x)}{x^3}, \qquad (C.1)$$

which immediately yield

$$\overline{k} = \frac{1-P_h}{P_h} \quad \text{and} \quad \overline{k^2} = \frac{(1-P_h)(2-P_h)}{{P_h}^2}.$$

D. Derivation of mean residual time R

The mean residual time R will be calculated by using a graphical argument (see [BeG92]).

In Figure 12 we plot the residual service time $r(\tau)$ (that is, the remaining time for completion of the session at the head of the link input-queue at time τ) as a function of τ. Note that when session j proceeds to the head of the queue, $r(\tau)$ takes the value $X_j + C_j$, and decays linearly to zero for $X_j + C_j$ time units. Consider a time t at which $r(\tau) = 0$. The time average of $r(\tau)$ in the interval $[0, t]$ is given by

$$\frac{1}{t}\int_0^t r(\tau)d\tau = \frac{1}{t}\left[\frac{1}{2}\sum_{j=1}^{M(t)}(X_j + C_j)^2\right] \qquad (D.1)$$

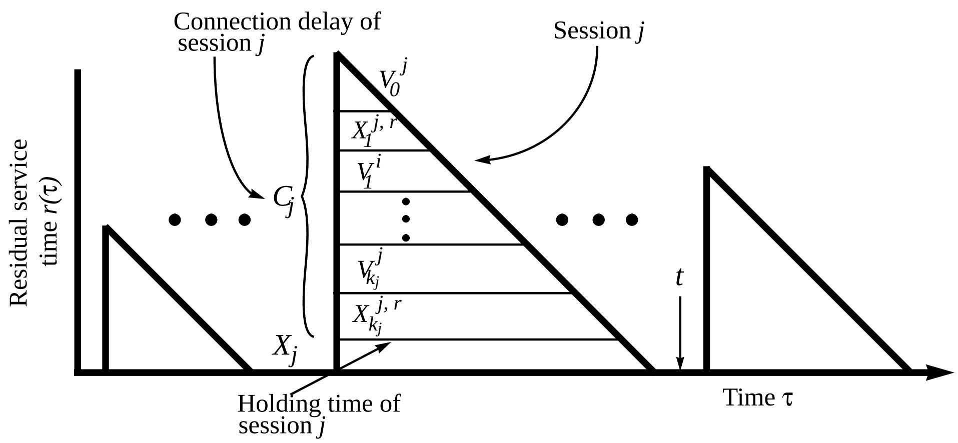

Fig. 12. Illustrating the residual service time of a session. The residual service time of the j^{th} session $X_j + C_j$, starts at $X_j + C_j$ when the session advances to the head of the queue, and decays linearly to zero for $X_j + C_j$ time units. The connection delay C_j of the session is given by $C_j = \sum_{l=1}^{k_j} \left(X_l^{j,r} + V_l^j \right) + V_0^j$.

where $M(t)$ is the number of sessions that have completed transmission in the time interval $[0, t]$. Equation (D.1) can be rewritten as

$$\frac{1}{t} \int_0^t r(\tau) d\tau = \frac{1}{2} \frac{M(t)}{t} \frac{\sum_{j=1}^{M(t)} (X_j + C_j)^2}{M(t)}.$$

Taking the limit as as $t \to \infty$, we obtain

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t r(\tau) d\tau = \frac{1}{2} \lim_{t\to\infty} \frac{M(t)}{t} \cdot \lim_{t\to\infty} \frac{\sum_{j=1}^{M(t)} (X_j + C_j)^2}{M(t)}, \qquad (D.2)$$

where we have assumed that the limits in Eq. (D.2) exist. Assuming that time averages can be replaced by ensemble averages, the first term at the right hand side of Eq. (D.2) corresponds to the time average of the departure rate at a link input-buffer, and it tends to the arrival rate $\lambda/2d$ as $t \to \infty$. Similarly, the second term at the right hand side of Eq. (D.2) tends to the second moment $\overline{(X+C)^2}$ of the sum of the connection and holding times. The limit at the left hand side is the time average of the residual service time, which tends to the mean residual time R. Thus, replacing time averages by ensemble averages, we obtain

$$R = \frac{\lambda}{4d} \overline{(X+C)^2}. \qquad (D.5)$$

E. Derivation of second moment $\overline{C^2}$ of the Connection delay

Squaring Eq. (19) and expanding the resulting expression we get

$$\begin{aligned}
C_i^2 &= \left[\sum_{j=1}^{k_i}\left(X_j^{i,r}+V_j^i\right)+V_0^i\right]^2 \\
&= \left[\sum_{j=1}^{k_i}\left(X_j^{i,r}+V_j^i\right)\right]^2 + 2V_0^i\sum_{j=1}^{k_i}(X_j^{i,r}+V_j^i)+{V_0^i}^2 \\
&= \left[\sum_{j=1}^{k_i}X_j^{i,r}\right]^2+\left[\sum_{j=1}^{k_i}V_j^i\right]^2+2\sum_{j=1}^{k_i}\sum_{j=1}^{k_i}X_j^{i,r}V_j^i \\
&\quad +2V_0^i\sum_{j=1}^{k_i}\left(X_j^{i,r}+V_j^i\right)+(V_0^i)^2 \\
&= \sum_{j=1}^{k_i}{X_j^{i,r}}^2+\sum_{j=1}^{k_i}\sum_{\substack{l=1\\l\neq j}}^{k_i}X_j^{i,r}X_l^{i,r}+\sum_{j=1}^{k_i}(V_j^i)^2+\sum_{j=1}^{k_i}\sum_{\substack{l=1\\l\neq j}}^{k_i}V_j^iV_l^i \\
&\quad +2\sum_{j=1}^{k_i}\sum_{j=1}^{k_i}X_j^{i,r}V_j^i+2V_0^i\sum_{j=1}^{k_i}\left(X_j^{i,r}+V_j^i\right)+{V_0^i}^2 . (E.1)
\end{aligned}$$

Taking expectations in Eq. (E.1), and conditioning on the number of retrial attempts k_i, we obtain

$$\begin{aligned}
E[{C_i}^2 \mid k_i] &= k_iE[{X^{i,r}}^2]+k_i(k_i-1)E^2[X^{i,r}]+k_iE[{V^i}^2]+k_i(k_i-1)E^2[V^i] \\
&\quad +2k_i^2E[X^{i,r}]E[V^i]+2k_iE[V^i]E[X^{i,r}+V^i], (E.2)
\end{aligned}$$

where we used the fact that $X_j^{i,r}$ and V_j^i are independent for all j together with the fact that both V_j^i and $X_j^{i,r}$ are i.i.d. for all j. Removing the conditioning on k_i in Eq. (E.2) and taking the limits as $i \to \infty$, we obtain

$$\begin{aligned}
\overline{C^2} &= \overline{k}\overline{(X^{r\,2})}+(\overline{k^2}-\overline{k})(\overline{X^r})^2+\overline{k}\,\overline{V^2}+(\overline{k^2}-\overline{k})\overline{V}^2 \\
&\quad +2\overline{k^2}\,\overline{X^r}\,\overline{V}+2\overline{k}\,\overline{V}(\overline{X^r}+\overline{V})+\overline{V}^2,
\end{aligned}$$

or more simply

$$\overline{C^2}=\overline{k}\overline{X^{r\,2}}+(\overline{k^2}-\overline{k})\overline{X^r}^2+(\overline{k}+1)\overline{V^2}+(\overline{k^2}+\overline{k})\overline{V}^2+2(\overline{k^2}+\overline{k})\overline{X^r}\,\overline{V}. \qquad (E.3)$$

Substituting Eq. (21) and Eq. (E.3) into Eq. (27) we get

$$\begin{aligned}
\overline{(X+C)^2} &= \overline{k}\overline{(X^r)^2}+(\overline{k^2}-\overline{k})(\overline{X^r})^2+(\overline{k}+1)\overline{V^2}+(\overline{k^2}+\overline{k})\overline{V}^2 \\
&\quad +2(\overline{k^2}+\overline{k})\overline{X^r}\,\overline{V}+2\overline{k}\,\overline{X^r}\,\overline{X}+2(\overline{k}+1)\overline{X}\,\overline{V}+\overline{X^2}. (E.4)
\end{aligned}$$

Since, as shown in Subsection 4.2.1, the distribution of the residual times X^r is related to the distribution of the session holding times X by $f_{X^r}(y) = (1 - F_X(y))/\overline{X}$, we have $\overline{X^r} = \overline{X^2}/2\overline{X}$ and $\overline{(X^r)^2} = \overline{X^3}/3\overline{X}$. Thus, Eq. (E.4) simplifies to

$$\begin{aligned}\overline{(X+C)^2} &= (\overline{k}+1)(\overline{X^2}+\overline{V^2}+2\overline{X}\,\overline{V}) + (\overline{k^2}+\overline{k})\overline{V}^2 \\ &\quad +2(\overline{k^2}+\overline{k})\frac{\overline{X^2}\,\overline{V}}{2\overline{X}} + (\overline{k^2}-\overline{k})\left(\frac{\overline{X^2}}{2\overline{X}}\right)^2 + \overline{k}\frac{\overline{X^3}}{3\overline{X}}. \quad (E.5)\end{aligned}$$

Finally, substituting for $\overline{k}$ and $\overline{k^2}$ from Appendix A3, we obtain Eq. (28) as

$$\begin{aligned}\overline{(X+C)^2} &= \frac{\left(\overline{X^2}+\overline{V^2}+2\overline{X}\,\overline{V}\right)}{P_h} + 2\frac{(1-P_h)}{P_h}\overline{V}\left(\overline{V}+\frac{\overline{X^2}}{\overline{X}}\right) \\ &\quad +\frac{(1-P_h)^2}{P_h^2}\left(\frac{\overline{X^2}}{2\overline{X}}\right)^2 + \frac{(1-P_h)}{P_h}\left(\frac{\overline{X^3}}{3\overline{X}}\right).\end{aligned}$$

F. Calculating $P_{success}(\infty)$

We perform the calculations for the case p odd, the case of p even being similar. Equations (10) and (11) can be rewritten as

$$\alpha = \frac{q_0}{1-q_2} = \frac{q_0}{1-(1-q_0)\left(\frac{p-3}{p+1}\right)} = \frac{q_0}{q_0+\frac{4(1-q_0)}{p+1}}, \qquad (F.1)$$

and

$$\beta = \frac{q_0}{1-\frac{q_1}{2}} = \frac{q_0}{1-\frac{2(1-q_0)}{d}\frac{\left(d(p-1)p^{d-1}-(p^d-1)\right)}{(p^{d+1}-p^{d-1})}}. \qquad (F.2)$$

In writing Eqs. (F.1) and (F.2), we have used the expressions for q_1 and q_2 given in Eqs. (7a) and (7b), respectively. Taking the limit $p \to \infty$, with q_0 being constant, we obtain

$$\lim_{p\to\infty}\alpha = 1, \qquad \text{and} \qquad \lim_{p\to\infty}\beta = q_0. \qquad (F.3)$$

Equation (F.1) also gives

$$\lim_{p\to\infty}(p+1)(1-\alpha) = \lim_{p\to\infty}\left(\frac{4(1-q_0)}{q_0+\frac{4(1-q_0)}{p+1}}\right) = \frac{4(1-q_0)}{q_0}. \qquad (F.4)$$

We define

$$P_{success}(\infty) = \lim_{p\to\infty} P_{success}(p),$$

where the dependence on the mesh size p is made explicit. To evaluate $P_{success}(\infty)$, we rewrite Eq. (13) as

$$P_{success} = \frac{q_0}{(p^d-1)}\frac{(p+1)^d}{\beta}$$

$$\times\left[\left(\frac{1}{(p+1)}+2\beta\left(\frac{1-\alpha^D}{(p+1)(1-\alpha)}\right)\right)^d-\frac{1}{(p+1)^d}\right].(F.5)$$

In order to evaluate $P_{success}(\infty)$ we also need to evaluate $\lim_{p\to\infty} 1-\alpha^{\lfloor\frac{p}{2}\rfloor}$. For p odd, we have

$$\begin{aligned}
\lim_{p\to\infty} 1-\alpha^{\lfloor\frac{p}{2}\rfloor} &= 1-\lim_{p\to\infty}\alpha^{\lfloor\frac{p}{2}\rfloor} = 1-\lim_{p\to\infty}\alpha^{\frac{(p-1)}{2}}\cdot\lim_{p\to\infty}\alpha \\
&= 1-\lim_{p\to\infty}\alpha^{(p+1)/2}\cdot\lim_{p\to\infty}\alpha^{-1} \\
&= 1-\lim_{p\to\infty}\left[\frac{q_0}{q_0+\frac{4(1-q_0)}{p+1}}\right]^{(p+1)/2} \\
&= 1-\lim_{p\to\infty}\left[\left(\frac{1}{1+\frac{4(1-q_0)}{q_0(p+1)}}\right)^{\frac{q_0(p+1)}{4(1-q_0)}}\right]^{\frac{2(1-q_0)}{q_0}} \\
&= 1-e^{-\frac{2(1-q_0)}{q_0}}, (F.6)
\end{aligned}$$

where we have used the identity $\lim_{x\to\infty}(1+\frac{1}{x})^x = e$. Thus, combining the results of Eqs. (F.3), (F.4), and (F.6) and substituting in Eq. (F.5) we get after some simplification

$$P_{success}(\infty) = q_0^{2d}\frac{(1-e^{-2\frac{(1-q_0)}{q_0}})^d}{2^d(1-q_0)^d}. \qquad (F.7)$$

IX. References

[ABP89] Abraham, S., Padmanabhan, K., "Performance of the direct binary n-cube network for multiprocessors," IEEE Trans. on Computers, Vol. 38, No. 7, pp. 1001-1011, July 1989.

[ALM91] Aiello, W. A., Leighton, F. T., Maggs, B. M., Newman, M., "Fast algorithms for bit-serial routing on a hypercube," Math. Sys. Theory, Vol. 24, No. 4, pp. 253-271, 1991.

[BeG92] Bertsekas, D., and Gallager, R, *Data Networks*, 2nd Edition, Prentice Hall, Englewood Cliffs, NJ, 1992.

[BrS91] Bruno, C., and Salvatore, T., "On modeling link conflict resolution strategies for circuit-switching hypercubes," Proc. Adv. Computer Technology, Reliable Systems, and Applications. 5th Annual European Computer Conf., CompEuro'91, pp. 662-666, May 1991.

[BYA89] Bhuyan, L. N., Yang, Q., and Agrawal, A. P., "Performance of multiprocessor interconnection networks," IEEE Computer, Vol. 22, No. 2, pp. 25-37, February 1989.

[CCR94] Chiu, Ge-Ming, Chalsani, C. S., and Raghvendra, C. S., "Flexible routing criteria for circuit-switched hypercubes," Jour. Parallel and Distributed Computing, Vol. 22, No. 2, pp. 279-284, August 1994.

[CGK91] Chlamtac, I., Ganz, A., and Kienzle, M., "A performance model of a connection based hypercube interconnection system," Proc. Fifth Int'l Conference Computer Performance Evaluation. Modelling Techniques and Tools, Torino, Italy, pp. 215-218, 13-15 February, 1991.

[ChS93] Choi, S. B., and Somani, A. K., "Rearrangeable circuit-switched architectures for routing permutations," Jour. Parallel and Distributed Computing, Vol. 19, No. 2, pp. 125-130, October 1993.

[CsS92] Csoppenszky, M. A., and Somani, A. K., "Efficient distributed routing algorithms for a synchronous circuit-switched hypercube," Proc. Int'l Phoenix Conf. on Computers and Communication, Scottsdale, AZ, pp. 15-22, 1-3 April 1992.

[Dal90] Dally, W. J., "Performance analysis of k-ary n-cube interconnection networks," IEEE Trans. on Computers, Vol. 39, No. 6, pp. 775-785, June 1990.

[DaS87] Dally, W. J., and Seitz, C. E., "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Trans. on Computers, Vol. C-36, No. 5, May 1987.

[GaY93] Gaughan, P. T., and Yalamanchili, S., "Analytical models of bandwidth allocation in pipelined k-ary n-cubes," Proc. Seventh Int'l Parallel Processing Symposium, Newport, CA, pp. 395-400, 13-16 April 1993.

[GeR88] Grunwald, D. C. and Reed, D. A., "Networks for parallel processors: measurements and prognostications," Conf. on Hypercube Multiprocessors, Pasadena, CA, pp. 610-619, January 1988.

[GeR89] Grunwald, D. C.,an Reed, D. A.,"Analysis of backtracking routing in binary hypercube computers," Research Report UIUCDCS-R-89-1486, Univ. of Illinois, Urbana-Champaign, Dept. of Computer Science, February 1989.

[Kle75] Kleinrock, L., *Queueing Systems – Volume I: Theory*, John Wiley and Sons, New York, 1975.

[KrS83] Kruskal, C. P., and Snir, M., "The performance of multistage interconnection networks for multiprocessors," IEEE Trans. on Computers, Vol. C-32, No. 12, pp.

10911098, Dec. 1983.

[ReF87] Reed, D. A., and Fujimoto, R. M., *Multiprocessor Networks: Message based Parallel Processing*, MIT Press, Scientific Computation Series, 1987.

[ReG87] Reed, D. A., and Grunwald, D. C., "The performance of multicomputer interconnection networks," IEEE Computer, Vol. 20 , No. 6 , pp. 63-73, June 1987.

[You92] Youssef, A., "Online communication on circuit-switched fixed routing meshes," Proc. Sixth Int'l Parallel Processing Symposium, Beverley Hills, CA, pp. 390-397, 23-26 March 1992.

Vishal Sharma received the B. Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India in 1991, and the M.S.(signals and systems) and M.S. (computer engineering) degrees in electrical and computer engineering in 1993, from the University of California, Santa Barbara, where he is currently completing the Ph.D. degree. During the summer of 1992, he worked at the Digital Technology Research Laboratory in Motorola's Corporate Research and Development Center, Schaumburg, IL, where he investigated algorithms for real-time resizing of decompressed video sequences. His research interests include communication networks and protocols, and parallel and distributed systems. More specifically, the design of protocols for multigigabit-per-second ATM networks, the performance evaluation of switching schemes for multiprocessor systems, and the performance analysis of lightwave networks. He is a student member of the IEEE Computer Society and the IEEE Professional Communication Society, and a student member of the ACM SIGCOMM and ACM SIGDOC.

Emmanouel A. Varvarigos was born in Athens, Greece, in 1965. He received a Diploma (1988) in electrical engineering from the National Technical University of Athens, Greece and the M.S. (1990), Electrical Engineer (1991), and Ph.D. (1992) degrees in electrical engineering and computer science from the Massachusetts Institute of Technology. In 1990 he conducted research on optical fiber communications at Bell Communications Research, Morristown. He is currently an assistant professor at the department of electrical and computer engineering at the University of California, Santa Barbara. His research interests are in the areas of parallel and distributed computation, optical-fiber data networks, and mobile communications. Dr. Varvarigos received the first panhellenic prize in the Greek Mathematic Olympiad in 1982. He is a member of the Technical Chamber of Greece.