Appeared in IEEE Transactions on Computers (June 91)

Performance Analysis of Virtual Cut-through Switching in HARTS: A Hexagonal Mesh Multicomputer *

James W. Dolter, P. Ramanathan, and Kang G. Shin

Real–Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109–2122
(313) 763–0391; E-Mail Address: kgshin@alps.eecs.umich.edu

Abstract

This paper presents a formal analysis of virtual cut-through in a C-wrapped hexagonal mesh multicomputer, called the **HARTS** (Hexagonal Architecture for Real–Time Systems), which is currently being built at the Real–Time Computing Laboratory, The University of Michigan. In virtual cut-through, packets arriving at an intermediate node are forwarded to the next node in the route without buffering if a circuit can be established to the next node.

The hexagonal mesh is first characterized using a combinatorial analysis to determine the probability that a packet will establish a cut-through at an intermediate node. Given this parameter the probability distribution function for packet delivery times in **HARTS** is derived. The delivery times obtained from the analytic model are then compared against results collected from a simulator of the routing hardware designed for use in **HARTS**. The results from both the analytic model and the simulator further reinforce the choice of the virtual cut-through routing scheme for use in **HARTS**.

Index Terms: Distributed real-time systems, wrapped hexagonal mesh, virtual cut-through, message buffering and delivery, queueing models.

* The work reported here is supported in part by the Office of Naval Research under Contracts N00014–85–K–0122 and N00014–85–K–0531. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the ONR.

1 Introduction

This paper derives an analytical model to evaluate the message passing scheme in a distributed computing system based on a hexagonal mesh architecture [1, 5, 6]. This effort is part of a larger research project to design and implement an experimental distributed real-time system, called the **HARTS** (Hexagonal Architecture for Real–Time Systems), at the Real–Time Computing Laboratory (RTCL), The University of Michigan.

A set of application processors along with a network processor form a node of **HARTS**. These nodes are interconnected in a C-wrapped[1] hexagonal mesh topology [1]. The application processors execute real-time tasks and the network processor handles all the intra– and inter– node communications. Since real-time applications normally require short response times, simple store-and-forward message passing schemes are not suitable for **HARTS**. Consequently, **HARTS** uses a message passing scheme commonly referred to as the *virtual cut-through* [3].

In virtual cut-through, packets arriving at an intermediate node are forwarded to the next node in the route without buffering if a circuit can be established to the next node. This differs from conventional packet-switching schemes in the sense that packets do not always get buffered at an intermediate node. It also differs from conventional circuit switching schemes since packets do not wait for the entire circuit to the destination to be established before proceeding along the route.

Although virtual cut-through was proposed almost a decade ago, it has not been implemented in real systems until recently. Since custom ASICs have become economically viable, several distributed systems are being designed and implemented that use virtual cut-through (or some variant thereof) as their basic message passing scheme. It is easy to see that virtual cut-through will perform better than a conventional packet-switching scheme in terms of packet delivery times. However, the actual improvement it offers over a packet-switching scheme for packet deliveries has not yet been clearly evaluated.

Kermani and Kleinrock carried out a mean value analysis of the performance of virtual cut-through for a general interconnection network [3]. However, a mean value analysis is not adequate for real-time applications because worst-case communication delays often play an important role in the design of real-time systems. For example, the mean value analysis cannot answer questions like what is the probability of a successful delivery given a delay

[1]To be defined later.

or what is the delay bound such that the probability of a successful delivery is greater than a specified threshold.

The authors of [3] wanted to avoid any dependence on the interconnection topology in their analysis. As a result, they assumed that the probability of a packet getting buffered at an intermediate node is a given parameter. Since one cannot get a reasonable estimate of the performance of virtual cut-through without an accurate estimate of the probability of buffering, the approach in [3] becomes useful only if we can accurately determine the probability of buffering for a given interconnection topology. However, determining the probability of buffering at an intermediate node for a given topology is not simple. This is because each node in a distributed system handles not only all packets generated at the node but also all packets passing through the node (or *transit packets*). Consequently, to evaluate the probability of buffering, we have to account for the fraction of packets generated at other nodes that pass through each given node.

In contrast to [3], in this paper, we first derive the probability that a packet is destined for a particular node by characterizing the hexagonal mesh topology. This *probability of branching* is then used as a parameter in a queueing network to determine the *throughput rates* at each node in the mesh. After the throughput rates are found, the probability that a packet can establish a cut-through at an intermediate node is derived. From these parameters we derive the probability distribution function of delivery times for a packet traversing a specified number of hops. The importance of this kind of analysis in a real-time system, as opposed to a mean value analysis, is then illustrated through some numerical examples and compared with simulation results that are based on relevant parameters in **HARTS**.

The paper is organized as follows. Section 2 formally describes a C-wrapped hexagonal mesh topology. For completeness, a brief description of **HARTS** is also presented there. The terms and notation used in the paper are introduced in Section 3.1. Analytical expressions for the branching probability and buffering probability are derived in Section 3.2 and the probability distribution function of packet delivery times is derived in Section 3.3. Numerical results from both the analytic model and simulations are presented and compared in Section 4. The paper concludes with Section 5.

2 Description of HARTS

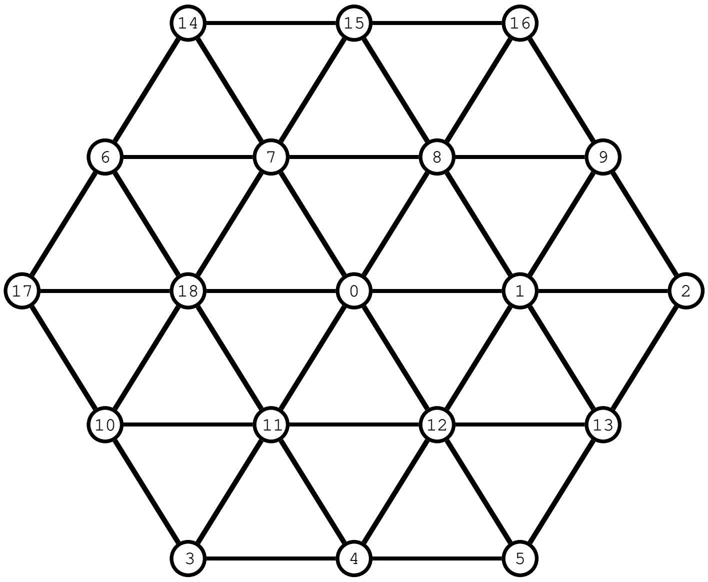
HARTS is an experimental testbed for research in distributed real-time computing. The primary goal of **HARTS** is to investigate low-level architectural issues in the design of real-time systems such as packet scheduling, routing and buffering. The *dimension* of a hexagonal (H–) mesh is defined as the number of nodes on a peripheral edge of the H-mesh. The current version of **HARTS** under construction at RTCL is a 3-dimensional H–mesh and is comprised of 19 nodes interconnected in a C-wrapped H–mesh topology, which is formally defined as follows.

Definition 1 A *C-wrapped hexagonal mesh* of dimension e is comprised of $3e(e-1)+1$ nodes, labeled from 0 to $3e(e-1)$, such that each node s has six neighbors $[s+1]_{3e^2-3e+1}$, $[s+3e-1]_{3e^2-3e+1}$, $[s+3e-2]_{3e^2-3e+1}$, $[s+3e(e-1)]_{3e^2-3e+1}$, $[s+3e^2-6e+2]_{3e^2-3e+1}$, and $[s+3e^2-6e+3]_{3e^2-3e+1}$, where $[a]_b$ denotes a mod b.

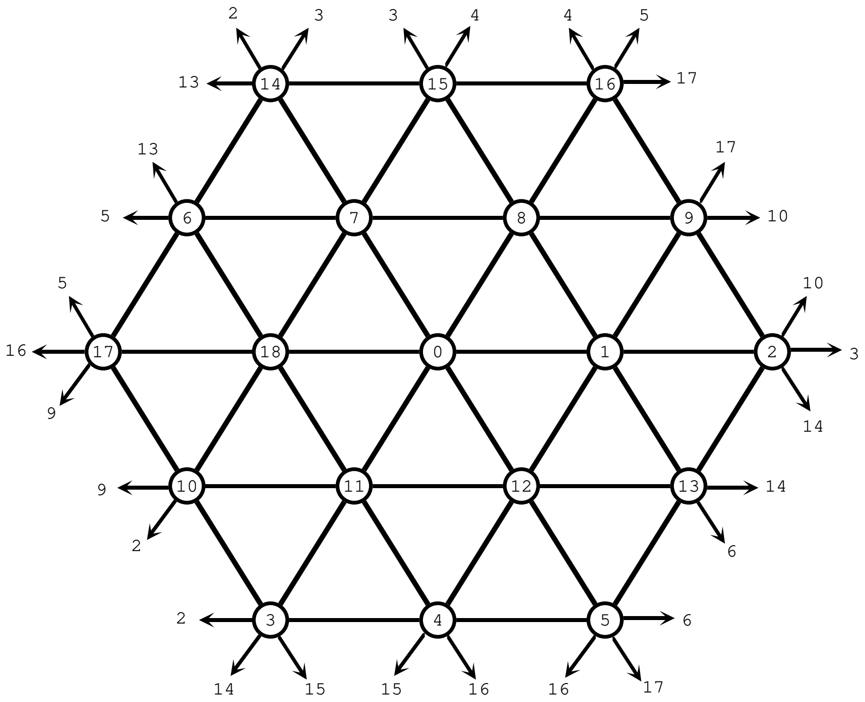
This topology can be visualized as follows. Consider an unwrapped H-mesh of dimension e. Figure 1(a) shows an unwrapped H-mesh of dimension 3. It is easy to see in this figure that the nodes of a H-mesh of dimension e can be partitioned into $2e-1$ rows in three possible ways: either along the horizontal direction or along the 60 degrees counter-clockwise direction or along 120 degrees counter-clockwise direction. Along any one of these directions, let R_0 be the top row, R_1 be the second row, and so on till R_{2e-2}. Then a *C-type* wrapping can be obtained by wrapping the last processor in R_i to the first processor in $R_{[i+e-1]_{2e-1}}$. For example, in Figure 1(b), the last processor in R_2 along the horizontal direction, namely node 2, is wrapped to the first processor in R_4, node 3.

A C-type wrapping has several nice properties as reported in [1]. First, this wrapping results in a homogeneous network. Consequently, any node can view itself as the center (labeled as node 0) of the mesh. Second, the diameter of a H-mesh of dimension e is $e-1$. Third, there is a simple, transparent addressing scheme such that the shortest paths between any two nodes can be determined by a $\Theta(1)$ algorithm given the address of the two nodes. At each node on a shortest path there are at most two different neighbors of the node to which the shortest path runs. Fourth, based on this addressing scheme it is possible to devise a simple routing and broadcast algorithms that can be efficiently implemented in hardware [2].

The six neighbors of a node in a C-wrapped H-mesh can be thought of as being in



(a)



(b)

Figure 1: A hexagonal mesh of dimension 3.

directions d_0, d_1, $\ldots$, d_5. The neighbor of a node s in direction d_i will be denoted by $cwhm(s, d_i)$. Similarly, given two nodes m and n in the H-mesh that have a direct link between them, we can denote the direction of n with respect to m by $cwhm^{-1}(m, n)$. Both $cwhm$ and $cwhm^{-1}$ can be formally described from the definition of a C-wrapped H-mesh. Another useful notation when dealing with the C-wrapped H-mesh is the concept of the complement of a direction d_i, denoted by $\overline{d_i}$, such that $\overline{d_i} \equiv d_{[i+3]_6}$

3 Modeling of Message Delivery

This section presents the derivation of the probability distribution of packet delivery times in a C-wrapped H–mesh that implements virtual cut-through. A queueing network will be used to carry out this analysis.

To make the analysis tractable, we make the following assumptions:

A1: Poisson packet generation with rate λ_G at each node.

A2: Exponentially distributed packet lengths with mean $\overline{\ell}$.

A3: The length of a packet is regenerated at each intermediate node of its route independently of its length at other intermediate nodes.

A4: Nodes have no preferential direction for communication.

Assumptions A1–A3 are consistent with Kermani and Kleinrock's assumptions in [3]. Although not completely accurate, it has been shown through empirical studies that these assumptions lead to a fairly accurate characterization of message arrivals. Assumption A4 implies that all minimal length paths between a source and destination are equally used. A4 does not imply uniform communication over all nodes of the mesh, but implies uniform communication with nodes reachable in the same number of hops. So, let q_k denote the probability of a node communicating with a node which is k hops away. The definition of q_k will be used to derive some of the base parameters for the queueing network.

Due to the homogeneity of a C-wrapped H-mesh, any node can be considered as the origin of the mesh and labeled 0. Without loss of generality, we can concentrate on evaluating the distribution of the packet delivery times for the packets generated at node 0. In order to determine the distribution of the delivery times it will be necessary to evaluate the transit load handled by node 0. This transit load is a function of both the packet generation rate at each node and the interconnection topology. Another parameter necessary to determine the distribution of the delivery times is the probability that a transit packet (at node 0) will be buffered (at node 0) as a result of not being able to establish a circuit to the neighboring node. The derivation of the analytical expressions for the transit load and the probability of buffering is presented in Section 3.2. The distribution for the packet delivery times is then presented in Section 3.3.

3.1 Terms and Notation

In the following analysis let e be the dimension of the H-mesh and let $[j]_i$ denote j mod i. Also let $N = \{0, 1, \ldots, 3e(e-1)\}$ be the set of all nodes in the H-mesh.

Definition 2 A *route* from a source node, $s \in N$, to a destination node, $d \in N$, is a sequence $n_0 n_1 \cdots n_i \cdots n_{k-1} n_k$ of nodes, $n_i \in N$, $\forall\, i \in \{0, 1, \ldots, k\}$, such that (i) $n_0 = s$, $n_k = d$, and (ii) there exists a direct link in the H-mesh between n_i and $n_{i+1}, \forall i \in \{0, \ldots, k-1\}$. The *length* of a route r is the number of components in the sequence and will be denoted by $len(r)$.

Definition 3 A *minimal route* from $s \in N$ to $d \in N$ is a route r_1 from s to d such that $len(r_1) \leq len(r_2)$ for all routes r_2 from s to d.

Definition 4 An *anchored route* is an ordered pair $(n_0 \cdots n_k,\ x)$ consisting of a route $n_0 \cdots n_k$ and $x \in N$ such that

1. $k \geq 2$,

2. $n_0 \cdots n_k$ is a minimal route from n_0 to n_k, and

3. $\exists\, i,\ 1 \leq i \leq k-1$, such that $n_i = x$.

x is called the *anchor* of $(n_0 \cdots n_k,\ x)$.

Definition 5 A *shape* s of length k, $2 \leq k \leq e-1$, is a sequence $a_1 a_2 \cdots a_i \cdots a_{k-1} a_k$, $a_i \in \{d_0, \ldots, d_5\}$, such that

$$\begin{aligned}\bigcup_{i=1}^{k} \{a_i\} \;\; &\in \;\; \bigcup_{j=0}^{5} \{\{d_j\}\} \;\cup\; \bigcup_{j=0}^{5} \left\{\{d_j, d_{[j+1]_6}\}\right\} \\ &\equiv \;\; \{\ \{d_0\}, \{d_1\}, \{d_2\}, \{d_3\}, \{d_4\}, \{d_5\}, \\ &\qquad \{d_0, d_1\}, \{d_1, d_2\}, \{d_2, d_3\}, \{d_3, d_4\}, \{d_4, d_5\}, \{d_5, d_0\}\ \}.\end{aligned}$$

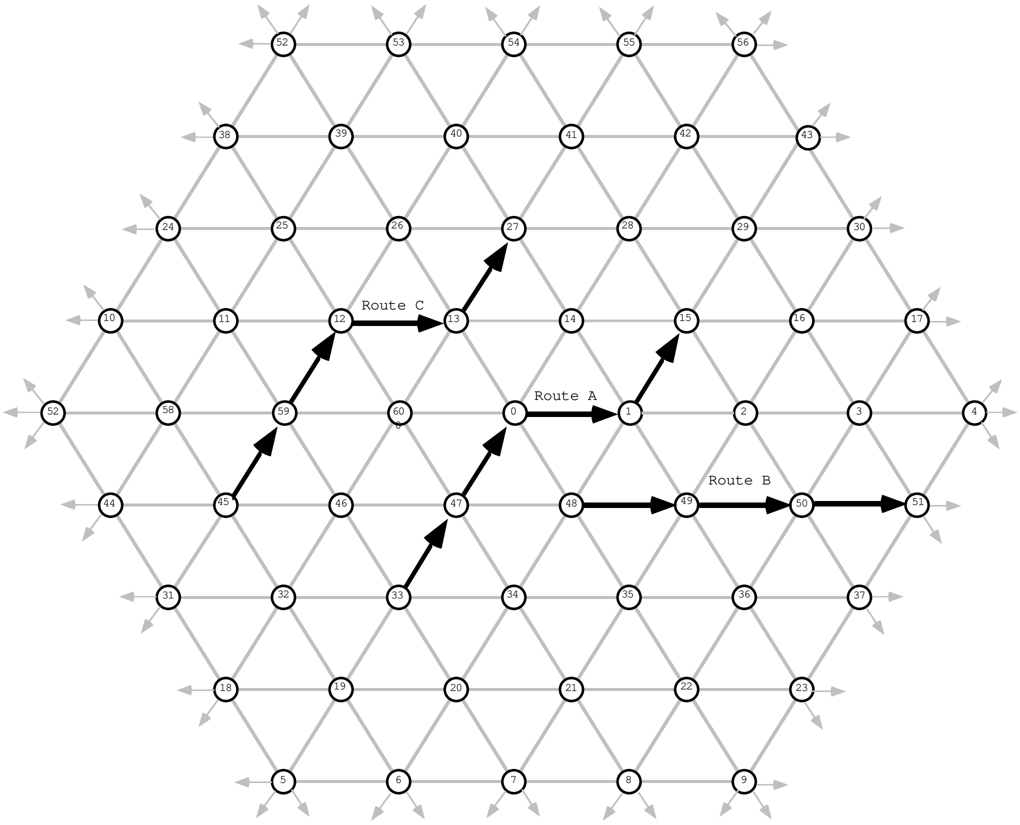


Figure 2: Example shapes in a H–mesh of dimension 5.

The length of shape s is denoted by $\ell(s)$.

A shape is a route that a packet can traverse. The above definition of a shape is motivated by the fact that all minimal routes between any pair of nodes are formed by links along one or two directions only [1]. For example, route A in Figure 2 corresponds to the shape $d_1d_1d_0d_1$ such that $\cup_{i=1}^{4}\ \{a_i\} = \{d_0, d_1\}$ and route B corresponds to the shape $d_0d_0d_0$ such that $\cup_{i=1}^{3}\ \{a_i\} = \{d_0\}$. A shape can represent routes between several different pairs of communicating nodes. For example, routes A and C in Figure 2 correspond to the same shape but represent routes between two different pairs of communicating nodes.

Definition 6 An *anchored shape* p is an ordered pair (s, k), where s is a shape, and $1 \leq k \leq \ell(s) - 1$ marks a position within the shape. The length of an anchored shape (s, k) is defined to be the length of the associated shape s.

There exists a one-to-one correspondence between the set of all anchored shapes and the set of all anchored routes with their anchor at 0. (In order to not detract from the main goal of this paper the proof that there is a one-to-one correspondence between these mappings has been given in Appendix A.) The mapping from an anchored shape $(a_1 \cdots a_\ell \cdots a_k, \ell)$ to an anchored route $(n_0 \cdots n_k, 0)$ is done as follows:

$$n_i = \begin{cases} cwhm(n_{i+1}, \overline{a_{i+1}}) & \text{if } 0 \leq i \leq \ell - 1 \\ 0 & \text{if } i = \ell \\ cwhm(n_{i-1}, a_i) & \text{if } \ell + 1 \leq i \leq k \end{cases} \tag{3.1}$$

where $\overline{a_i} = d_{[m+3]_6}$ if $a_i = d_m$, $0 \leq m \leq 5$. The mapping from an anchored route $(n_0 \cdots n_k, 0)$ to an anchored shape $(a_1 \cdots a_\ell \cdots a_k, \ell)$ is

$$\begin{aligned} a_i &= cwhm^{-1}(n_{i-1}, n_i) \quad 1 \leq i \leq k \\ \ell &= \arg_{1 \leq j \leq k-1}(n_j = 0). \end{aligned} \tag{3.2}$$

where $\arg_{1 \leq j \leq k-1}(n_j = 0)$ refers to the value of j such that $n_j = 0$. For example, consider the anchored route $(33 \rightarrow 47 \rightarrow 0 \rightarrow 1 \rightarrow 15,\ \ 0)$ obtained from route A in Figure 2. From Definition 1, we know that $cwhm^{-1}(33, 47) = d_1$, $cwhm^{-1}(47, 0) = d_1$, $cwhm^{-1}(0, 1) = d_0$ and $cwhm^{-1}(1, 15) = d_1$. Since 0 is the third node in the route $\arg_{1 \leq j \leq 4}(n_j = 0) = 2$. It then follows from Equation (3.1) that the anchored shape corresponding to the anchored route $(33 \rightarrow 47 \rightarrow 0 \rightarrow 1 \rightarrow 15,\ \ 0)$ is $(d_1 d_1 d_0 d_1,\ 2)$.

Similarly, the anchored route $(n_0 n_1 n_2 n_3 n_4,\ 0)$ corresponding to the anchored shape $(d_1 d_1 d_0 d_1,\ 2)$ can be obtained as follows. Since the second element in the anchored shape is 2, $n_2 = 0$. Since $\overline{d_1} = d_4$ and $cwhm(0, d_4) = 47$, $n_1 = 47$. Proceeding further, we get $n_0 = 33$ because $cwhm(47, d_4) = 33$, and $n_3 = 1$ because $cwhm(0, d_0) = 1$. Finally, $n_4 = 15$ since $cwhm(1, d_1) = 15$. As expected, combining these results we get the anchored route corresponding to the anchored shape $(d_1 d_1 d_0 d_1, 2)$ as $(33 \rightarrow 47 \rightarrow 0 \rightarrow 1 \rightarrow 15,\ \ 0)$.

The minimal route corresponding to the anchored shape p is one of the possibly many minimal routes between the source and the destination. The other minimal routes between the source and the destination can be obtained by permuting the components of the shape associated with p and applying a mapping function similar to the one above.

All of the routes associated with these permutations will not necessarily go through node 0. Only the fraction of the total number of routes from the source to the destination that pass through node 0 will influence the transit load at node 0.

3.2 Model Derivation and Parameter Calculation

The packet transmission in the H-mesh can be modeled as a Jackson queueing network, consisting of $3e(e-1)+1$ service centers of the M/M/1 type. For each service center a packet completing its service may go to either of its six immediate neighbors or exit from the system. Packets whose final destinations are immediate neighbors will not use the service centers of their immediate neighbors and will exit the system at the current service center. Packets whose final destination are <u>not</u> immediate neighbors travel to a neighboring service center.

Let $p_{b_{ij}}$ denote the probability that a packet completing its service at a node i will be routed to neighboring node j. Using assumption A4 and the fact that the C-Wrapped H-Mesh is a homogeneous surface it is easily seen that all the $p_{b_{ij}}$ have to be equivalent and thus will be denoted by p_b. Figure 3 shows a portion of the queuing network centered around node i.

The rest of this section concentrates on deriving an expression for p_b. Once given an expression for this, we can derive the probability that a packet will establish a cut-through when arriving at a node in the H-mesh.

3.2.1 Calculation of p_b

The following symbols are used to identify the different packet arrival rates:

- λ_G : the rate of generating packets at a node.
- $\lambda_{G^{2+}}$: the rate of generating packets at a node that are not destined for an immediate neighbor.
- λ_T : the rate of transit packets arriving at a node.
- $\lambda_{T^{2+}}$: the rate of transit packets arriving at a node that are not destined for an immediate neighbor.

It is convenient to define a function $\Phi(d_j, p)$ that counts the total number of d_j's in the shape associated with the anchored shape p, that is,

$$\Phi(d_j, p) = |\ \{a_k : a_k = d_j,\ a_k \text{ is in the shape associated with anchored shape } p\ \}\ | .$$

Considering the anchored shape A from the previous example, $\Phi(d_0\ ,\ (d_1 d_1 d_0 d_1, 2)) = 1$ and $\Phi(d_1\ ,\ (d_1 d_1 d_0 d_1, 2)) = 3$. This function is used to derive the transit load associated

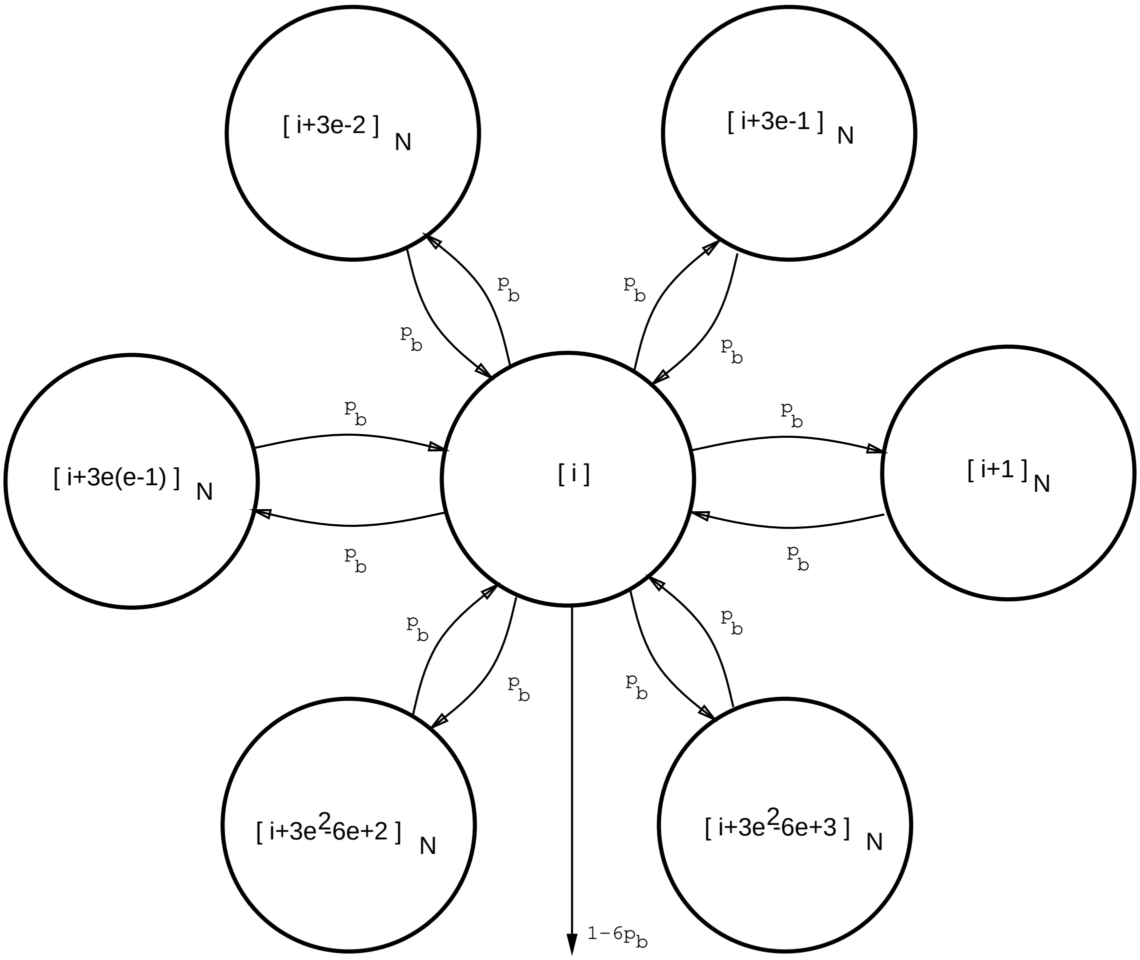


Figure 3: Network model around node i.

with an anchored shape on node 0.

Lemma 1 The contribution of an anchored shape p to the transit load of node 0 is

$$L(p) = \frac{\lambda_G \cdot q_k}{M(p)}$$

where $k = \sum_{j=0}^{5} \Phi(d_j, p)$ and $M(p) = \dfrac{\left[\sum_{j=0}^{5} \Phi(d_j, p)\right]!}{\prod_{j=0}^{5} \left[\Phi(d_j, p)!\right]}$.

Proof: It follows from the definition of anchored shape and a simple combinatorial analysis that the total number of shortest routes between a source–destination pair is $M(p)$.

By the definition of q_k, the rate at which a source sends packets to a destination is $\lambda_G \cdot q_k$. By 4, all routes between the source and the destination are equally used. Hence, $L(p) = \frac{q_k \cdot \lambda_G}{M(p)}$. ■

Lemma 1 allows us to calculate the transit load for a single route through node 0. In order to calculate the total transit loads λ_T and λ_{T^2+}, we will need to determine the total number of minimal routes passing through node 0 for all pairs of communicating nodes. To determine this number we will partition the set of all anchored shapes into sets that can be counted. Since there is a one-to-one correspondence between the anchored shapes and anchored routes with their anchor at node 0, counting all anchored shapes is equivalent to counting all pair of nodes that have a minimal route passing through node 0.

Partition the set of all anchored shapes P into the sets $P_{mn} \stackrel{\mathrm{def}}{=} \{\ p : \ \Phi(d_m, p) \geq 1\ \}$ for $0 \leq m \leq 5$ and $n = [m+1]_6$. Intuitively, each P_{mn} contains anchored shapes with one or more d_m and possibly some d_n components.

Lemma 2 The sets $P_{mn} \stackrel{\mathrm{def}}{=} \{\ p : \ \Phi(d_m, p) \geq 1, p \in P\ \}, \quad n = [m+1]_6, \quad 0 \leq m \leq 5$ partition P.

Proof: We will first show that sets P_{mn} cover the entire set P. For an anchored shape $p \equiv (a_1 a_2 \cdots a_i \cdots a_k, \ell)$, there are two cases to consider.

In the first case,

$$\bigcup_{i=1}^{k} \{a_i\} \quad = \quad \{d_{i_1}\}, \quad \text{for some } i_1 \in \{0, 1, 2, 3, 4, 5\}.$$

From this fact we can conclude that $p \in P_{i_1 [i_1+1]_6}$.

In the second case,

$$\bigcup_{i=1}^{k} \{a_i\} \quad = \quad \left\{d_{i_1}, d_{[i_1+1]_6}\right\}, \quad \text{for some } i_1 \in \{0,1,2,3,4,5\}.$$

In this case, $p \in P_{i_1[i_1+1]_6}$.

We will now show that the sets P_{mn} are disjoint. Suppose not. Then, $\exists\ P_{i_1 j_1}$ and $P_{i_2 j_2}$, $i_1 \neq i_2$, such that $P_{i_1 j_1} \cap P_{i_2 j_2} \neq \emptyset$. Consider an anchored shape $p \in P_{i_1 j_1} \cap P_{i_2 j_2}$ with the shape $a_1 a_2 \cdots a_k$.

Case 1: $j_1 = i_2$.

$p \in P_{i_1 j_1}$ implies $d_{i_1} \in \bigcup_{i=1}^{k} \{a_i\}$ and, $p \in P_{i_2 j_2}$ implies $\bigcup_{i=1}^{k} \{a_i\} \in \{\{d_{i_2}, d_{j_2}\}, \{d_{i_2}\}\}$. Since by construction $j_1 = [i_1+1]_6$ and $j_2 = [i_2+1]_6$, and by the case under consideration $i_2 = j_1$ we can conclude that $i_1 \neq i_2$ and $i_1 \neq j_2$. But, $d_{i_1} \notin \{d_{i_2}, d_{j_2}\}$ and $d_{i_1} \notin \{d_{i_2}\}$, a contradiction.

Case 2: $j_1 \neq i_2$.

$p \in P_{i_1 j_1}$ and $p \in P_{i_2 j_2}$ imply $\{d_{i_1}, d_{i_2}\} \subseteq \bigcup_{j=1}^{k} \{a_j\}$. This would violate the definition of an anchored shape since $j_1 = [i_1+1]_6 \neq i_2$. ∎

In order to calculate the total transit load λ_T we will need to further refine the partition P_{mn} into the sets $P_{mn}^{ab} \equiv \{\ p :\ p \in P_{mn},\ \Phi(d_m,\ p) = a,\ \Phi(d_n,\ p) = b\ \}$. The proof that P_{mn}^{ab} is a refinement of P_{mn} is straightforward and thus omitted.

We are now in a position to derive λ_T.

Lemma 3 The total transit load at node 0 is given by

$$\lambda_T = \lambda_G \sum_{k=2}^{e-1} 6k(k-1) \cdot q_k$$

where λ_G is the total rate of packet generation at a node, and q_k is the probability of a node communicating with a node k hops away.

Proof: Since there is a one-to-one correspondence between the anchored shapes and all minimal routes through node 0,

$$\begin{aligned}\lambda_T &= \sum_{p \in P} L(p), \quad \text{where } P \text{ is the set of all anchored shapes} \\ &= \sum_{i=0}^{5} \sum_{p \in P_{i[i+1]_6}} L(p).\end{aligned}$$

From the definitions of shapes and anchored shapes, the length of the shape associated with the above anchored shape p lies between 2 and $e-1$. From the definition of P_{mn} we know that $\Phi(d_m,\ p) \geq 1$. It follows from these observations that

$$\lambda_T \;=\; \sum_{i=0}^{5} \sum_{\substack{1 \leq a \leq e-1 \\ 2 \leq a+b \leq e-1}} \sum_{p \in P^{ab}_{i[i+1]_6}} L(p). \tag{3.3}$$

Note that all of the anchored shapes $p \in P^{ab}_{mn}$ have length $a+b$. Furthermore, since each shape associated with the anchored shapes of P^{ab}_{mn} has only components in the d_m and d_n directions, there are $\begin{pmatrix} a+b \\ a \end{pmatrix}$ shapes in P^{ab}_{mn}. Given each shape one can then derive $a+b-1$ anchored shapes. Therefore

$$\begin{aligned} \left|P^{ab}_{i[i+1]_6}\right| &= \begin{pmatrix} a+b \\ a \end{pmatrix} \cdot (a+b-1) \\ &= \frac{(a+b)!}{a! \cdot b!} \cdot (a+b-1) \\ &= \frac{[\sum_{l=0}^{5}\ \Phi(d_l,p)]!}{\prod_{l=0}^{5}\ [\Phi(d_l,p)!]} \cdot (a+b-1), \quad p \in P^{ab}_{i[i+1]_6}. \end{aligned} \tag{3.4}$$

Combining Equations (3.3) and (3.4) with Lemma 1, we get

$$\lambda_T = \sum_{i=0}^{5} \sum_{\substack{1 \leq a \leq e-1 \\ 2 \leq a+b \leq e-1}} \lambda_G \ \cdot q_{a+b} \cdot (a+b-1). \tag{3.5}$$

Since Equation (3.5) depends only on $(a+b)$, we can substitute k for $a+b$ to obtain

$$\begin{aligned} \lambda_T &= \sum_{i=0}^{5} \sum_{k=2}^{e-1} \lambda_G \ \cdot q_k \cdot (k-1) \cdot k \\ &= \lambda_G \sum_{k=2}^{e-1} \sum_{i=0}^{5} k(k-1) \cdot q_k \\ &= 6\lambda_G \sum_{k=2}^{e-1} k(k-1) \cdot q_k. \qquad \blacksquare \end{aligned}$$

Lemma 4 The transit load at node 0 for packets not bound for an immediate neighbor is given by

$$\lambda_{T^{2+}} = \lambda_G \sum_{k=3}^{e-1} 6k(k-2) \cdot q_k.$$

Proof: The proof of this lemma follows closely that of Lemma 3 with the additional restriction that node 0 cannot be in the last position for the anchored shapes being counted. Having node 0 in the last position of a anchored shape corresponds to having the anchored shape terminate in an immediate neighbor. This is exactly the traffic that we are trying to eliminate.

Since there is a one-to-one correspondence between the anchored shapes and all minimal routes through node 0,

$$\begin{aligned}\lambda_{T^{2+}} &= \sum_{\substack{(s,k)\in P\\ k\neq len(s)-1}} L((s,k)), \quad \text{where } P \text{ is the set of all anchored shapes}\\ &= \sum_{i=0}^{5} \sum_{\substack{(s,k)\in P_{i[i+1]_6}\\ k\neq len(s)-1}} L((s,k)).\end{aligned}$$

In contrast to Lemma 3, the lengths of shape s associated with the above restricted anchored shape (s,k) lies between 3 and $e-1$. From the definition of P_{mn} we know that $\Phi(d_m,\ (s,k)) \geq 1$. It follows from these observations that

$$\lambda_{T^{2+}} = \sum_{i=0}^{5} \sum_{\substack{1\leq a\leq e-1\\ 3\leq a+b\leq e-1}} \sum_{\substack{(s,k)\in P^{ab}_{i[i+1]_6}\\ k\neq len(s)-1}} L((s,k)). \tag{3.6}$$

Note that all of the anchored shapes $(s,k)\in P^{ab}_{mn}$ have length $a+b$. Furthermore, since each shape s associated with the anchored shapes of P^{ab}_{mn} has only components in the d_m and d_n directions, there are $\begin{pmatrix} a+b \\ a \end{pmatrix}$ shapes in P^{ab}_{mn}. Given each shape one can then derive $a+b-2$ anchored shapes. The number of anchored shapes generated from each shape differs by 1 from Lemma 3 since we cannot use the last position in the shape. Therefore following that same steps as in Lemma 3 we arrive at

$$\begin{aligned}\lambda_{T^{2+}} &= \sum_{i=0}^{5} \sum_{\substack{1\leq a\leq e-1\\ 3\leq a+b\leq e-1}} \lambda_G \cdot q_{a+b}\cdot (a+b-2)\\ &= \sum_{i=0}^{5} \sum_{k=3}^{e-1} \lambda_G \cdot q_k \cdot (k-2)\cdot k\\ &= \lambda_G \sum_{k=2}^{e-1} \sum_{i=0}^{5} k(k-2)\cdot q_k\\ &= 6\lambda_G \sum_{k=3}^{e-1} k(k-2)\cdot q_k. \qquad \blacksquare\end{aligned}$$

Theorem 1 The branching probability, p_b, between adjacent service centers in the model is

$$p_b = \frac{1}{6} \cdot \left(1 - \frac{1}{6 \cdot \sum_{k=1}^{e-1} k^2 \cdot q_k}\right)$$

Proof: p_b can be derived as the ratio of traffic bound for immediate neighbors to all traffic leaving a service center.

$$p_b = \frac{1}{6} \cdot \frac{\lambda_{G^{2+}} + \lambda_{T^{2+}}}{\lambda_G + \lambda_T}$$

Using the results of Lemmas 3 and 4 along with

$$\begin{aligned} \lambda_{G^{2+}} &= \lambda_G(1 - 6q_1) \\ \sum_{k=1}^{e-1} 6k \cdot q_k &= 1 \end{aligned}$$

the theorem follows after some algebraic manipulation. ∎

It should be noted that p_b only depends on q_k and the topology.

Lemma 5 The throughput at each service center is $6 \cdot \lambda_G \cdot \sum_{k=1}^{e-1} k^2 \cdot q_k$.

Proof: Jackson theorem[4] states that the total throughput T_i at service center i is given by the solution to the set of traffic flow equations

$$T_i = \lambda_G + \sum_{k=0}^{3e(e-1)} p_b \cdot T_k \ , \ i = 0, \ldots, 3e(e-1).$$

By assumption A4 and the homogeneous nature of the C-wrapped H-mesh all T_i are equal. Therefore,

$$T_i = \frac{\lambda_G}{1 - 6p_b} \ , \ i = 0, \ldots, 3e(e-1).$$

Substituting p_b from Theorem 1 the lemma immediately follows. ∎

Theorem 2 The probability of a packet cutting-through an intermediate node is

$$p_c = 1 - \left(\lambda_G \sum_{k=1}^{e-1} k^2 \cdot q_k\right) \cdot \overline{\ell}$$

where $\overline{\ell}$ is the mean length or service time for packets.

Proof: A packet can establish a cut-through at an intermediate node only if there are no packets being serviced or waiting for service at that node. Using Lemma 5 and Jackson's theorem that the probability of having zero packets at any node is $1-\rho$ where ρ is the traffic intensity. ρ in terms of the throughput and service rate μ is given as follows:

$$\rho = \frac{T}{\mu} = \frac{T \cdot \overline{\ell}}{6} = \left(\lambda_G \sum_{k=1}^{e-1} k^2 \cdot q_k\right) \cdot \overline{\ell}.$$

and hence the theorem follows. ■

3.3 Distribution of Message Delivery Times

In a virtual cut-through message passing scheme, the delay that a packet incurs at a node depends on whether the packet is able to establish a cut-through at that node. If the packet establishes a cut-through, the delay incurred is negligible and assumed to be 0. Otherwise, the packet incurs both waiting and service time delays. Furthermore, since a packet cannot establish a cut-through unless there are no other packets waiting for service at that node, the FCFS queueing discipline is preserved at each node. From Jackson's theorem we know that the queueing network described in Section 3.2 has a product form solution. Therefore, each service center behaves as an M/M/1 queueing system.

The delivery time for a packet traveling n hops, denoted by D_n, can be expressed as

$$D_n = Y_0 + X_{n-1},$$

where Y_0 (X_{n-1}) is a random variable that represents the total time spent by a packet at the source node ($n-1$ intermediate nodes). Also let Y_k be a random variable that represents the total time spent by a packet buffered in an intermediate node.

Therefore,

$$\begin{aligned}
P[D_n \le t] &= P[Y_0 + X_{n-1} \le t] \\
&= \sum_{m=0}^{n-1} P[Y_0 + X_{n-1} \le t \mid \text{buffered at } m \text{ int. nodes}] \cdot P[\text{buffered at } m \text{ int. nodes}] \\
&= \sum_{m=0}^{n-1} P[\sum_{k=0}^{m} Y_k \le t] \cdot P[\text{buffered at } m \text{ int. nodes}].
\end{aligned}$$

Note the $P[\sum_{k=0}^{m} Y_k \le t]$ corresponds to an Erlang distribution with parameters $\mu(1-\rho)$ and $m+1$, i.e., ERL$(\mu(1-\rho), m+1)$. This allows us to derive the probability density function of D_n as:

$$f_{D_n}(t) = \sum_{m=0}^{n-1} \begin{pmatrix} n-1 \\ m \end{pmatrix} (1-p_c)^m {p_c}^{n-1-m} \mu(1-\rho)^{m+1} \cdot \frac{t^m e^{-\mu(1-\rho)t}}{m!}.$$

Using the result

$$\int x^n e^{ax} dx = \frac{e^{ax}}{a} \sum_{k=0}^{n} \begin{pmatrix} n \\ k \end{pmatrix} (-1)^k k! \frac{x^{n-k}}{a^k}$$

and integrating $f_{D_n}(t)$ from 0 to t we get the delivery time distribution as

$$F_{D_n}(t) = \sum_{m=0}^{n-1} \begin{pmatrix} n-1 \\ m \end{pmatrix} (1-p_c)^m {p_c}^{n-1-m} \frac{[\mu(1-\rho)]^{m+1}}{m!} \cdot \left\{ \frac{m!}{[\mu(1-\rho)]^{m+1}} - \frac{e^{-\mu(1-\rho)t}}{\mu(1-\rho)} \sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} \frac{k! \cdot t^{m-k}}{[\mu(1-\rho)]^k} \right\}.$$

4 Numerical Examples and Simulation Comparison

In this section, parameters derived from the actual **HARTS** routing hardware are used to evaluate the probability distribution function for delivery times discussed in the previous section. Also presented is a comparison of the analytic results against a low-level functional simulation of the routing hardware of **HARTS**.

In contrast to the analytical model, the simulator makes very few simplifying assumptions in modeling the behavior of virtual cut-through in **HARTS**. The simulator accurately models the delivery of each message by emulating the timing of the routing hardware[2] along the route of a packet at the microcode level. Also captured are the internal bus access overheads that the packets experience if they are unable to cut-through an intermediate node. For example, when a transit packet arrives at an intermediate node, the following sequence of timed events are set into action. First, the receiver for that particular direction waits for the packet header to become available to attempt a routing decision. For the case of the H-mesh any incoming packet may have either arrived at its final destination or could be transmitted in one of possibly three directions. Second, the receiver schedules an access to an internal bus to reserve the first choice for a direction to transmit the packet. If the transmitter for this direction is free, the packet will cut-through this node with only the slight delay of waiting for the header and the single status query of the transmitter. If the first attempt to reserve the transmitter was unsuccessful, an attempt at an alternate transmitter is made, if applicable. If both of these attempts are unsuccessful, the packet is queued at this node for later transmission. Third, the receiver schedules events to signal the completion of the packet at this node. This may involve either unreserving a transmitter if the packet successfully cut-through or informing the module that simulates the handling of buffered messages. This detailed timing and tracking of messages allows different message scheduling, access protocols, and memory management strategies to be investigated. However, for the results presented in this section only a First-Come First-Serve single queue with unlimited memory was used.

In addition to the exponentially distributed packet lengths, the simulator can also use a discrete distribution of packet lengths where the user specifies the number of different types of messages, their lengths, and the probability of each type of message. Similar to the analytic model, packet arrivals are assumed to follow a Poisson arrival process.

For the examples presented in this section the following parameters were used. (Note that choice of these parameters is arbitrary and will not in general change our conclusions drawn in this section.) The dimension of the mesh was 7 resulting in 127 nodes in the system. The probability of a node communicating with a specific node k hops away was assumed to be inversely proportional to the number of hops, i.e., $q_k = \frac{1}{36k}$. The mean packet length for the analytic model was assumed to be 185.6 bytes. The distribution of packet lengths for the simulation were 64, 128, 512 bytes, each with probability 0.3, 0.5, 0.2, respectively. The results for three different packet generation rates are obtained. These correspond to 15%, 30%, 45% of the peak packet generation rate that can be supported by the routing hardware. Currently, the peak packet generation rate that can be supported by the routing hardware is 4 MBytes per second. All the distributions either generated or collected were for messages having their destination 5 hops from their source node.

Figure 4 shows a plot of the probability distribution function of the delivery times of a message traveling 5 hops in a H-mesh of dimension 7. The three curves in the figure show the variation in the probability distribution function with respect to the assumed message generation rate λ_G at each node. As would be expected, the delivery time distributions shift to the right as the load on the network is increased.

Figure 5 shows the inverse of the probability distribution functions in Figure 4. The inverse of the distribution function is useful to determine design parameters like delay bounds. For instance, one can select a delay bound such that the probability of a message being delivered within that bound is greater than a specified threshold. This would provide

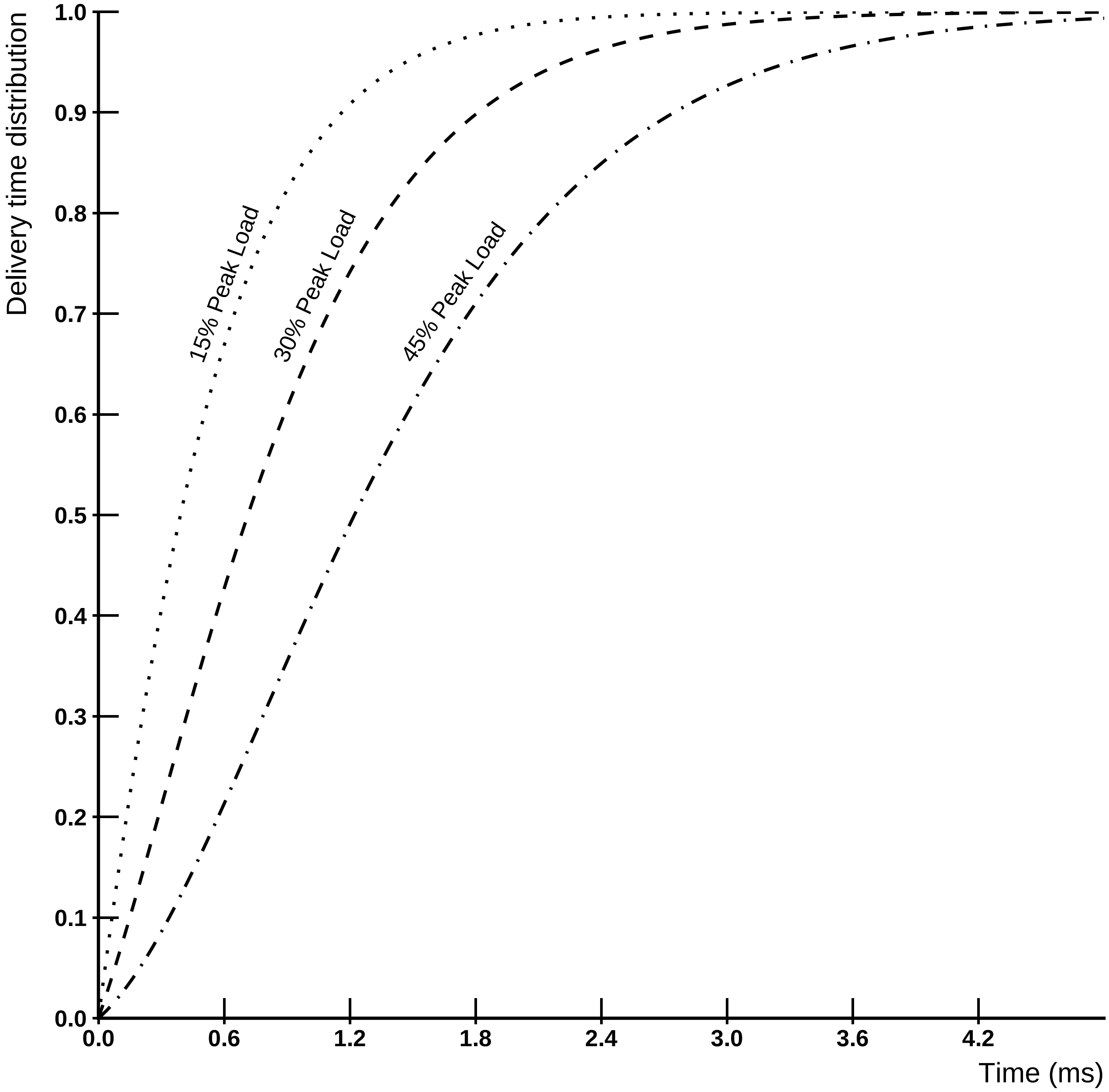


Figure 4: Probability distribution of D_5 in a H-mesh of dimension 7.

a probabilistic measure on the guarantees that can be provided in a real-time system during its operation.

Figures 6, 7, and 8 compare the analytic model against a low-level functional simulation of the routing hardware in **HARTS**. The results show that the analytic model predicts, with a reasonable accuracy, the delivery times for the loads shown. The jumps in the simulation results are due to the discrete distribution of the message length. It is found that at higher loads (greater than 65% of the peak load) the differences between the simulation and the model can be significant. Reasons for these differences are currently being investigated. Also note, the analytic model overestimates the actual delivery times and therefore the model produces a pessimistic result. The slight discrepancy at small delivery times between the model and the simulation result from the model not taking into account the overheads of processing the message headers.

5 Conclusion

The main contribution of this paper is the derivation of the distribution of message delivery times in a C-wrapped H–mesh that has virtual circuit cut-through capabilities. The techniques used in this paper can be extended to other interconnection topologies like hypercubes or rectangular meshes. The parameters p_b, T, and p_c can be calculated for a hypercube or a rectangular mesh using techniques similar to the ones in Section 3.2 because the techniques depend only on the ability to determine the fraction of minimal routes between a pair of nodes passing through a given node. Once T and p_c are determined, the derivation of the distribution for delivery times does not depend on the topology.

The distribution functions derived in this paper are essential in the design of real-time systems with deadlines. They provide a probabilistic measure on the guarantees that the system can support for message exchanges during the operation of a real-time system.

References

[1] M.-S. Chen, K. G. Shin, and D. D. Kandlur, "Addressing, routing and broadcasting in hexagonal mesh multiprocessors," *IEEE Trans. Computers*, vol. 39, no. 1, pp. 10–18, January 1990.

[2] J. W. Dolter, P. Ramanathan, and K. G. Shin, "A microprogrammable VLSI routing controller for HARTS," in *International Conference on Computer Design: VLSI in Computers*, pp. 160–163, October 1989.

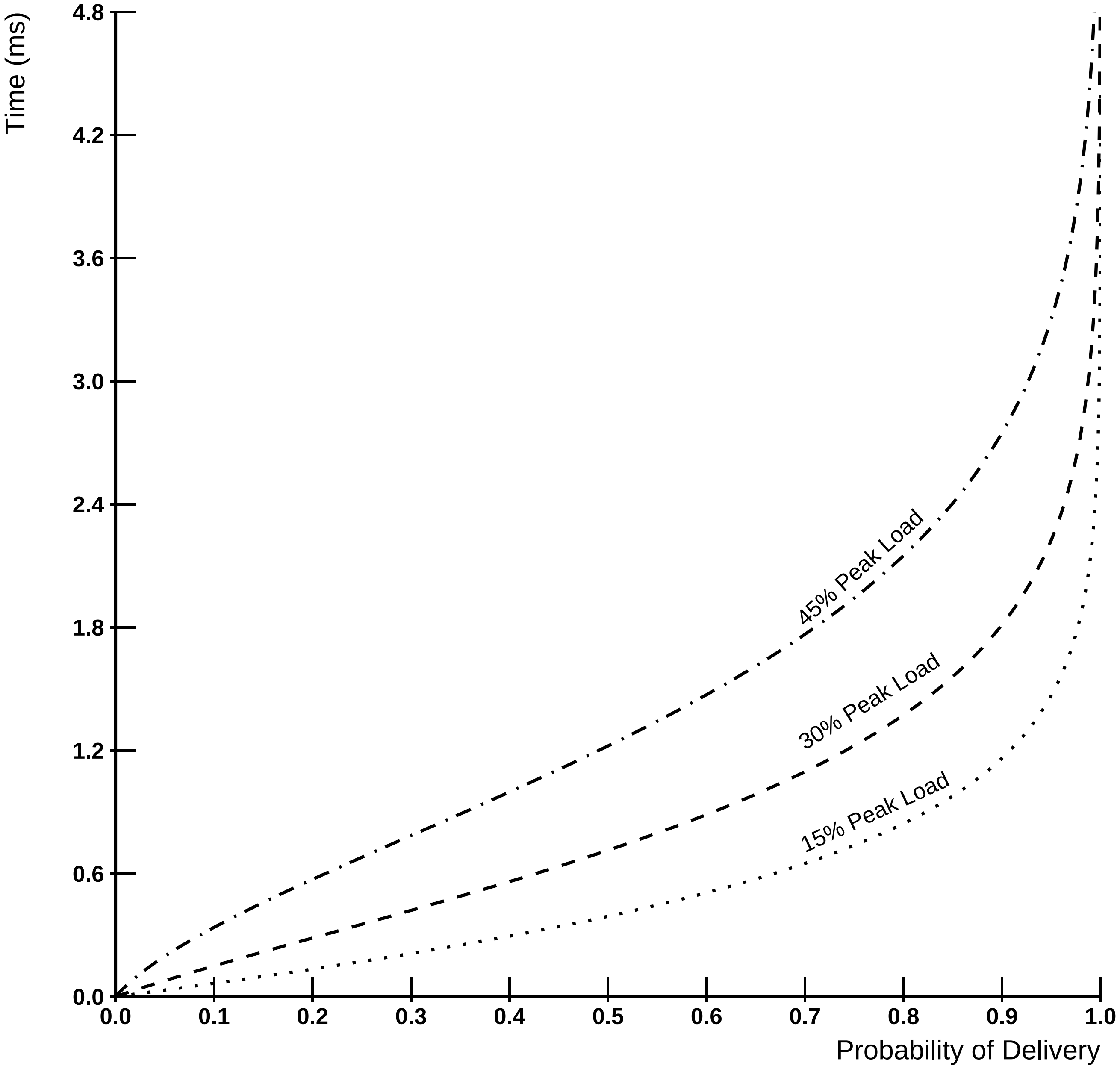


Figure 5: Delivery time vs. Probability of successful delivery.

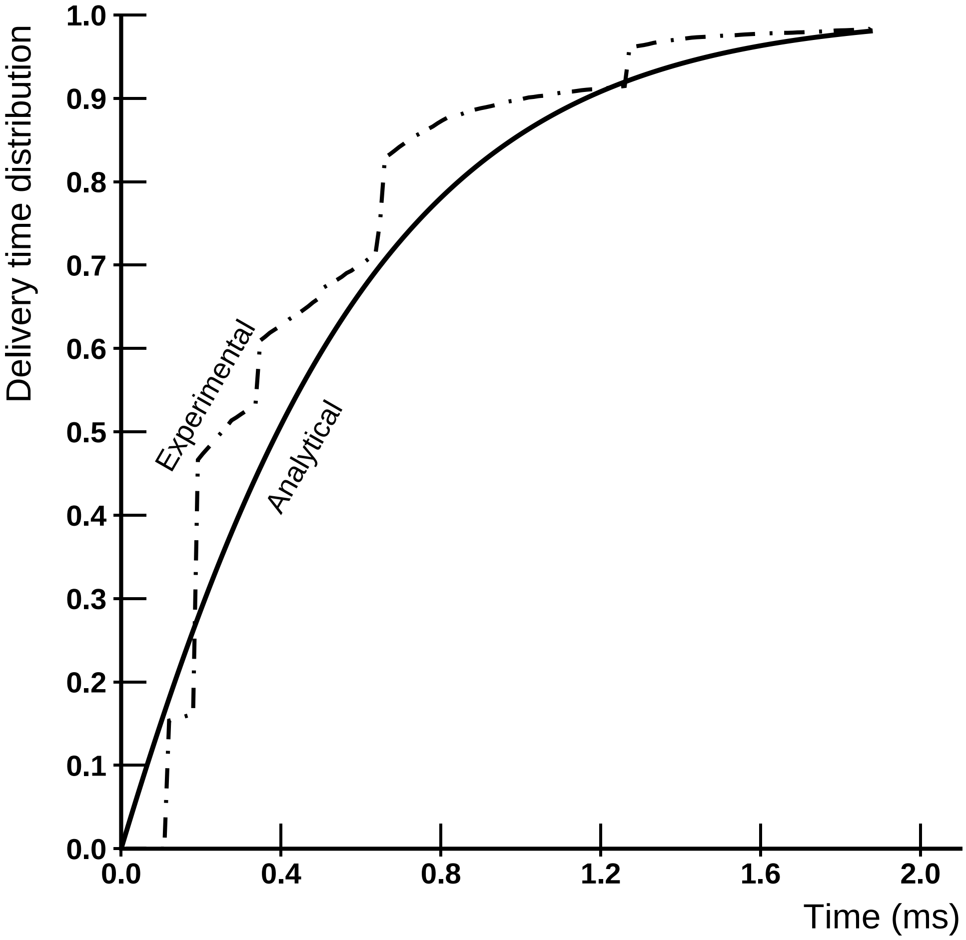


Figure 6: F_{D_5} at 15% Peak Load (H-mesh dimension 7.)

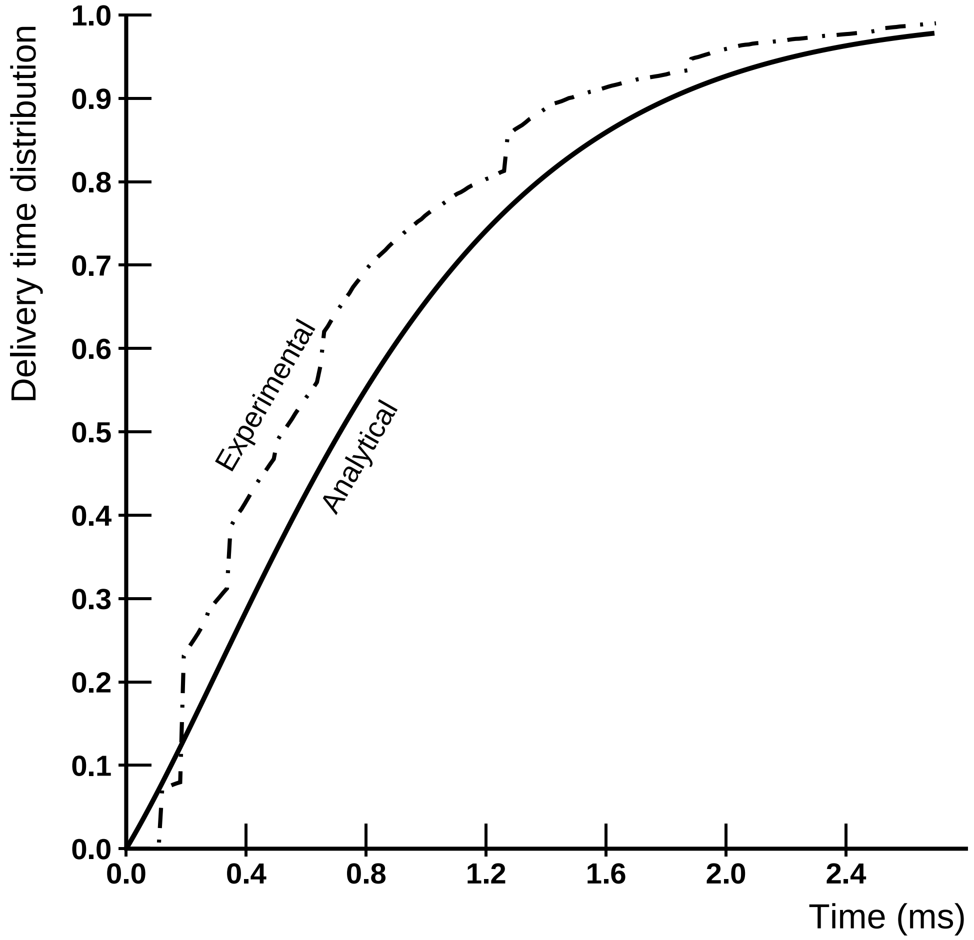


Figure 7: F_{D_5} at 30% Peak Load (H-mesh dimension 7.)

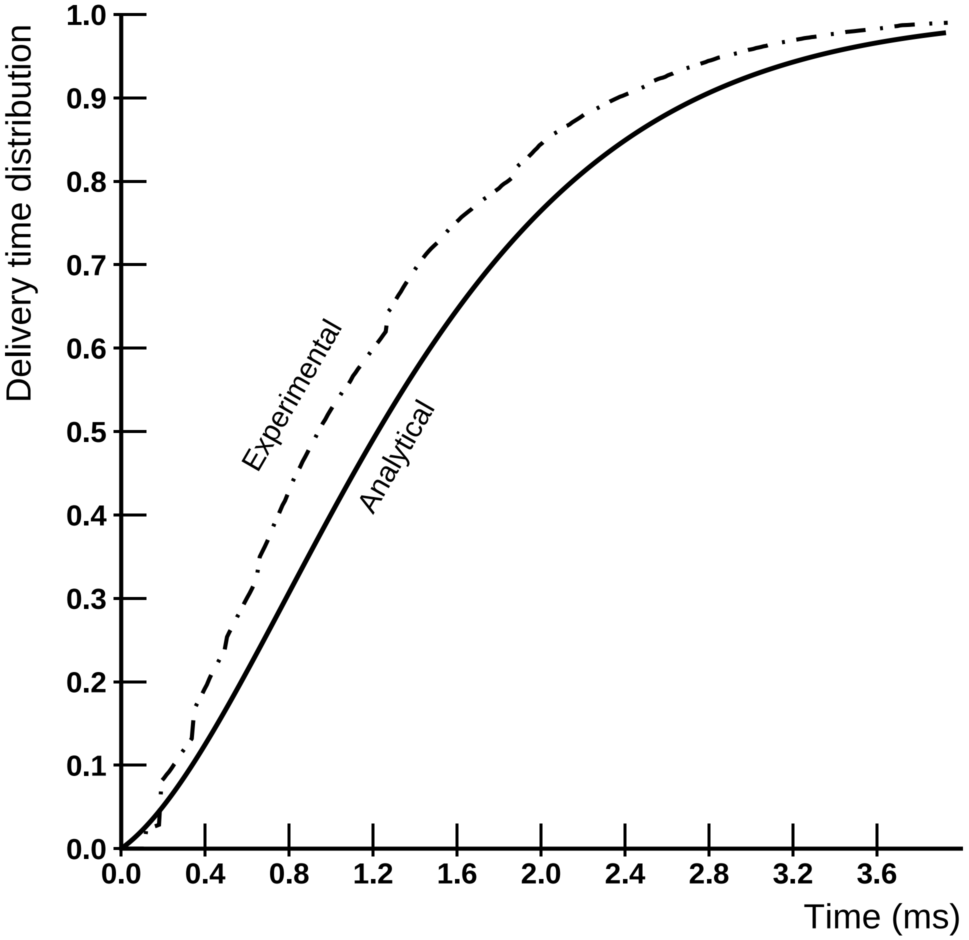


Figure 8: F_{D_5} at 45% Peak Load (H-mesh dimension 7.)

[3] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique," *Computer Networks*, vol. 3, no. 4, pp. 267–286, September 1979.

[4] L. Kleinrock, *Queueing systems*, volume I: Theory, John Wiley & Sons, 1975.

[5] A. J. Martin, "The torus: An exercise in constructing a processing surface," in *Proc. Caltech Conf. on VLSI*, pp. 527–537, 1981.

[6] K. S. Stevens, "The communication framework for a distributed ensemble architecture," AI Technical Report 47, Schlumberger Research Laboratory, February 1986.

APPENDIX A

Proof of one-to-one correspondence

Definition 7 A *pseudo-shape* corresponding to the route $n_0 \cdots n_k$ is the sequence $a_1 \cdots a_i \cdots a_k$ of directions such that $a_i = cwhm(n_{i-1}, n_i)$ for $1 \leq i \leq k$.

Note that a shape is a pseudo-shape with constraints on the permissible directions (to form minimal routes). Define an operator $\oplus$ between two directions in $\{d_0, d_1, \ldots d_5\}$ as follows.

$$d_i \oplus d_j = \begin{cases} d_{[i+1]_6} & \text{if } j = [i+2]_6 \text{ and } i \in \{0, \ldots 5\} \\ \emptyset & \text{if } j = [i+3]_6 \text{ and } i \in \{0, \ldots 5\} \\ d_{[i+5]_6} & \text{if } j = [i+4]_6 \text{ and } i \in \{0, \ldots 5\} \end{cases} .$$

The $\oplus$ operator is undefined between two directions d_i and d_j such that $j = i$ or $j = [i+1]_6$ or $j = [i+5]_6$. Intuitively, traveling first along direction d_i and then along d_j equivalent to traveling a single step along direction $d_i \oplus d_j$ if $d_i \oplus d_j$ is well-defined. For instance, traveling along d_0 and then along d_2 is equivalent to a single step along d_1.

Observation 1 A route $n_0 \cdots n_k$ can be transformed to any other route $m_0 \cdots m_{k'}$ with $k' < k$ using the following procedure:

1. Transform $n_0 \cdots n_k$ to the pseudo-shape $a_1 \cdots a_k$.
2. Transform the $a_1 \cdots a_k$ to $b_1 \cdots b_{k'}$ by a finite number of applications of the following operations:
   (a) Replace a component a_i with a component $a_{i'} \neq a_i$.
   (b) Replace any two components a_i and a_j, $i \neq j$, by $a_i \oplus a_j$, where $a_i \oplus a_j$ is well-defined.
   (c) Permute the components of $a_1 \cdots \cdots a_k$.
3. Transform the pseudo-shape $b_1 \cdots b_{k'}$ to the route $m_0 \cdots m_{k'}$.

Note that operation (2a) does not "preserve the source–destination node pair". This observation can be formally stated as follows. Let $a_1 \cdots a_k$ be the pseudo-shape associated with the route $n_0 \cdots n_k$. Let $b_1 \cdots b_k$ be a pseudo-shape obtained from $a_1 \cdots a_k$ by a single application of (2a). Also let $m_0 \cdots m_k$ be the route associated with the pseudo-shape $b_0 \cdots b_k$. Then by "preserving the source-destination pair" we mean $b_1 \cdots b_k$ is such that $m_0 = n_0$ implies $m_k = n_k$. With this definition of preserving the source-destination pair we can conclude that the operations (2b) and (2c) preserve the source–destination node pairs. Operation (2b) is the only operation that reduces the length of a pseudo-shape and the corresponding route.

Lemma 6 A route $n_0 \cdots n_k$ is a minimal route iff the associated pseudo-shape is a shape.

Proof: We will first prove that the pseudo-shape of a minimal route is a shape.

Suppose not. Then there exist components a_i and a_j in the pseudo-shape such that we can apply operation (2b) to reduce the length of the pseudo-shape. The route associated with this reduced pseudo-shape will be shorter than the assumed minimal route. A contradiction.

Now consider the reverse direction of the lemma, i.e., the route associated with a shape is minimal.

Suppose not. Then there exists a minimal route between the same source–destination pair whose pseudo-shape is shorter than the given shape. Therefore we should be able to reduce our given shape to the pseudo-shape of the minimal route using operations that preserve the source–destination pair. But this cannot happen since no operation of type (2b) can be applied to this shape. Thus our initial assumption the route associated with our shape is not minimal is false. ■

Theorem 3 There is a one-to-one correspondence between anchored shapes and anchored routes anchored at node 0.

Proof: We first show that Equation (3.2) transforms anchored shapes to anchored routes at node 0. Consider the anchored shape (s, ℓ). Construct the pair $(r, 0)$ using Equation (3.2). We show that $(r, 0)$ satisfies the three necessary properties of an anchored route.

1. Since s has a length of at least 2, the corresponding route r has a length at least 3.

2. Follows from Lemma 6 that r is a minimal route.

3. Follows directly from the construction that node 0 is contained in the route r.

We now show that Equation (3.1) transforms anchored routes at node 0 to anchored shapes. Consider the anchored route $(r, 0)$ anchored at 0. Construct the pair $(a_1 \cdots a_k, \ell)$ using Equation (3.1). By Lemma 6, $a_1 \cdots a_k$ will be a shape since the route r is minimal by definition. ℓ is bounded by construction between 1 and $k-1$ as required by the definition of an anchored shape. ■