AVERAGE-CASE “MESSY” BROADCASTING

CHENKUAN LI
Department of Mathematics and Computer Science
Brandon University
Brandon, MB, Canada, R7A 6A9
lic@BrandonU.ca

THOMAS E. HART
Department of Computer Science
University of Toronto
Toronto, ON, Canada, M5S 3G4

KEVIN J. HENRY
Department of Mathematics and Computer Science
Brandon University
Brandon, MB, Canada, R7A 6A9

IAN A. NEUFELD
Department of Mathematics and Computer Science
Brandon University
Brandon, MB, Canada, R7A 6A9

Received 27 August 2004
Revised 26 November 2006

Current studies of “messy” broadcasting have so far concentrated on finding worst-case times. However, such worst-case scenarios are extremely unlikely to occur in general. Hence, determining average-case times or tight upper bounds for completing “messy” broadcasting in various network topologies is both necessary and meaningful in practice. In this paper, we focus on seeking the average-case “messy” broadcast times of stars, paths, cycles, and $d$-ary trees, and finding good upper bounds for hypercubes. Finally, we derive a recursive formula to express the average-case time for a specific “messy” broadcast model on a complete graph using a classical occupancy problem in probability theory, and provide a nice simulation result which indicates that this model behaves like classical broadcasting.

Keywords: “Messy” broadcasting; network topology; hypercube; $d$-ary tree; average-case analysis.

1. Introduction

Broadcasting is an information dissemination problem in which we model a computer network using a graph $G = (V, E)$. The main motivation of broadcasting research is to provide a theoretical structure for general interconnection networks for multiprocessor computers. One vertex, the originator, has a message which it must distribute to all other vertices. A
vertex \( v \) may communicate with any other vertex adjacent to \( v \), and communication takes place in discrete time units. There are many variants of the broadcasting problem (classical broadcasting, \( k \)-broadcasting, reliable broadcasting, etc.); a survey of these problems is given in [10]. “Messy” broadcasting is the variant which is the subject of this paper.

In the classical broadcast model, it is assumed that every vertex in a network broadcasts optimally. Such a model therefore requires either

- a leader who knows the network’s topology and coordinates the actions of all members during the entire broadcasting process, which seems unrealistic in practice; or,

- that each vertex must have a coordinated set of protocols with respect to any originator, the space to store these protocols, and each message must be labeled with the originator and the time at which the message was sent.

“Messy” broadcasting does away with these assumptions, instead assuming that the network’s vertices have no knowledge of the network’s topology, and that when a vertex receives a message it knows neither the originator of the message, nor the time at which the message was sent. The vertices are therefore unable to devise any sophisticated broadcast scheme, but the network can be built more cheaply and easily. This variant of broadcasting was introduced by Ahlswede, Haroutunian, and Khachatrian in [1]. The worst case problems on various network topologies such as paths, cycles, hypercubes, and \( d \)-ary trees were studied by Harutyunyan and Liestman in [9]. Hart and Harutyunyan derived the “messy” broadcast times for complete bipartite graphs and improved the lower bound for “messy” broadcast times on hypercubes of arbitrary dimension in [8]. In [2], Comellas, Harutyunyan, and Liestman determined the exact values and bounds for the broadcast times on multi-dimensional directed tori.

A survey of existing methods of communication in usual networks was given by P. Fraigniaud and E. Lazard [6] which focused on the study of complete networks, the ring, the torus, the grid, the hypercube, the cube connected cycle, undirected de Bruijn graphs, the star graph, the suffle-exchange graph, and the butterfly graph with the constant model and the linear model. In [11] Hromkovic, Klasing, Monien, and Peine summarized the main techniques and results relating to broadcasting and gossiping problems using a one-way communication mode.

At each time unit every informed vertex broadcasts randomly to a neighbor which it believes is uninformed. There are three models of “messy” broadcasting, and each makes different assumptions about how a vertex \( v \) comes to believe that its neighbor \( u \) is informed:

- **Model \( M_1 \):** \( v \) believes that \( u \) is informed iff \( u \) is informed.

- **Model \( M_2 \):** \( v \) believes that \( u \) is informed iff either \( v \) has informed \( u \) or \( u \) has informed \( v \).

- **Model \( M_3 \):** \( v \) believes that \( u \) is informed iff \( v \) has informed \( u \).

Note that we assume that each informed vertex may broadcast to only one neighbor during each time unit. Assuming that vertices may broadcast to \( k \geq 2 \) neighbors in a single
time unit would yield “messy” $k$-broadcasting, a problem which, to our knowledge, has yet to be investigated. Since in our study the number of informed vertices may at most double during a single time unit, no “messy” broadcast scheme can be completed in fewer than $\lceil \log_2 n \rceil$ time units, where $n$ is the number of nodes in the network in question. We consider broadcasting to be completed when each node in the network has received the message.

We follow the notation of [1] and [9], with some additions to accommodate our average-case analysis.

Let $u \in G$ be the originator of a “messy” broadcast. We say that $\sigma(u)$ is a strategy or scheme for model $M_i$ for $i = 1, 2, 3$ if all calls in $\sigma(u)$ are legitimate under model $M_i$, and $\sigma(u)$ informs all vertices in $G$. For $i = 1, 2, 3$, let $\Omega_i(u) = \{\sigma(u) \mid \sigma(u) \text{ is legitimate under model } M_i\}$ be the set of all broadcast strategies from the originator $u$ under model $M_i$.

We define $t_i(u)$ to be the maximum time of any scheme in $\Omega_i(u)$, and we define $t_i(G) = \max \{t_i(u) \mid u \in G\}$. We also define $T_i^{\sigma(u)}_{u,G}$ to be the time taken to inform all vertices in $G$ according to the “messy” broadcast scheme $\sigma(u) \in \Omega_i(u)$. We can now define the average “messy” broadcast time of $u$ in $G$ under $M_i$ to be

$$E[T_i,u,G] = \frac{\sum_{\sigma(u) \in \Omega_i(u)} T_i^{\sigma(u)}_{u,G}}{|\Omega_i(u)|}.$$ 

Furthermore, we define the average “messy” broadcast time of $G$ under $M_i$ to be

$$E[T_i,G] = \sum_{k=1}^{V(G)} E[T_i,u_k,G] \Pr\{u_k\},$$ 

where $\Pr\{u_k\}$ is the probability of $u_k$ being the originator of a “messy” broadcast in $G$. Of course, these probabilities must be chosen such that $\sum_{k=1}^{V(G)} \Pr\{u_k\} = 1$. Throughout this paper, we will assume that $\Pr\{u\} = \frac{1}{|G|}$ for all $u \in G$; however, allowing the probabilities to be chosen in other ways gives us the opportunity generalize the model by choosing a high probability for an originator that can distribute a message quickly over a network topology, thereby speeding up the average broadcast time.

In [9], Harutyunyan and Liestman investigated the worst-case times to complete “messy” broadcasting under $M_1$, $M_2$ and $M_3$ in hypercubes, complete $d$-ary trees, and some simple graphs. These times are highly improbable for almost all popular topologies, and the probability for these worst-case times to occur is almost zero when the number of vertices in a network is large. The objective of this paper is to open a new direction of research in the area of “messy” broadcasting - the study of average case “messy” broadcast times, which certainly play an important role in determining the efficiency of networks. In sections 2 and 4 we focus on finding exact values for the average case “messy” broadcast times on stars, paths, cycles and complete $d$-ary trees. In section 4 we investigate the hypercube structure and provide tight upper bounds on “messy” broadcast times under models $M_2$ and $M_3$, which are far better than the worst-case times. As an example to illustrate difference between worst-case and average-case times, Theorem 3.1 of [9] states that the worst-case time to finish “messy” broadcasting under model $M_3$ on a $d$-dimensional hypercube is $d(d+1)/2$, and we estimate that the probability of randomly-chosen “messy” broadcast scheme taking
Fig. 1. A star graph on 9 vertices.

Fig. 2. Messy broadcasting in star graphs under models $M_1$ and $M_2$. We adopt the convention of drawing the originator in black.

this many time units is far less than $1/d^4$, which vanishes very quickly when $d$ is increasing. However, the average-case time we find in Section 3 is significantly better and is bounded by

$$(d + 1) \sum_{j=1}^{d} \frac{1}{j} = \Theta(d \log d).$$

In addition, we believe that the average “messy” broadcast time for a complete graph is $[\log n] + 2$ (see Section 5), which is much faster than the worst-case time of $n - 1$, where $n$ is the total number of vertices. Our intuition is guided by simulations we performed for several graphs under all three models, which we validate by the proofs presented in this paper.

2. Simple Graphs

In order to demonstrate the techniques and considerations required when examining average-case “messy” broadcasting times, we first look at the analysis of some simple graphs. For these simple graphs, we can calculate average-case “messy” broadcast times in a very straightforward manner.

2.1. Stars

The star $S_n$ has $n$ vertices. One of these vertices (the central vertex) has degree $n - 1$, and the others have degree 1. $S_9$, the star on 9 vertices, is shown in Fig. 1. Stars are particularly simple to analyze, since the number of possible “messy” broadcast schemes is very small.

**Theorem 2.1.** $E[T_1,S_n] = E[T_2,S_n] = n - 1$, and $E[T_3,S_n] = n - 1 + \frac{n-2}{n}$.

**Proof.** Let $u$ denote the central vertex of $S_n$, and let the other vertices of $S_n$ be enumerated $v_1, v_2, \ldots, v_{n-1}$.
For models $M_1$ and $M_2$, all legitimate broadcast schemes obviously complete in exactly $n-1$ time units, as can be confirmed by examining Fig. 2. Hence, $E[T_{1,S_n}] = E[T_{2,S_n}] = n-1$.

The analysis is slightly more complicated for model $M_3$. If the originator is $u$, then broadcasting must complete in $n-1$ time units, since $u$ must call its $n-1$ neighbors at times $1, 2, \ldots, n-1$. If the originator is $v_i$ for some $i$, then we must consider two cases:

**Case 1:** Once $u$ is informed, it broadcasts to vertices $v_j$ ($j \neq i$) at times $2, 3, \ldots, n-1$. Broadcasting therefore completes in $n-1$ time units since all nodes are informed.

**Case 2:** Once $u$ is informed, it broadcasts to $v_i$ at time $j$, for some $j$ in the range $2 \leq j \leq n-1$. Broadcasting therefore completes in $n$ time units.

Given that $v_i$ is the originator, the probability that Case 1 occurs is the probability that $u$ does not inform $v_i$ at time $2, 3, \ldots, n-1$. Given that $u$ has not informed $v_i$ at time $t$, the probability of this occurring at time $t$ is $\frac{n-t}{n-1}$. Thus, the probability that Case 1 occurs is given by

$$\frac{n-2}{n-1} \times \frac{n-3}{n-2} \times \frac{n-4}{n-3} \times \cdots \times \frac{2}{3} \times \frac{1}{2} = \frac{1}{n-1}.$$ 

Hence,

$$E[T_{3,v_i,S_n}] = Pr\{\text{Case 1}\} \times (n-1) + Pr\{\text{Case 2}\} \times n = n - \frac{n-2}{n}.$$ 

For the star $S_n$, we thus have, by Eq. (1.1), that,

$$E[T_{3,S_n}] = \frac{E[T_{3,S_n}] + \sum_{i=1}^{n-1} E[T_{3,v_i,S_n}]}{n} = n - 1 + \frac{n-2}{n}.$$ 

Note that this last result could be simplified to $n - \frac{n}{2}$, however, we leave it as $n - 1 + \frac{n-2}{n}$ to show more clearly the base cost of $n - 1$ plus the cost of $\frac{n-2}{n}$ due to poor choices made while broadcasting.

### 2.2. Paths

A path $P_n$ is a graph on $n$ vertices labeled $v_0, v_1, \ldots, v_{n-1}$ where vertex $v_k$ is connected to vertex $v_{k+1}$ for $k = 0, 1, \ldots, n-2$. Paths are only marginally more complex to analyze than the stars.

**Theorem 2.2.** $E[T_{i,P_1}] = 0$ and $E[T_{i,P_2}] = 1$ for $i = 1, 2, 3$, $E[T_{1,P_3}] = E[T_{2,P_3}] = 2$, and $E[T_{3,P_3}] = \frac{7}{2}$. For $n \geq 4$,

$$E[T_{1,P_n}] = E[T_{2,P_n}] = \begin{cases} \frac{3}{4}n - \frac{1}{n} & \text{if } n \text{ is even} \\ \frac{3}{4}n - \frac{3}{4n} & \text{if } n \text{ is odd, and} \end{cases}$$

$$E[T_{3,P_n}] = \begin{cases} \frac{2}{3}n - \frac{3}{4} - \frac{1}{n} & \text{if } n \text{ is even} \\ \frac{2}{3}n - \frac{3}{4} - \frac{7}{8n} & \text{if } n \text{ is odd.} \end{cases}$$
Proof. The results for \( n < 4 \) are easy to verify. To prove \( E[T_{1,v_0}] = \frac{7}{3} \) we must consider two cases. The first, when \( v_1 \) is the originator, is the simplest. At times \( k = 1, 2, v_1 \) sends the message to \( v_0 \) and \( v_2 \) with broadcasting completing in 2 time units. This case occurs with probability \( \frac{1}{3} \) as each vertex is equally likely to begin broadcasting. The second case, when \( v_0 \) or \( v_2 \) originates can proceed in two possible ways depending on whether or not \( v_1 \) sends the message back to the originator before passing it to the last vertex. If \( v_1 \) sends back to the originator then broadcasting completes in 3 time units. Otherwise broadcasting completes in 2 time units. Each of these cases occur with equal probability making the average time to complete 2.5 time units. Combining these times yields

\[
E[T_{1,v_0}] = 2 \times \frac{1}{3} + 2.5 \times \frac{2}{3} = \frac{7}{3}.
\]

We now consider the general case. In \( P_n \), if a vertex \( v_k \) is informed then the only way for an adjacent vertex to be informed is if it was the vertex that informed \( v_k \), or \( v_k \) was the vertex that informed it. Thus, \( \Omega_1(u) = \Omega_2(u) \) for all \( u \in P_n \) for all \( n \); that is, models \( M_1 \) and \( M_2 \) are equivalent for paths. We will consider these models first.

Consider the case in which \( v_0 \) is the originator. Broadcasting must then complete in \( n - 1 \) time units; similarly for \( v_{n-1} \). If \( v_1 \) is the originator, then broadcasting may complete in either \( n - 1 \) or \( n - 2 \) time units, depending on whether \( v_1 \) broadcasts to \( v_0 \) or \( v_2 \) at time 1. Hence \( E[T_{1,v_1,P_n}] = n - 2 + \frac{1}{2} \). If \( v_1 \) is the originator for \( i \leq \frac{n}{2} \), then broadcasting completes in either \( n - i \) or \( n - (i + 1) \) time units depending on whether \( v_i \) broadcasts to \( v_{i-1} \) or \( v_{i+1} \) at time 1. Because each case occurs with equal probability, \( E[T_{1,v_i,P_n}] = n - (i + 1) + \frac{1}{2} \).

The case for \( i > \frac{n}{2} \) is symmetric.

If \( n \) is even, then

\[
\sum_{i=0}^{n-1} E[T_{1,v_i,P_n}] = 2 \sum_{i=0}^{\frac{n}{2}-1} E[T_{1,v_i,P_n}]
= 2((n - 1) + (n - 2 + \frac{1}{2}) + (n - 3 + \frac{1}{2}) + \cdots + (n - \frac{n}{2} + \frac{1}{2}))
= \frac{3}{4}n^2 - 1.
\]

Hence, for even \( n \),

\[
E[T_{1,P_n}] = \frac{\sum_{i=0}^{n-1} E[T_{1,v_i,P_n}]}{n} = \frac{\frac{3}{4}n^2 - 1}{n} = \frac{3}{4}n - \frac{1}{n}.
\]

The result follows similarly for odd \( n \).

We now turn our attention to model \( M_3 \), first considering vertex \( v_0 \). There are \( n - 1 \) edges between \( v_0 \) and \( v_{n-1} \). Hence, when \( v_0 \) is the originator, it needs at least \( n - 1 \) time units in order to inform all vertices in \( P_n \). Each of the \( n - 2 \) vertices \( v_1, v_2, \ldots, v_{n-2} \) may, upon being informed, broadcast either in the direction of \( v_0 \) or \( v_{n-1} \); that is, for \( 1 \leq i \leq n - 2 \), vertex \( v_i \) may broadcast first to \( v_{i-1} \), or to \( v_{i+1} \), and either case occurs with probability \( \frac{1}{2} \). For each vertex \( v_i \) that broadcasts first in the direction of \( v_{i-1} \), 1 time unit is wasted. Hence,

\[
E[T_{3,v_0,P_n}] = n - 1 + \frac{n - 2}{2}.
\]
It follows by symmetry that $E[T_{3,v_0,P_n}] = E[T_{3,v_{n-1},P_n}]$.

Vertex $v_1$ has 1 vertex on one side, and $n-2$ vertices on the other. If $v_1$ is the originator, and it broadcasts first to $v_0$, then it wastes 1 time unit. By an analysis similar to the case in which $v_0$ is the originator,

$$E[T_{3,v_1,P_n}] = n - 2 + \frac{n - 3}{2} + \frac{1}{2} = E[T_{3,v_{n-2},P_n}].$$

From our analysis of models $M_1$ we have $E[T_{1,v_i,P_n}] = n - (i + 1) + \frac{1}{2}$. We can modify this expression for $M_3$ by taking into account that after time 1, the average time needed to forward the message is now 1.5 due to possibly sending the message back where it came from. Thus,

$$E[T_{3,v_i,P_n}] = n - (i + 1) + \frac{n - (i + 2)}{2} + \frac{1}{2}.$$  

Hence, for even $n$,

$$\sum_{i=0}^{n-1} E[T_{3,v_i,P_n}] = 2((n-1 + \frac{n-2}{2}) + (n-2 + \frac{n-3}{2} + \frac{1}{2}) + \cdots + (n - \frac{n}{2} + \frac{n-(\frac{n+1}{2}) + \frac{1}{2})) = n + 1.$$  

For odd $n$, vertex $v_{\frac{n-1}{2}}$ has $\frac{n-1}{2}$ vertices on either side. Hence,

$$E[T_{3,v_{\frac{n-1}{2}},P_n}] = (\frac{n+1}{2} - 1) + \frac{\frac{n+1}{2} - 2}{2} + 1 = \frac{3n-1}{4},$$

and we have

$$\sum_{i=0}^{n-1} E[T_{3,v_i,P_n}] = 2((n-1 + \frac{n-2}{2}) + (n-2 + \frac{n-3}{2} + \frac{1}{2}) + \cdots + (n - \frac{n+1}{2} + \frac{n-(\frac{n+1}{2} + \frac{1}{2})) + \frac{1}{2}(n+1) - 1.$$  

Simple calculations imply that

$$E[T_{3,P_n}] = \begin{cases} \frac{9}{8} n - \frac{3}{4} - \frac{1}{n} & \text{if } n \text{ is even} \\ \frac{9}{8} n - \frac{3}{4} - \frac{7}{8n} & \text{if } n \text{ is odd.} \end{cases}$$

This completes our proof. \[\square\]

Paths also give us an opportunity to demonstrate very easily that many worst-case “messy” broadcast times are highly improbable. It was proven in [9] that $t_3(P_n) = 2n - 3$, but this worst-case time occurs only when the originator is $v_0$, and, for $1 \leq i \leq n-2$, $v_i$ broadcasts to $v_{i-1}$ before broadcasting to $v_{i+1}$, or, by symmetry, when the originator is $v_{n-1}$ and, for $1 \leq i \leq n-2$, $v_i$ broadcasts to $v_{i+1}$ before broadcasting to $v_{i-1}$.

**Corollary 2.1.** $P_r(T_{3,v_0,P_n} = 2n - 3) = Pr(T_{3,v_{n-1},P_n} = 2n - 3) = \frac{1}{2^{n-1}}$.

**Proof.** Assume, without loss of generality, that $v_0$ is the originator. Each vertex $v_i$, $1 \leq i \leq n-2$, broadcasts to $v_{i-1}$ before broadcasting to $v_{i+1}$ with probability $\frac{1}{2}$, and these are independent events. The worst-case time of $2n - 3$ occurs only when each $v_i$ broadcasts first to $v_{i-1}$. There are $n-2$ vertices between $v_0$ and $v_{n-1}$ that must first send the message
back to \( v_{i-1} \), each doing so with probability \( \frac{1}{2} \). Hence, the probability of the worst-case time occurring is \( \frac{1}{2^n} \).

\[ \square \]

### 2.3. Cycles

A cycle \( C_n \) on \( n \) vertices is a path on \( n \) vertices with an additional edge added between \( v_0 \) and \( v_{n-1} \). Fig. 3 shows the cycles \( C_6 \) and \( C_7 \).

**Theorem 2.3.** Let \( C_n \) be a cycle on \( n \) vertices. Then,

\[
E[T_{1,C_n}] = E[T_{2,C_n}] = \left\lceil \frac{n}{2} \right\rceil, \quad \text{and}
\]

\[
E[T_{3,C_n}] = \begin{cases} 
\frac{3n-2}{4} & \text{if } n \text{ is even} \\
\frac{3n-1}{4} & \text{if } n \text{ is odd}.
\end{cases}
\]

**Proof.** Without loss of generality, assume that \( v_0 \) is the originator and that \( v_0 \) informs \( v_1 \) at time \( k = 1 \) and \( v_{n-1} \) at time \( k = 2 \). In models \( M_1 \) and \( M_2 \), at time \( k \geq 3 \) the message is passed from from vertex \( v_{k-1} \) to \( v_k \) and from vertex \( v_{n-k+2} \) to \( v_{n-k+1} \). Thus, at time \( \left\lceil \frac{n}{2} \right\rceil \) all vertices on the path \( v_0, v_1, \ldots, v_{\left\lceil \frac{n}{2} \right\rceil} \) are informed as well as all vertices on the path \( v_0, v_{n-1}, \ldots, v_{\frac{n}{2}-1} \). Hence, all vertices are informed and broadcasting is completed at time \( \left\lceil \frac{n}{2} \right\rceil \).

The results for model \( M_3 \) can be derived similarly, taking into account that the average time to pass a message from \( v_k \) to \( v_{k+1} \) is 1.5 due to the potential time spent sending the message back to \( v_{k-1} \). \( \square \)

### 3. Hypercubes

Let \( Q_d \) denote the binary hypercube of dimension \( d \). We define hypercubes as follows: The graph \( Q_d \) has \( 2^d \) vertices. These vertices are labeled with binary strings of length \( d \). Two vertices are connected if their labels differ by exactly one bit. For example, \((1000 \ldots 01)\) and \((0000 \ldots 01)\) are connected, but \((1010 \ldots 01)\) and \((0000 \ldots 01)\) are not. We will, in general, denote the label of a vertex \( u \) by \((u_1 u_2 \ldots u_d)\), where \( u_i \in \{0,1\} \) for \( i = 1, 2, \ldots, d \), and the edge between vertices \( u \) and \( v \) by \((u,v)\).
Hypercubes are interesting for a number of reasons. They are not only of theoretical interest, but are a popular topology for real-world interconnection networks. Hypercubes also have optimal performance under the classical broadcasting model; in fact, hypercubes are minimum broadcast graphs [3] and [6].

Under “messy” broadcasting, however, the worst case performance of hypercubes is $\Theta(d^2)$ for models $M_2$ and $M_3$, and the worst case time for model $M_1$ is between $\Theta(d)$ and $\Theta(d^2)$ [9]. To our knowledge, finding the exact value of $t_1(Q_d)$ remains an open problem. In the following, we focus on tightly estimating $E[T_{3,Q_d}]$ and $E[T_{2,Q_d}]$; these average bounds are good benchmarks for the performance of hypercubes under “messy” broadcasting. Finding or approximating $E[T_{1,Q_d}]$ is future work.

We denote the $d + 1$ levels of $Q_d$ by $Q^0, Q^1, \ldots, Q^d$, where $Q^i$ is the set of all vertices $u \in Q_d$ such that $u$’s label contains exactly $i$ 1’s (see Fig. 4). For example, (1010 . . . 0) $\in Q^2$ and (1111 . . . 1) $\in Q^d$. The number of binary strings in $Q^i$ is \( \binom{d}{i} = \frac{d!}{i!(d-i)!} \); in particular, $Q^0 = \{(000 . . . 0)\}$ and $Q^d = \{(111 . . . 1)\}$.

Before moving further we must establish the following identity.

**Lemma 3.1.** If $d$ and $i$ are both integers and $d > i$ then

$$
\sum_{m=0}^{i-1} \frac{(d - i + m - 1)!}{m!} = \frac{i(d-1)!}{(d-i)!}.
$$
Proof. We proceed by induction on $i$. Clearly, when $i = 1$ we have
\[
\sum_{m=0}^{i-1} \frac{(d-i+m-1)!}{m!} = \frac{(d-1+0-1)!}{0!} = (d-2)!
\]
and
\[
\frac{i(d-1)!}{(d-i)!} = \frac{1(d-1)!}{(d-1)!} = (d-2)!.
\]
Hence, the identity holds for $i = 1$. Assuming the identity holds for $i$ we must now show that it also holds for $i+1$, i.e.,
\[
\sum_{m=0}^{(i+1)-1} \frac{(d-(i+1)+m-1)!}{m!} = \frac{(i+1)(d-1)!}{(d-(i+1))(i+1)!}.
\]
Substituting $d = x + 1$ into the left hand side and splitting off the $m = i$ term yields
\[
\sum_{m=0}^{i-1} \frac{(x-i+m-1)!}{m!} + \frac{(x-1)!}{i!}.
\]
Applying our inductive hypothesis to the sum, this simplifies to
\[
\frac{i(x-1)!}{(x-i)!} + \frac{(x-1)!}{i!}.
\]
Substituting $x = d - 1$ back into the above leaves us with
\[
\frac{i(x-1)!}{(x-i)!} + \frac{(x-1)!}{i!} = \frac{i(d-2)!}{(d-i-1)!} + \frac{(d-2)!}{i!}
\]
which simplifies to
\[
\frac{(d-2)!(d-1)}{(d-i-1)!} = \frac{(i+1)(d-1)!}{(d-(i+1))(i+1)!}.
\]
Hence, the identity holds for all $i \geq 1$ and the result is proved.

In order to analyze broadcasting in the hypercube we must first introduce the idea of the initial path of a broadcast. If vertex $v_0 = (00\ldots0)$ originates the broadcast it will inform some vertex $v_1$ in level one at time 1. We define $v_2$ to be the first vertex in level two that is informed by $v_1$. In general, we define $v_i$ to be the first node in level $i$ that is informed by $v_{i-1}$. The path continues until we reach vertex $v_d = (11\ldots1)$. The sequence of vertices $v_0, v_1, \ldots, v_d$ is defined as the initial path. It is important to note that other paths the broadcast follows may propagate through the levels of the hypercube at different speeds than the initial path.

Lemma 3.2. Let $X_i$ be the random variable denoting the amount of time needed to pass a message from a vertex in level $i$ to a vertex in level $i+1$ for $i = 0, 1, 2, \ldots, d-1$ on the initial path. Then,
\[
E[X_i] = \frac{d+1}{d-i+1}.
\]
**Proof.** Using the expected value formula, we first calculate \(E[X_1], E[X_2],\) and \(E[X_3]\) as follows:

\[
E[X_1] = 1 \cdot Pr\{X_1 = 1\} + 2 \cdot Pr\{X_1 = 2\}
= d - 1 \cdot d + 2 \cdot \left( \frac{1}{d} \cdot \frac{d - 1}{d} \right)
= \frac{d + 1}{d},
\]

\[
E[X_2] = 1 \cdot Pr\{X_2 = 1\} + 2 \cdot Pr\{X_2 = 2\} + 3 \cdot Pr\{X_2 = 3\}
= d - 1 + 2 \cdot \left( \frac{2}{d} \cdot \frac{d - 2}{d - 1} \right) + 3 \cdot \left( \frac{1}{d - 1} \cdot \frac{d - 2}{d - 2} \right)
= \frac{d + 1}{d - 1},
\]

\[
E[X_3] = 1 \cdot Pr\{X_3 = 1\} + 2 \cdot Pr\{X_3 = 2\} + 3 \cdot Pr\{X_3 = 3\} + 4 \cdot Pr\{X_3 = 4\}
= d - 1 + 2 \cdot \left( \frac{3}{d} \cdot \frac{d - 3}{d - 1} \right) + 3 \cdot \left( \frac{2}{d - 1} \cdot \frac{d - 3}{d - 2} \right)
+ 4 \cdot \left( \frac{1}{d - 2} \cdot \frac{d - 3}{d - 3} \right)
= \frac{d + 1}{d - 2},
\]

Continuing in this fashion we arrive at:

\[
E[X_i] = \frac{d - i}{d} + \sum_{j=2}^{i+1} \left( j \cdot \frac{(j - 2) \cdot \frac{i - k}{d - k}}{d - (j + 1)} \right).
\]

Obviously,

\[
\prod_{k=0}^{j-2} \frac{i - k}{d - k} = \frac{i! (d - j + 1)!}{d! (i - j + 1)!}.
\]

Hence, we have

\[
E[X_i] = \frac{d - i}{d} + \frac{i! (d - i)}{d!} \sum_{j=2}^{i+1} \frac{j (d - j)!}{(i - j + 1)!}.
\]

Next, we make the substitution \(m = i - j + 1\)

\[
E[X_i] = \frac{d - i}{d} + \frac{i! (d - i)}{d!} \sum_{m=0}^{i-1} \frac{((i + 1) - m) (d - (i + m - 1))!}{m!}.
\]

Clearly,

\[
E[X_i] = \frac{d - i}{d} + \frac{i! (d - i)}{d!} \left( (i + 1) \sum_{m=0}^{i-1} \frac{(d - i + m - 1)!}{m!} - \sum_{m=1}^{i-1} \frac{(d - i + m - 1)!}{(m - 1)!} \right).
\]
Applying Lemma 3.1 to the left sum we can simplify the above to

\[
E[X_i] = \frac{d - i}{d} + \frac{i!(d - i)}{d!} \left( (i + 1) \frac{i(d - 1)!}{(d - i)!} - \sum_{m=1}^{i-1} \frac{(d - i + m - 1)!}{(m - 1)!} \right). 
\]

Next, we re-index the remaining sum so that it starts at \( m = 0 \) and make the substitution \( d = x - 1 \)

\[
E[X_i] = \frac{d - i}{d} + \frac{i!(d - i)}{d!} \left( (i + 1) \frac{i(d - 1)!}{(d - i)!} - \sum_{m=0}^{i-2} \frac{(x - i + m - 1)!}{m!} \right). 
\]

Using the fact that

\[
\sum_{m=0}^{i-2} \frac{(x - i + m - 1)!}{m!} = \sum_{m=0}^{i-1} \frac{(x - i + m - 1)!}{m!} - \frac{(x - 2)!}{(i - 1)!} 
\]

we can apply Lemma 3.1 to the above to get

\[
E[X_i] = \frac{d - i}{d} + \frac{i!(d - i)}{d!} \left( (i + 1) \frac{i(d - 1)!}{(d - i)!} - \frac{(x - 1)!}{(x - i)!} + \frac{(x - 2)!}{(i - 1)!} \right). 
\]

Substituting \( x = d + 1 \) back into the equation yields

\[
E[X_i] = \frac{d - i}{d} + \frac{i!(d - i)}{d!} \left( (i + 1) \frac{i(d - 1)!}{(d - i)!} - \frac{d!i}{(d - i + 1)!} + \frac{(d - 1)!}{(i - 1)!} \right) 
\]

which can be simplified to

\[
E[X_i] = \frac{d + 1}{d - i + 1}. 
\]

This completes the proof. \( \square \)

**Theorem 3.1.** \( E[T_{3,Qd}] \leq (d + 1) \sum_{j=1}^{d} \frac{1}{j} = \Theta(d \log d). \)

**Proof.** Using the symmetry of hypercubes, we assume, without loss of generality, that \( v_0 = (00 \ldots 0) \) is the originator. We first show that the average number of time units necessary for \((11 \ldots 1)\) to be informed is less than or equal to \(1 + (d + 1) \sum_{j=2}^{d} \frac{1}{j}\). One thing we should point out, is that broadcasting does not complete in general even if the furthest vertex \((11 \ldots 1)\) from the originator has been informed.

Let \( X \) be the random variable denoting the total amount of time spent to inform vertex \((11 \ldots 1)\) from the originator on the initial path. Then \( X = X_0 + X_1 + \cdots + X_{d-1} \) and the expected value of \( X \) is

\[
E[X] = E[X_0] + E[X_1] + E[X_2] + \cdots + E[X_{d-1}] 
\]

\[
= 1 + \frac{d + 1}{d} + \frac{d + 1}{d - 1} + \cdots + \frac{d + 1}{2} 
\]

\[
= 1 + (d + 1) \sum_{j=2}^{d} \frac{1}{j}. 
\]
Since we are tracing only one of many paths from the originator to vertex \((11\ldots 1)\), we must take into account that on average some of the other paths will be faster than our initial route. Therefore, the average number of time units to inform vertex \((11\ldots 1)\) is less than or equal to \(1 + (d + 1) \sum_{j=2}^{d} \frac{1}{j}\), which is only the average speed of the initial route.

Because broadcasting is not necessarily completed by time \(1 + (d + 1) \sum_{j=2}^{d} \frac{1}{j}\) when vertex \((11\ldots 1)\) is informed by our initial path, we allow an additional \(d\) time units to pass so that every vertex in \(Q_d\) can complete broadcasting to its neighbors in the levels above and below it. Each vertex in \(Q_d\) is connected to \(d\) other vertices, so these \(d\) additional time units ensure that each informed vertex can broadcast to each of its \(d\) neighbors. Since we have already formed an initial path, one or more vertices (all, or nearly all in the upper levels) are informed in each level, so vertices in all levels are broadcasting simultaneously to their \(d\) neighbors during these \(d\) time units ensuring that broadcasting does indeed complete. Therefore, the average-speed of this route is an upper bound for the average-case broadcasting time. This completes the proof of Theorem 3.1.

We must admit that the above result is only an upper bound of average-case broadcasting time and our simulation of \(M_3\) supports an even stronger conclusion of \(\Theta(d)\), rather than \(\Theta(d \log d)\). Unfortunately, we are unable to provide an actual proof at this moment.

**Theorem 3.2.** \(E[T_{2,Q_d}] \leq d \sum_{j=1}^{d} \frac{1}{j} = \Theta(d \log d)\).

**Proof.** Mimicking the proof of Theorem 3.1, we note that under model \(M_2\), no vertex ever informs the vertex (or vertices) that informed it. Thus, the average amount of time needed to pass a message from a vertex on level \(i\) to a vertex on level \(i+1\) is \(\frac{d}{d+1}\). Hence, the average-case time to form an initial path from the originator to vertex \((11\ldots 1)\) is less than or equal to \(1 + d \sum_{j=2}^{d} \frac{1}{j} = \Theta(d \log d)\). We allow \(d - 1\) additional time units to pass, rather than \(d\), accounting for the fact that no vertex will ever call back to the one that informed it. This extra time ensures that the broadcast completes after the initial path is formed.

Unfortunately, it seems impossible to deal with \(M_1\) in the same manner due to difficulties of computing expected values of \(E[X_i]\) on any initial path, because some vertices have been informed at level of \(i - 1\) with unknown probabilities.

Feige, Peleg, Raghavan and Upfal [4] found high-probability bounds for broadcasting in a hypercube under a slightly different model: At each step, every vertex that knows of the message chooses one of its neighbors in \(Q_d\) uniformly at random, and informs it of the message. Let \(H_d\) be the number of time units required for broadcasting to complete. Then \(H_d = \Theta(d)\) is true with probability \(1 - 1/2^d\).

4. Complete \(d\)-ary Trees

According to [9], a full \(d\)-ary tree can be defined recursively as follows: A full \(d\)-ary tree is either a single vertex (the root), or it consists of a distinguished vertex (the root) with \(d\)
children, each of which is the root of a full $d$-ary tree. A complete $d$-ary tree is a full $d$-ary tree such that every leaf is at the same distance ($h-1$) from the root (the height of such a $d$-ary tree is $h$). We use $T_{d,h}$ to denote the complete $d$-ary tree of height $h$ and define the depth of a node to be its distance from the root. The ternary tree $T_{3,3}$ is shown in Fig. 5.

**Theorem 4.1.** Let $u$ be the originator in the complete $d$-ary tree $T_{d,h}$. Then

$$E[T_{3,u,T_{d,h}}] = \begin{cases} 
    d + \frac{d^2 + 2d}{d+1} (h-2) & \text{if } u \text{ is the root,} \\
    1 + d - \frac{1}{d} + \frac{(d+2)(3d+1)}{2(d+1)} (h-2) & \text{if } u \text{ is a leaf,} \\
    \frac{d+2}{2i} + d - \frac{1}{d} + \frac{d^2 + 2d}{d+1} (h-2) & \text{if } u \text{ is at } i \text{th depth for } i = 1, 2, \ldots, h-2.
\end{cases}$$

**Proof.** Let us first consider case 1: the originator $u$ is the root. It then takes $d$ time units for $u$ to inform its $d$ children. Without loss of generality, we assume that the right-most child, which we will call $v$, is the last to be informed. Under model $M_3$, at each time unit every vertex must transmit the message to one of its neighbors to which it has not yet sent the message. Thus, each vertex that is not the root or the last leaf to be informed must send the message once along each of its $d+1$ (or 1, if the node is a leaf, and not the last leaf to be informed) edges. Hence, $v$ takes $d + 1$ time units to broadcast to its $d+1$ neighbors, and, on average, $d + 1 - \frac{1}{d+1}$ time units to inform its $d$ children due to the contribution of $u$.

The same argument applies to subsequent vertices, down to and including the vertices whose children are leaves. We therefore get that the average time is

$$d + (d + 1 - \frac{1}{d+1})(h-2) = d + \frac{d^2 + 2d}{d+1} (h-2).$$

Now we assume the originator is a leaf (case 2) which has distance $h-1$ from the root. Obviously it takes one time unit to inform its parent and $\frac{d+2}{2} \left( \frac{1}{d+1} + \frac{2}{d+1} + \ldots + \frac{d+1}{d+1} \right) = \frac{d+2}{2}$ according to the expected value formula) time units to go to each higher level up to the root. Thus, it needs $1 + \frac{d+2}{2}(h-2)$ to reach the root from the leaf. Since one child of the root has been informed, it only takes $d - \frac{1}{d}$ time units on average for the root to pass a message to its all $d$ children. The rest follows from case 1. We sum the total broadcasting time as

$$1 + \frac{d+2}{2} (h-2) + d - \frac{1}{d} + \frac{d^2 + 2d}{d+1} (h-2)$$

which proves case 2.
Case 3, in which the originator is neither the root nor a leaf, follows easily.

We can similarly derive the following:

**Theorem 4.2.** Let \( u \) be the originator in the complete \( d \)-ary tree \( T_{d,h} \). Then

\[
E[T_{1,u,T_{d,h}}] = E[T_{2,u,T_{d,h}}] = \begin{cases} 
  d(h - 1) & \text{if } u \text{ is the root,} \\
  d + \frac{3d+1}{2}(h - 2) & \text{if } u \text{ is a leaf,} \\
  \frac{d+1}{2}(i - 1) + d(h - \frac{1}{2}) & \text{if } u \text{ is at } i\text{th depth} \\
  & \text{for } i = 1, 2, \ldots, h - 2.
\end{cases}
\]

**Proof.** If \( u \) is the root, it takes \( d \) time units for the message to fully propagate to the next level of the tree. Multiplying this by the \( h - 1 \) levels the message must pass through yields \( d(h - 1) \) time units for broadcasting to complete.

If \( u \) is a leaf, one time unit is spent broadcasting to level \( h - 2 \), and \( \frac{d+1}{2}(h - 2) \) time units are spent on average propagating the message up to the root. Once the message reaches the root it takes \( d - 1 \) time units to propagate to each child of the root, and \( d(h - 2) \) time units for the message to travel back down the tree. Summing each of these times yields our result.

Finally, if \( u \) is neither the root nor a leaf, at the beginning it takes \( \frac{d+2}{2} \) time units on average for the message to move up one level, as no adjacent nodes are informed, and \( \frac{d+1}{2} \) time units to move up subsequent levels, as exactly one adjacent node is informed. Thus, the time to inform the root is \( \frac{d+2}{2} + \frac{d+1}{2}(i - 1) \). The rest follows from our previous analysis.

5. Complete Graphs under Model \( M_1 \)

We now turn our attention to the average-case analysis of model \( M_1 \) on \( K_n \), the complete graph on \( n \) vertices. For convenience, let these vertices be enumerated \( v_0, v_1, \ldots, v_{n-1} \).

We assume at time \( k \) that \( r \) vertices are informed and \( n - r \) are uninformed. Each informed vertex randomly informs one of \( n - r \) uninformed vertices at time \( k + 1 \) with probability \( \frac{1}{n-r} \); hence, each arrangement occurs with probability \( \frac{1}{(n-r)^r} \). We seek the probability \( P_m(r, n-r) \) of finding exactly \( m \) vertices uninformed at time \( k + 1 \).

Let \( A_i \) be the event that vertex \( v_i \) is uninformed, for \( i = 1, 2, \ldots, n - r \). In this event, the \( r \) informed vertices could inform all the remaining \( n - r - 1 \) vertices, and this could be done in \( (n - r - 1)^r \) different ways. Similarly, there are \( (n - r - 2)^r \) arrangements, leaving two preassigned vertices uninformed, etc. Accordingly,

\[
P_i = Pr\{A_i\} = \frac{(n-r-1)^r}{(n-r)^r} = \left(1 - \frac{1}{n-r}\right)^r, \text{ where } 1 \leq i \leq n - r,
\]

\[
P_{ij} = Pr\{A_i A_j\} = \left(1 - \frac{2}{n-r}\right)^r, \text{ where } 1 \leq i < j \leq n - r,
\]

\[
P_{ijk} = Pr\{A_i A_j A_k\} = \left(1 - \frac{3}{n-r}\right)^r, \text{ where } 1 \leq i < j < k \leq n - r, \text{ etc.}
\]
Let $S_1 = \sum P_i$, $S_2 = \sum P_{ij}$, $S_3 = \sum P_{ijk}$, etc. Note that in these sums each combination appears once and only once. Hence, for every $\nu \leq n - r$,

$$S_\nu = \binom{n-r}{\nu} \left(1 - \frac{\nu}{n-r}\right)^r.$$

The probability that at least one vertex is uninformed is given by

$$P = S_1 - S_2 + S_3 - S_4 + \ldots \pm S_{n-r},$$

(see [5]) and hence, we find that the probability that all vertices are informed is

$$P_0(r, n-r) = 1 - P = 1 - (S_1 - S_2 + S_3 - S_4 + \ldots \pm S_{n-r})$$

$$= \sum_{\nu=0}^{n-r} (-1)^\nu \binom{n-r}{\nu} (1 - \frac{\nu}{n-r})^r.$$

Consider now a distribution in which exactly $m$ vertices are uninformed. These $m$ vertices can be chosen in $\binom{n-r}{m}$ ways. The $r$ informed vertices broadcast to the remaining $n - r - m$ vertices so that each of these vertices is informed. The number of such distributions is

$$(n - r - m)^r P_0(r, n-r - m).$$

Dividing by $(n-r)^r$, we find that the probability that exactly $m$ vertices remain uninformed is

$$P_m(r, n-r) = \binom{n-r}{m} \left(1 - \frac{m}{n-r}\right)^r P_0(r, n-r - m)$$

$$= \binom{n-r}{m} \sum_{\nu=0}^{n-r-m} (-1)^\nu \binom{n-r-m}{\nu} (1 - \frac{\nu}{n-r-m})^r$$

by using

$$(1 - \frac{m}{n-r})^r (1 - \frac{\nu}{n-r-m})^r = (1 - \frac{m+\nu}{n-r})^r.$$

Finally, we are ready to prove the following:

**Theorem 5.1.** Let $V_k$ be the number of vertices informed by time $k$. Then,

$$E[T_{1, K_n}] = \min \{ k \mid E[V_k] = n \}$$

where

$$E[V_k] = n - \sum_{m=1}^{n-E[V_{k-1}]} mP_m([E[V_{k-1}]], n - [E[V_{k-1}]]$$

for $k > 1$, with initial condition $E[V_1] = 2$.

**Proof.** Clearly, we have

$$E[V_k] = E[V_{k-1}] + \text{average number of new vertices informed at time } k$$

$$= E[V_{k-1}] + (n - E[V_{k-1}]) - \text{average vertices uninformed at time } k$$

$$= n - \sum_{m=1}^{n-E[V_{k-1}]} mP_m([E[V_{k-1}]], n - [E[V_{k-1}]]).$$
This completes the proof of theorem. \[\]

We have encountered difficulties while solving the above recurrence mathematically. However, we both stimulated model $M_1$ directly using the following algorithm for $n$ up to $2^{29}$ with 1000 trials, and computed the recursion in Theorem 5.1 using Maple 8. The two results coincide and are very approximately equal to $[\log n] + 2$ for $n \geq 2^{10}$, and are between $[\log n]$ and $[\log n] + 2$ for $n < 2^{10}$, which indicates that the performance of $M_1$ on complete graphs is close to that of classical broadcasting.

**Algorithm M1Simulation($n$)**

```plaintext
k ← 0  // time units
i ← 1  // informed nodes
u ← n - 1  // uninformed nodes

START:
  j ← 0  // newly informed nodes at time k
  k ← k + 1
  for x ← 1 to i do
    z ← random($u$)  // z gets a random number in [1, $u$
    if z > j then  // if no collision
      j ← j + 1  // inform a new node
    if j = u then  // if all informed this cycle
      goto end
    i ← i + j  // increase informed nodes
    u ← u - j  // decrease uninformed nodes
  goto start

END:
print $k + \text{ "time units required"}$
```

Table 1 summarizes the results of this algorithm averaged over 1000 trials, which matches with data computed directly from the recurrence in Theorem 5.1 and clearly shows that the average-case time of $[\log n] + 2$ is much faster than the worst-case time of $n - 1$. The column *Time* represents the number of time units needed to complete broadcasting while the column *Time*(log($n$) + $k$) shows the time needed to finish broadcasting split into two terms, the first being log($n$) and the second being the additional time needed beyond the log($n$) time units.

We know from [9] that $t_i(K_n) = n - 1$ for all $i$. Here, we have another opportunity to show that worst-case messy broadcast times are often highly improbable.

**Corollary 5.1.** $Pr\{T_{1,v_0,K_n} = n - 1\} = \Pi_{i=2}^{n-1} \frac{1}{(n-i)^{3/2}}$.

**Proof.** Assume, without loss of generality, that vertex $v_0$ broadcasts to $v_i$ at time $i$ for all $i$ until broadcasting is complete.

There is a $\frac{1}{n^2}$ chance that $v_0$ and $v_1$ both broadcast to $v_2$ at time 2. Given that this occurs, there is a $\frac{1}{(n-3)^2}$ chance that vertices $v_0$, $v_1$, and $v_2$ all broadcast to $v_3$ at time 3. In general, there is a $\frac{1}{(n-i)^{3/2}}$ chance that vertices $v_0$ through $v_{i-1}$ broadcast to $v_i$ at time $i$, and
Table 1. Simulation Results for Model $M_1$.

<table>
<thead>
<tr>
<th># of Nodes ($n$)</th>
<th>$\log(n)$</th>
<th>Time</th>
<th>Time ($\log(n) + k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^1$</td>
<td>1</td>
<td>1.000</td>
<td>1 + 0.000</td>
</tr>
<tr>
<td>$2^2$</td>
<td>2</td>
<td>2.499</td>
<td>2 + 0.499</td>
</tr>
<tr>
<td>$2^3$</td>
<td>3</td>
<td>3.990</td>
<td>3 + 0.990</td>
</tr>
<tr>
<td>$2^4$</td>
<td>4</td>
<td>5.149</td>
<td>4 + 1.149</td>
</tr>
<tr>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$2^8$</td>
<td>8</td>
<td>9.941</td>
<td>8 + 1.941</td>
</tr>
<tr>
<td>$2^9$</td>
<td>9</td>
<td>10.998</td>
<td>9 + 1.998</td>
</tr>
<tr>
<td>$2^{10}$</td>
<td>10</td>
<td>12</td>
<td>10 + 2.000</td>
</tr>
<tr>
<td>$2^{11}$</td>
<td>11</td>
<td>13</td>
<td>11 + 2.000</td>
</tr>
<tr>
<td>$2^{12}$</td>
<td>12</td>
<td>14</td>
<td>12 + 2.000</td>
</tr>
<tr>
<td>$2^{13}$</td>
<td>13</td>
<td>15</td>
<td>13 + 2.000</td>
</tr>
<tr>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$2^{26}$</td>
<td>26</td>
<td>28</td>
<td>26 + 2.000</td>
</tr>
<tr>
<td>$2^{27}$</td>
<td>27</td>
<td>29</td>
<td>27 + 2.000</td>
</tr>
<tr>
<td>$2^{28}$</td>
<td>28</td>
<td>30</td>
<td>28 + 2.000</td>
</tr>
<tr>
<td>$2^{29}$</td>
<td>29</td>
<td>31</td>
<td>29 + 2.000</td>
</tr>
</tbody>
</table>

given that for all $k < i$, all informed vertices broadcasted to $v_k$ at time $k$. The result follows from the product of these conditional probabilities.

For large $n$, the probability shown in Corollary 5.1 is effectively zero.

Finally, we would like to mention that the following variant of broadcasting has been given by Frieze and Grimmett [7]. A complete graph contains $n$ vertices with one originator. At each time unit, each informed vertex informs another vertex randomly chosen from the graph independently of other choices (hence repetition is possible during broadcasting, which is different from $M_1$). Let $S_n$ be the number of time units required for broadcasting to complete. Then

$$\frac{S_n}{\log n} \rightarrow 1 + \log_e 2$$

in probability as $n \rightarrow \infty$. Furthermore, Pittel [12] improved this result, showing that

$$S_n = \log n + \log_e n + o(1).$$

Given that this model of “messy” broadcasting makes less assumptions about the informed status of neighbours than models $M_1$, $M_2$, and $M_3$, it is reasonable to expect that models $M_1$, $M_2$, and $M_3$ perform more efficiently than the model considered in [7].

Acknowledgments

We would like to thank Dr. Michael J. Wichura of the University of Chicago for his prompt advice concerning the occupancy problem which helped us in the analysis of complete graphs.
under model $M_1$, and also Marc Berndl, University of Toronto, for his help in refining Theorem 3.1.

The Natural Sciences and Research Council of Canada (NSERC) provided three of the authors with research grants. Chenkuan Li was supported by an NSERC Research Grant, and Tom Hart and Kevin Henry were supported by NSERC Undergraduate Student Research Awards.

References