Optimal Odd Gossiping*

Guillaume Fertin

LaBRI - U.M.R. CNRS 5800
Université Bordeaux I
351 Cours de la Libération
33405 Talence Cedex, France

Joseph G. Peters †

School of Computing Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6, Canada

CMPT TR 1998-24
School of Computing Science
Simon Fraser University
December, 1998

Abstract

In the *gossiping* problem, each node in a network starts with a unique piece of information and must acquire the information of all other nodes using two-way communications between pairs of nodes. In this paper we investigate gossiping in n-node networks with n odd. We use a *linear cost model* in which the cost of communication is proportional to the amount of information transmitted. In *synchronous* gossiping, the pairwise communications are organized into *rounds*, and all communications in a round start at the same time. We present optimal synchronous algorithms for all odd values of n. In *asynchronous* gossiping, a pair of nodes can start communicating while communications between other pairs are in progress. We provide a short intuitive proof that an asynchronous lower bound due to Peters, Raabe, and Xu is not tight.

Keywords: gossiping, communication, networks
AMS (MOS) subject classifications: 94A05, 68M10

*Part of this research was done while both authors were visiting Université de Nice - Sophia Antipolis
†Supported by NSERC of Canada and Université de Nice - Sophia Antipolis.

1 Introduction

Gossiping is an information dissemination problem in which each node of a communication network has a piece of information that must be acquired by all the other nodes. Information is communicated between pairs of nodes using two-way communications or *calls* along the communication links of the network. Gossiping is a well-studied problem. There are many papers describing algorithms that minimize the gossip time on various interconnection networks such as hypercubes and meshes. See [5, 3, 6] for surveys of these results.

There has been less study of the minimum time needed to gossip when the topology of the interconnection network does not restrict the communication patterns. Knödel [7] proved that the number of *rounds* of communication necessary to gossip is $\lceil \log_2(n) \rceil$ when n is even, and $\lceil \log_2(n) \rceil + 1$ when n is odd. He also proved sufficiency by describing gossip algorithms that meet the lower bounds on numbers of rounds. The *half-duplex* version of this problem, in which communication links can only be used in one direction at any given time, has also been studied [1, 8]. All of these papers assume a *unit cost* model in which a communication takes one time unit independent of the amount of information being transmitted. When messages are long, a *linear cost* model is more realistic since the length of the messages in most gossip algorithms grows exponentially.

In this paper, we assume a *store-and-forward, 1-port, full-duplex* model in which each communication involves two nodes and the single communication link that connects them, each node communicates with at most one other node at any given time, and information can flow simultaneously in both directions along a link. Each node starts with a message of length 1. Messages can be concatenated and sent as a single communication. We assume a *linear cost* model in which the time to send a message of length k is $\beta + k\tau$ where β is the *start-up* time to initiate a call between a pair of nodes and τ is the *propagation* time of a message of length 1 along a link. If the two nodes involved in a call send messages of different lengths, then the time for both nodes to complete the call is determined by the length of the longer message. A call involving messages of length k can be thought of as a start-up period that takes time β followed by a sequence of k *steps* each of which takes time τ.

A linear cost model can be either *synchronous* or *asynchronous*. In the synchronous linear cost model, a gossip algorithm consists of a sequence of *rounds* of simultaneous pairwise communications. All calls in a round start at the same time. Calls in a round may end at different times, depending on the lengths of the messages, but no node can start a new call until all nodes are ready to start new calls. In the asynchronous linear cost model, a call can start as soon as both nodes are ready to communicate. Thus, a pair of nodes can start communicating while calls between other pairs are in progress. The unit cost model is always synchronous since each call takes one time unit.

Fraigniaud and Peters [4] investigated the structure of minimum-time gossip algorithms using a linear cost model. They established lower and upper bounds on the time to gossip when the number of nodes n is even and showed that there is a synchronous minimum-time gossip algorithm for every even n. They also gave examples to show that minimum-time gossip algorithms for some odd values of n must be asynchronous - any synchronous algorithm requires strictly more than minimum time.

Peters, Raabe, and Xu [9] studied gossiping with n odd and a linear cost model. They proved a general lower bound of $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$ on the time to gossip. This lower bound holds for all odd n for both the synchronous and asynchronous models. The bound is achievable in the asynchronous case for some odd values of n, but for $n = 2^k - 1$, they proved that every gossip algorithm requires time strictly greater than $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$. For the synchronous case, they proved stronger lower bounds and conjectured that their lower bounds are achievable for all odd n. They gave an ad hoc synchronous algorithm that achieves their lower bound for $n = 2^k - 1$.

In Section 2, we briefly review the lower bounds for synchronous gossiping from [9]. In a preliminary version of this paper [2], we constructed gossip algorithms that achieve the lower bounds in [9] for approximately 35% of the odd values of n. Our main result in this paper is a collection of algorithms that achieves the lower bounds for all odd values of n, thereby establishing the truth of the conjecture in [9]. Our treatment of the synchronous upper bounds is split into two sections. In Section 3, we consider odd values of n that are in the *top half* of any range between two consecutive powers of 2. In Section 4, we consider the bottom halves of the ranges. The proof in [9] that the general asynchronous lower bound cannot be achieved when $n = 2^k - 1$ is long and complicated. In Section 5, we give a much shorter and more intuitive proof of this result.

2 Lower Bounds for Synchronous Gossiping

Knödel [7] showed that gossiping in the unit cost model requires $\lceil \log_2(n) \rceil + 1$ rounds when n is odd. This lower bound on the number of rounds is also valid for the synchronous linear cost model. We say that a node is *idle* during a round of a synchronous gossip algorithm if it is not participating in a call during that round. Since calls involve pairs of nodes and n is odd, there will be at least one idle node at any given time. It is now immediate that at least n steps are required to gossip when n is odd because each node needs to acquire $n - 1$ pieces of information, and at least one node is idle during each step. This gives a lower bound of $\max\{(\lceil \log_2(n) \rceil + 1)\beta, n\tau\}$. Peters, Raabe, and Xu [9] proved a lower bound of $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$ for odd n for both the synchronous and asynchronous cases. They proved stronger lower bounds for the synchronous case by fixing the number of rounds to be $\lceil \log_2(n) \rceil + 1$ and then focussing on the required number of steps. We take the same approach to synchronous upper bounds.

The required number of rounds, $\lceil \log_2(n) \rceil + 1$, is the same for every odd n between $2^{k-1} + 1$ and $2^k - 1$, where $k = \lceil \log_2(n) \rceil$. The required total number of steps and also the required numbers of steps in each of the rounds depend on whether n is in the *bottom half* of the range, $2^{k-1} < n < 2^{k-1} + 2^{k-2}$, or the *top half* of the range, $2^{k-1} + 2^{k-2} < n < 2^k$. We will often refer to the bottom halves of all ranges collectively as *the bottom half* and similarly for *the top half.*

Theorem 1 ([9]) *A synchronous gossip algorithm for odd n in the top half which has $\lceil \log_2(n) \rceil + 1$ rounds requires at least $2n - 2^{k-1} - 1$ steps where $k = \lceil \log_2(n) \rceil$, and $k \geq 2$. A feasible sequence of numbers of steps in the rounds is $1\ 2\ 4\ 8\ \ldots\ 2^{k-2}\ x\ x$ where $x = n - 2^{k-1}$.*

Proof Outline: The numbers of steps listed in the statement of the theorem for the first $k - 1$ rounds are the maximum numbers of usable steps since nodes can at most double the amount of information that they know each round. During each round (including the last two rounds), at least one node must be idle. There must be enough steps in the last two rounds for nodes that are idle in earlier rounds to receive the information that they are missing. It can be shown that decreasing the number of steps in any round results in an increase in at least one other round. □

Theorem 2 ([9]) *A synchronous gossip algorithm for odd n in the bottom half which has* $\lceil \log_2(n) \rceil +$ 1 *rounds requires at least* $2^{k-2} - 1 + 2\lceil \frac{n-2^{k-2}}{2} \rceil + \lfloor \frac{n-2^{k-2}}{2} \rfloor$ *steps where* $k = \lceil \log_2(n) \rceil$, *and* $k \geq 2$. *A feasible sequence of numbers of steps is* 1 2 4 8 ... 2^{k-3} *in the first* $k-2$ *rounds. Two of the last three rounds have* $z = \lceil \frac{n-2^{k-2}}{2} \rceil$ *steps and the other round has* $y = \lfloor \frac{n-2^{k-2}}{2} \rfloor$ *steps.*

Proof Outline: Similar to Theorem 1. □

Conjecture 1 ([9]) *There are synchronous gossip algorithms that achieve the lower bounds of Theorems 1 and 2 for every odd n.*

We note that there can be a trade-off between the number of rounds and the number of steps in a synchronous gossip algorithm. If more than $\lceil \log_2(n) \rceil + 1$ rounds are permitted, then the number of steps can often be reduced. Depending on the relative values of β and τ, the fastest algorithm could have more than $\lceil \log_2(n) \rceil + 1$ rounds. We do not investigate this trade-off in this paper. See [4] for results and some examples.

3 Synchronous Gossiping in the Top Half

In this section, we describe algorithms that achieve the lower bound of Theorem 1 for all values of n in the top half. This proves Conjecture 1 for every odd n in the top half of any range between two consecutive powers of 2, that is, for every odd n, $2^{k-1} + 2^{k-2} + 1 \leq n \leq 2^k - 1$, $k \geq 3$. Our result is the following.

Theorem 3 *For any odd n in the top half, there is a synchronous gossip algorithm with* $\lceil \log_2(n) \rceil +$ 1 *rounds and* $2n - 2^{k-1} - 1$ *steps, where* $k = \lceil \log_2(n) \rceil$, $k \geq 2$.

Note: Strictly speaking, the terms *top half* and *bottom half*, and the corresponding mathematical definitions do not make sense for $k = 2$ since $n = 3$ is the only odd value in this range. For convenience, we will consider $n = 3$ to be in the top half. Note that Theorems 1 and 2 are both true for $n = 3$.

As dictated by Theorem 1, our algorithms have $k + 1$ rounds, where $k = \lceil \log_2(n) \rceil$, and the numbers of steps in the rounds are 1 2 4 8 ... 2^{k-2} x x respectively where $x = n - 2^{k-1}$. Our algorithms and proofs of correctness in this section and in Section 4 are based on two properties of *partial gossip algorithms* and on the notion of *experts*. We say that a node is an *expert* of a set S if it knows the information of every node in S. Our *partial gossip algorithms* for n nodes have $k = \lceil \log_2(n) \rceil$ rounds with 2^{i-1} steps in each round $i = 1, 2, \ldots, k$. Note that a partial gossip algorithm cannot be extended to an optimal complete gossip algorithm in the linear cost model by the addition of a $k + 1^{\text{st}}$ round because round k of an optimal algorithm has $x < 2^{k-1}$ steps. Our optimal complete gossip algorithms will use one or more partial gossip algorithms as subroutines during the first $\lceil \log_2(n) \rceil - 2$ or $\lceil \log_2(n) \rceil - 1$ rounds.

Property A For any odd $n \geq 3$, we say that property A_n is true if there is a partial gossip algorithm for n nodes with $k = \lceil \log_2(n) \rceil$ rounds and 2^{i-1} steps in each round $i = 1, 2, \ldots, k$ such that after k rounds:

1. 2^{k-1} nodes are experts (i.e., know all n pieces of information), and
2. each of the remaining $x = n - 2^{k-1}$ nodes knows at least 2^{k-1} pieces of information.

Property B For any odd $n \geq 5$, we say that property B_n is true if there is a partial gossip algorithm for n nodes with $k = \lceil \log_2(n) \rceil$ rounds and 2^{i-1} steps in each round $i = 1, 2, \ldots, k$ such that at least $x = n - 2^{k-1}$ nodes are idle in round k.

To prove Theorem 3, we will first prove that Property A_n is true for every odd $n \geq 3$. Theorem 3 will then follow easily. The organization of the proof of Property A consists of three propositions as shown in Figure 1. Proposition 1 establishes Property A in the *top half*, i.e., A_n is true for every odd n in the top half. The inductive steps are shown with solid lines and arrows in Figure 1. Proposition 2, shown with dashed lines and arrows in Figure 1, proves that both Properties A and B hold in the *bottom quarter*, i.e., for every odd n, $2^{k-1} < n < 2^{k-1} + 2^{k-3}$. Proposition 3 is a more complicated induction; it shows that if Property A holds for an entire range $2^{k-1} < n < 2^k$, then both Properties A and B hold in the *second quarter* two ranges up, i.e., for every $2^{k+1} + 2^{k-1} < n < 2^{k+1} + 2^k$. The steps of Proposition 3 are shown with dotted lines and arrows in Figure 1. Collectively, the three propositions cover all odd values of n.

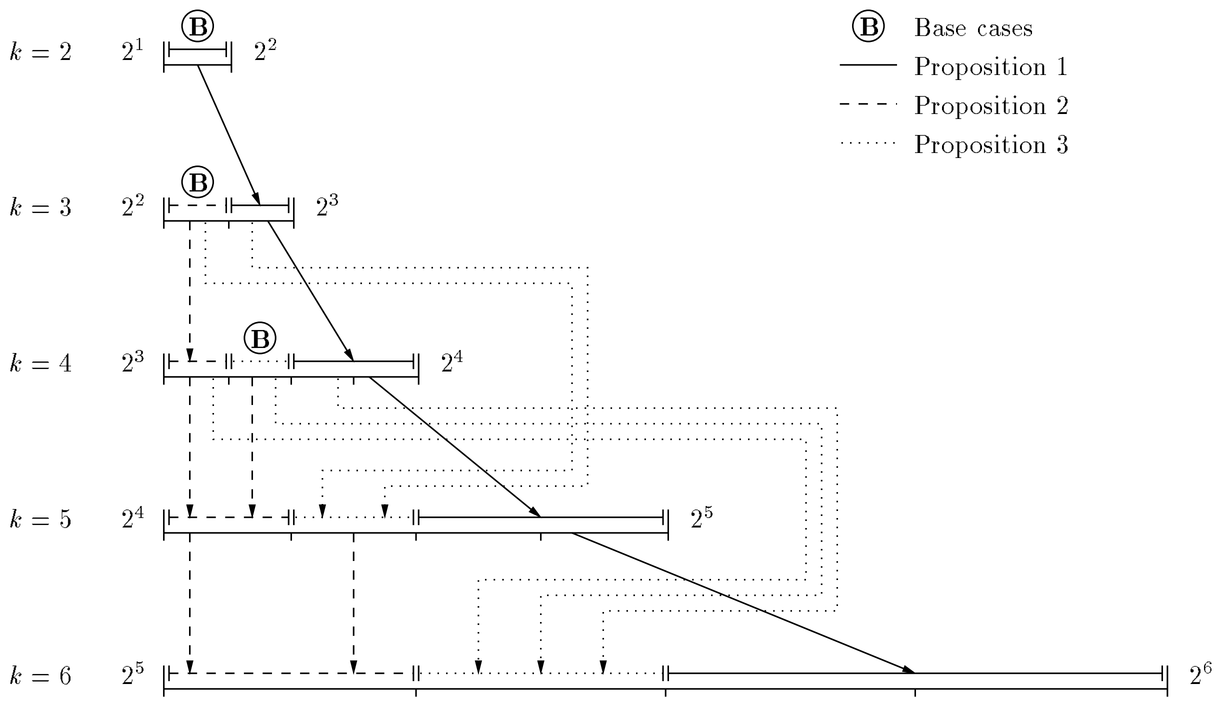

Figure 1: Organization of the proof of Property A

Proposition 1 *Property A_n is true for every odd $n \geq 3$ in the top half.*

We will prove Proposition 1 by induction on $k = \lceil \log_2(n) \rceil$ using three lemmas.

Lemma 1 *Let $n = 2^{k-1} + x$ be an odd number such that $k = \lceil \log_2(n) \rceil$ and $k \geq 2$. If A_n is true, then $A_{n'}$ is true where $n' = 2^k + 2^{k-1} + x$.*

Proof: Let $n = 2^{k-1} + x$ for any fixed $k \geq 2$ and odd x. Note that $1 \leq x \leq 2^{k-1} - 1$, since $\lceil \log_2(n) \rceil = k$. Also note that a gossip algorithm for n nodes has $k+1$ rounds [7]. Suppose that A_n is true and let $n' = 2^k + 2^{k-1} + x = 2^k + n$. Partition the set S of n' nodes into two subsets S_1 and S_2, such that $|S_1| = 2^k$ and $|S_2| = 2^{k-1} + x = n$. Gossiping among n' nodes requires $k+2$ rounds. During the first k rounds of a partial gossip algorithm, the nodes of S_1 and the nodes of S_2 communicate independently. The 2^k nodes of S_1 can all be experts of S_1 after k rounds. The $n = 2^{k-1} + x$ nodes of S_2 can satisfy property A_n after k rounds by assumption. In round $k+1$ of the partial gossip algorithm, the 2^{k-1} experts of S_2 exchange all of their information with 2^{k-1} of the experts of S_1 to create 2^k experts of $S_1 \cup S_2 = S$. The remaining x nodes of S_2 exchange information with x nodes of S_1. After round $k+1$, these $2x$ nodes and the remaining $2^{k-1} - x$ nodes of S_1 each know at least 2^k pieces of information. Since round $k+1$ has 2^k steps, all conditions of Property $A_{n'}$ are satisfied.

The algorithm is illustrated in Figure 2. The column of two boxes with bold outlines on the left shows the situation after k rounds. Round $k+1$ is shown in detail and the box on the right shows the situation after round $k+1$. The gray shading is used to indicate nodes that are idle during round $k+1$. □

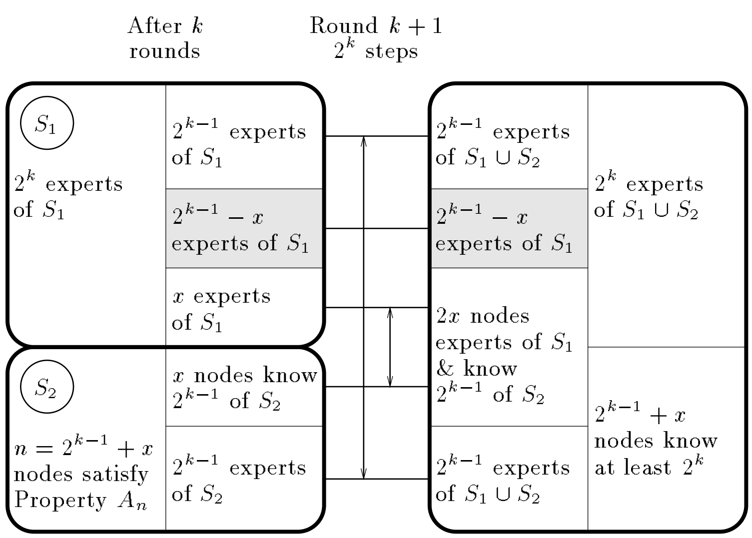

Figure 2: Proof of Lemma 1

Lemma 2 *Let $n = 2^{k-1} + x$ be an odd number such that $k = \lceil \log_2(n) \rceil$, $2^{k-2} + 1 \leq x \leq 2^{k-1} - 1$, and $k \geq 3$. If A_n is true, then A_{2n-1} is true.*

Proof: Suppose that A_n is true and consider a set S of $2n-1$ nodes. Partition S into three subsets S_1, S_2, and S_3 such that $|S_1| = n$, $|S_2| = 2^{k-1}$, and $|S_3| = x - 1$. Note that $k+2$ rounds are required to gossip among $2n-1$ nodes. During the first $k-1$ rounds of a partial gossip algorithm, the nodes of S_2 and S_3 communicate within their own subsets. After $k-1$ rounds, all nodes of S_2 and S_3 can be experts of their respective subsets because the number of nodes in each subset is at most 2^{k-1} and is even. After a partial gossip algorithm of k rounds, the $|S_1| = n$ nodes of S_1 can satisfy Property A_n by assumption. However, n is odd, so at least one node u of S_1 will be idle during round k. Node u knows at least 2^{k-1} pieces of information after $k-1$ rounds by assumption A_n, and is free to communicate with a node v of S_2 in round k. As shown in Figure 3, round k of the partial gossip algorithm for S_1 has been modified to include this communication between u and v. The remaining $n-1$ nodes of S_1 continue to communicate within S_1 during round k. Using Figure 3, it is not difficult to verify that $S = S_1 \cup S_2 \cup S_3$ satisfies all conditions of Property A_{2n-1} after round $k+1$. Note that the condition $x \geq 2^{k-2}+1$ in the statement of the lemma is required in round $k+1$ to ensure that $2x - 2 - 2^{k-1} \geq 0$. □

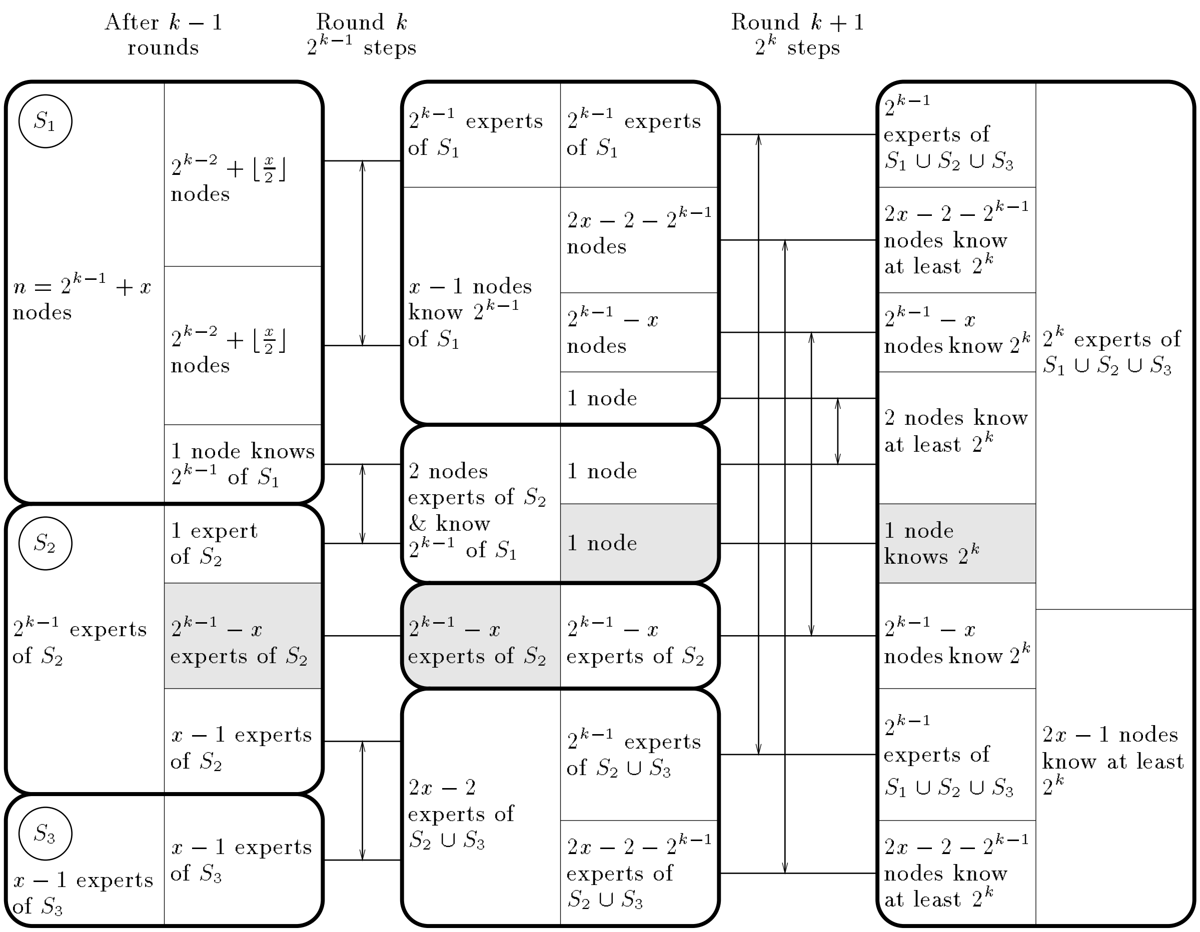

Figure 3: Proof of Lemma 2

Lemma 3 *Let $n = 2^{k-1} + x$ be an odd number such that $k = \lceil \log_2(n) \rceil$, $2^{k-2}+1 \leq x \leq 2^{k-1} - 3$, and $k \geq 4$. If A_n is true, then A_{2n+1} is true.*

Proof: This proof is similar to the proof of Lemma 2. Partition the set S of $2n+1$ nodes into three subsets S_1, S_2, and S_3 such that $|S_1| = n$, $|S_2| = 2^{k-1}$, and $|S_3| = x+1$. Gossiping among $2n+1$ nodes requires $k+2$ rounds. As in the proof of Lemma 2, the nodes of S_2 and S_3 communicate within their own subsets during the first $k-1$ rounds of a partial gossip algorithm. After $k-1$ rounds, all nodes of S_2 and S_3 can be experts of their respective subsets because S_2 and S_3 are of even order at most 2^{k-1}. After k rounds, the nodes of S_1 can satisfy A_n by assumption. At least one node u of S_1 is idle during round k of a partial gossip algorithm for S_1 and knows at least 2^{k-1} pieces of information by assumption A_n. As in the proof of Lemma 2, round k of the partial gossip algorithm for S_1 is modified to include a communication between u and a node v of S_2. The communications during rounds k and $k+1$ of the partial gossip algorithm are shown in Figure 4 and prove that $S = S_1 \cup S_2 \cup S_3$ satisfies all conditions of Property A_{2n+1} after round $k+1$. The condition $x \geq 2^{k-2}+1$ is needed to ensure that $2x+1-2^{k-1} \geq 0$ in round $k+1$. The reason for the condition $x \leq 2^{k-1}-3$ (instead of $x \leq 2^{k-1}-1$ as in Lemma 2) is to ensure that $|S_2| > |S_3|$ so that S_2 has a node v to communicate with node u of S_1 in round k. □

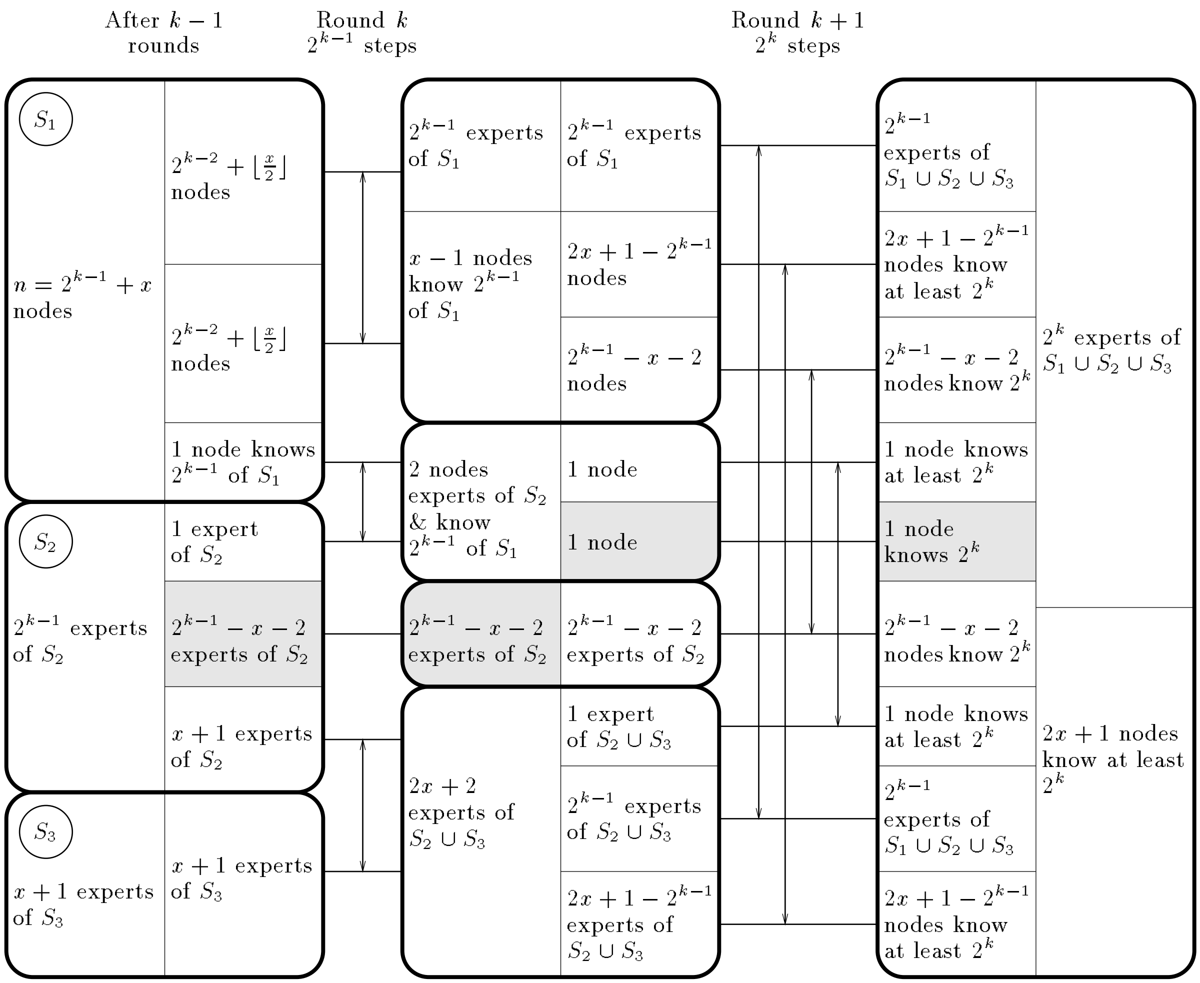

Figure 4: Proof of Lemma 3

Proof of Proposition 1: The case A_3 is true by inspection and case A_7 follows from A_3 by Lemma 1. Suppose that Property A_n is true for every odd n in the top half of some range between two consecutive powers of 2. That is, A_n is true for $2^{k-1}+2^{k-2} < n < 2^k$ with $\lceil \log_2(n) \rceil = k$ for some $k \geq 3$. Then A_n is true for $n = 2^{k+1}-1$ by Lemma 1, and for every other odd n in the top half with $\lceil \log_2(n) \rceil = k+1$ by Lemmas 2 and 3. Hence A_n holds for every odd $n \geq 3$ in the top half by induction. □

The next two propositions together show that Properties A and B hold in the bottom half. The two propositions depend on each other: the proof of Proposition 2 assumes the truth of Proposition 3 and vice versa. See Figure 1.

Proposition 2 *Property A_n and Property B_n are true for every odd $n \geq 5$ in the bottom quarter.*

Proof: We will prove this result by induction on $k = \lceil \log_2(n) \rceil$. The base case $n = 5$, shown in Figure 5, proves the proposition for $k = 3$. (For convenience, we will consider $n = 5$ to be in the bottom quarter of its range even though the term does not really make sense when $k = 3$.) In Figure 5, each horizontal line represents one node and the numbers in the boxes indicate the information received during the communication immediately to the left. For example, in round 2, node 4 sends items 3 and 4 to node 5 and receives item 5. Shading indicates that a node was idle during the round. The induction step shows that if Properties A and B hold in the bottom

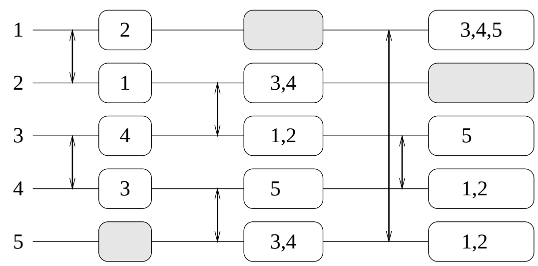

Figure 5: Base case $n = 5$ for proof of Proposition 2

half of some range, then both properties hold in the bottom *quarter* of the next higher range. We assume the truth of Proposition 3 which states that Properties A and B hold in the *second* quarter of every range. Now suppose that the Properties A_n and B_n are true for some $n = 2^{k-1} + x$ with $k = \lceil \log_2(n) \rceil$ and $1 \leq x < 2^{k-2}$. Thus, n is in the bottom half $2^{k-1} < n < 2^{k-1} + 2^{k-2}$ for some $k \geq 3$. To show that Properties $A_{n'}$ and $B_{n'}$ are true for $n' = 2^k + x = n + 2^{k-1}$, partition the set S of n' nodes into subsets S_1 and S_2 such that $|S_1| = n = 2^{k-1} + x$ and $|S_2| = 2^{k-1}$. The nodes of S_2 all become experts of S_2 by gossiping among themselves for $k - 1$ rounds. By assumption, there is a partial gossip algorithm for S_1 that has k rounds and satisfies Properties A_n and B_n. In particular, there will be x idle nodes of S_1 during round k of this partial gossip algorithm by Property B_n and these x idle nodes will know 2^{k-1} pieces of information of S_1 by Property A_n. These x nodes are free during round k to exchange information with x nodes of S_2 as shown in Figure 6. Using Figure 6, it can be verified that the n' nodes of $S = S_1 \cup S_2$ satisfy Properties $A_{n'}$ and $B_{n'}$ after $k + 1$ rounds. This proves the proposition for all n in the bottom quarter assuming that Proposition 3 is true. □

Proposition 3 *Property A_n and Property B_n are true for every odd $n \geq 11$ in the second quarter.*

Proof: We will prove this result by induction on $k = \lceil \log_2(n) \rceil$. The base case $n = 11$, shown in Figure 7, proves the proposition for $k = 4$. The induction step will show that if Property A holds in an entire range between two consecutive powers of 2, then Properties A and B both hold in the second quarter <u>two</u> ranges up. More precisely, if Property A holds for $2^{k-1} < n < 2^k$ with

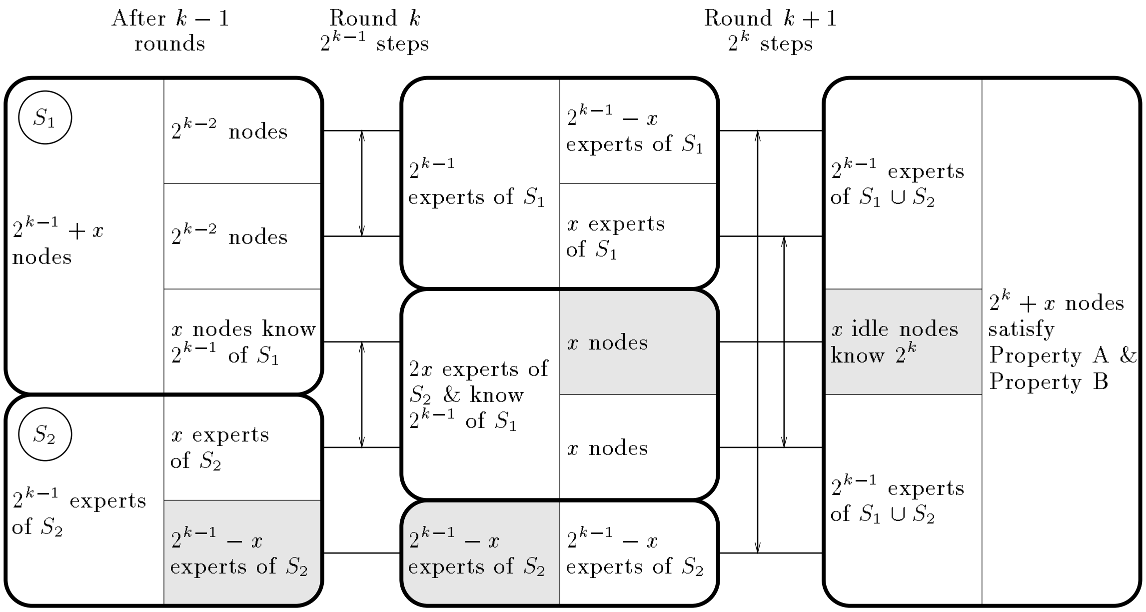

Figure 6: Proof of Proposition 2

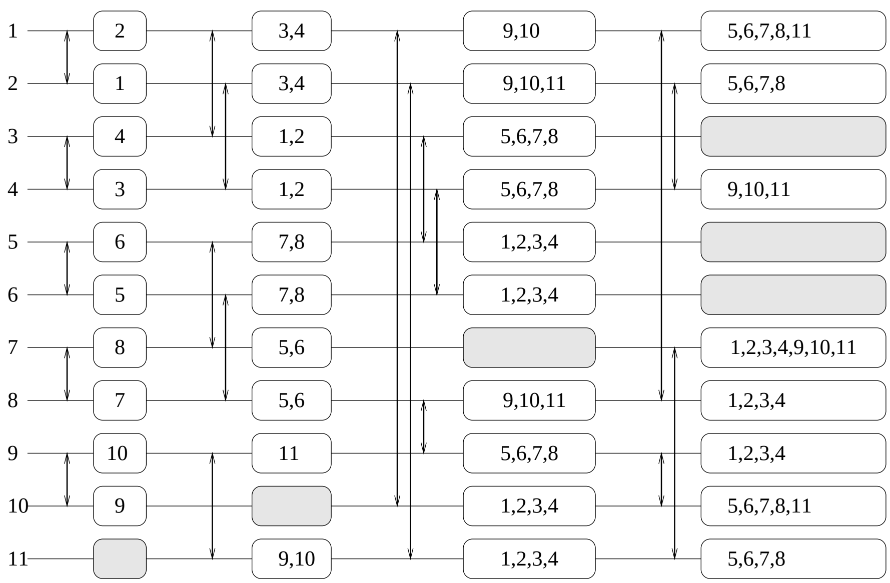

Figure 7: Base case $n = 11$ for proof of Proposition 3

$k = \lceil \log_2(n) \rceil$ for some $k \geq 3$, then both Properties A and B hold for $2^{k+1} + 2^{k-1} < n < 2^{k+1} + 2^k$. Proposition 1 proves that Property A holds in the top half of any range. We assume the truth of Proposition 2 which states that Property A (and also Property B) holds in the bottom quarter of every range. Now suppose that Property A_n is true for some $n = 2^{k-1} + x$ with $k = \lceil \log_2(n) \rceil$ and $1 \leq x < 2^{k-1}$ for some $k \geq 3$. The algorithm in Figure 8 proves that Properties $A_{n'}$ and $B_{n'}$ are true for $n' = 2^{k+1} + 2^{k-1} + x = n + 2^{k+1}$. This proves the proposition for all n in the second quarter assuming that Proposition 2 is true. □

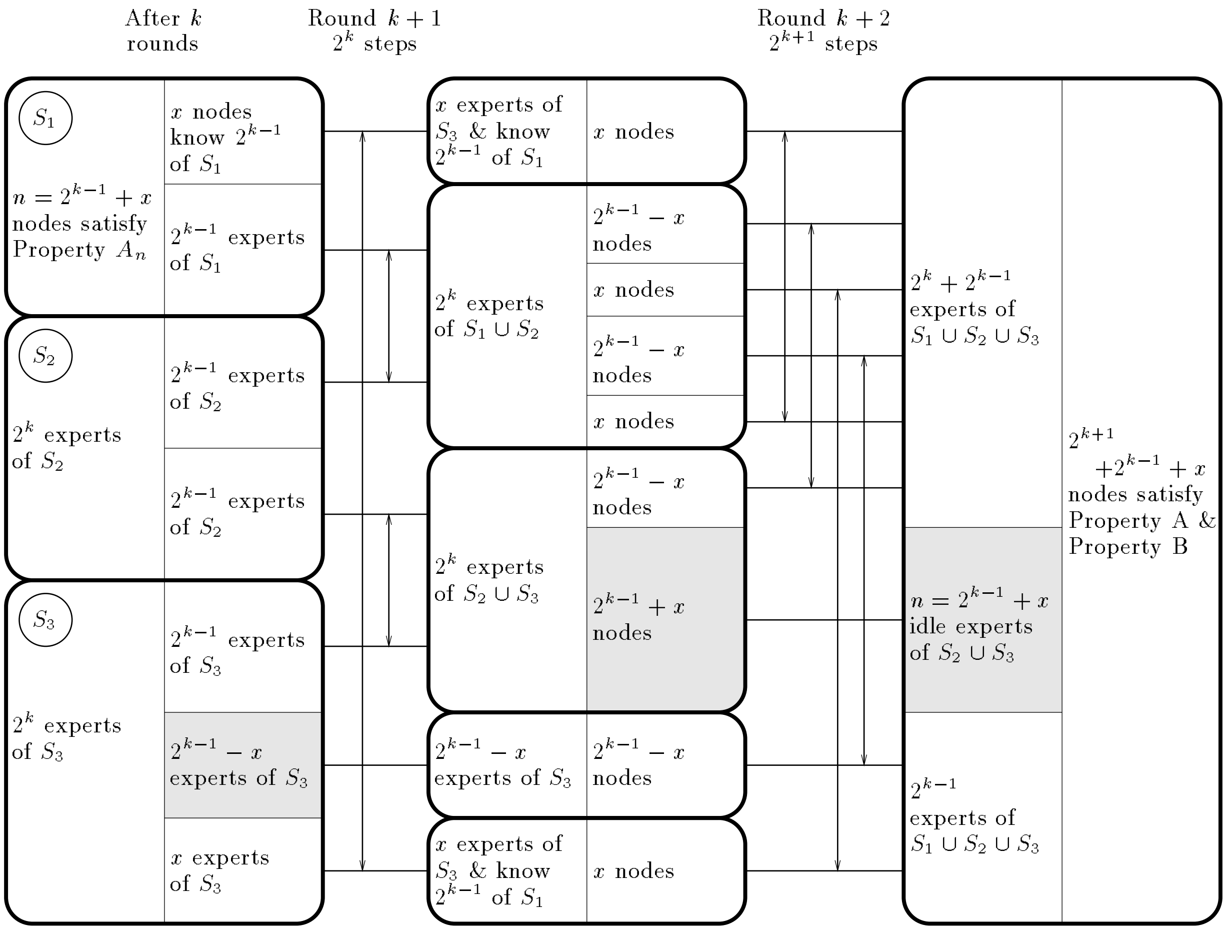

Figure 8: Proof of Proposition 3

Proof of Theorem 3: Before describing our optimal gossip algorithm for the top half, we need to complete the proof that Property A_n holds for all odd n. The proof is illustrated in Figure 1. We again use induction on $k = \lceil \log_2(n) \rceil$. There are several base cases. Properties A_3, A_5, A_7, and A_{11} have already been established in the proofs of the three propositions. Property A_9 follows from A_5 by Proposition 2, and Properties A_{13} and A_{15} follow from A_7 by Proposition 1. This proves the theorem for $k = 2$, $k = 3$, and $k = 4$. We need to show that if Property A is true for all n with $\lceil \log_2(n) \rceil = k - 1$ and $\lceil \log_2(n) \rceil = k$, then Property A is true for all n with $\lceil \log_2(n) \rceil = k + 1$. Property A_n is true in the top half, $2^k + 2^{k-1} < n < 2^{k+1}$ by Proposition 1 from the previous top half, $2^{k-1} + 2^{k-2} < n < 2^k$. Properties A_n and B_n are true in the first quarter, $2^k < n < 2^k + 2^{k-2}$, by Proposition 2 from the previous bottom half, $2^{k-1} < n < 2^{k-1} + 2^{k-2}$. Properties A_n and B_n are true in the second quarter, $2^k + 2^{k-2} < n < 2^k + 2^{k-1}$, by Proposition 3 from the entire range for $k - 1$, $2^{k-2} < n < 2^{k-1}$. Thus, Property A_n is true for every odd $n \geq 3$ by induction.

To complete the proof of the theorem, we use the optimal gossip algorithm shown in Figure 9. The number of rounds $k + 1 = \lceil \log_2(n) \rceil + 1$, and the number of steps $x = n - 2^{k-1}$ in each of the last two rounds, match the lower bound in Theorem 1, as does the total number of steps for the algorithm, $2^{k-1} - 1 + 2x = 2n - 2^{k-1} - 1$. It is not difficult to verify that all nodes of $S = S_1 \cup S_2$ will be experts of S after round $k + 1$. Note that $2^{k-2} < x < 2^{k-1}$, so $n = 2^{k-1} + x$ is in the top half. □

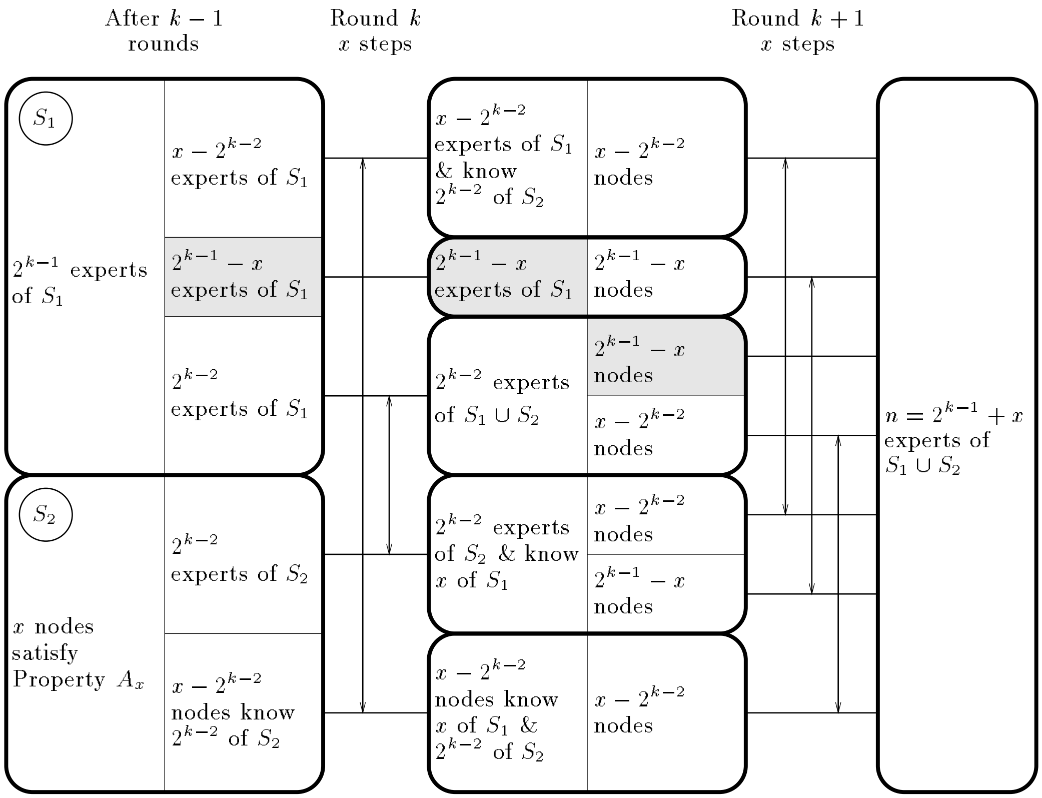

Figure 9: Optimal gossip algorithm in the top half

4 Synchronous Gossiping in the Bottom Half

In this section, we describe algorithms that achieve the lower bound of Theorem 2 for all odd values of n in the bottom half. As dictated by Theorem 2, our algorithms have $k+1$ rounds, where $k = \lceil \log_2(n) \rceil$, the numbers of steps in the first $k-2$ rounds are $1\ 2\ 4\ 8\ \ldots\ 2^{k-3}$ respectively, two of the last three rounds have $z = \lceil \frac{n-2^{k-2}}{2} \rceil$ steps, and the other round has $y = \lfloor \frac{n-2^{k-2}}{2} \rfloor$ steps.

Theorem 4 *For any odd n in the bottom half, there is a synchronous gossip algorithm with $\lceil \log_2(n) \rceil + 1$ rounds and $2^{k-2} - 1 + 2\lceil \frac{n-2^{k-2}}{2} \rceil + \lfloor \frac{n-2^{k-2}}{2} \rfloor$ steps, where $k = \lceil \log_2(n) \rceil$, $k \geq 3$.*

Proof: The organization of this proof is more straightforward than the proof for the top half, but the diagrams are more complicated. The special cases $n = 2^{k-1} + 2^{k-2} - 1$ and $n = 2^{k-1} + 2^{k-2} - 3$ are shown in Figures 10 and 11 respectively.

For the case $n = 2^{k-1} + 2^{k-2} - 1$ (Figure 10), note that $z = 2^{k-2}$ and $y = 2^{k-2} - 1$, and we have chosen to have y steps in the last round (round $k+1$). There are enough steps in the last round for a node to learn all of the information of S_1, but not quite enough steps to learn S_2 or S_3. Therefore, some nodes need to learn at least one item from S_2 or S_3 in round k. Fortunately, there are more than enough steps in round k to permit this.

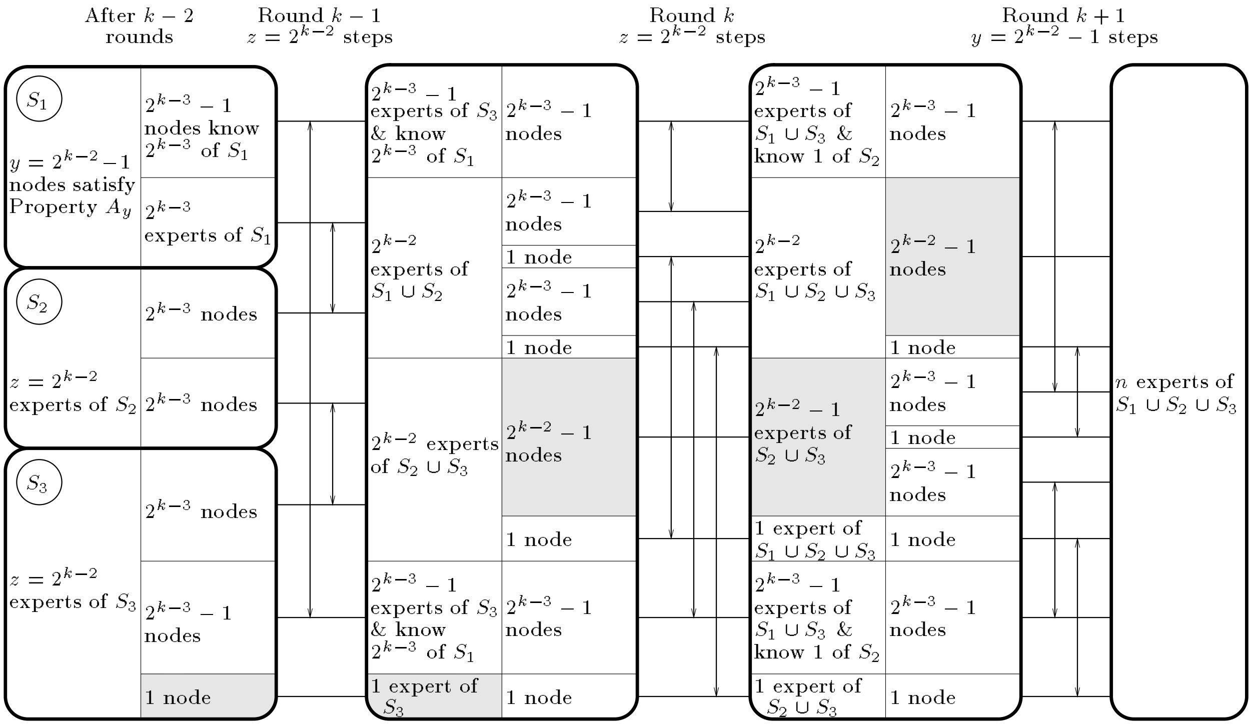

Figure 10: Gossip algorithm for $n = 2^{k-1} + 2^{k-2} - 1$

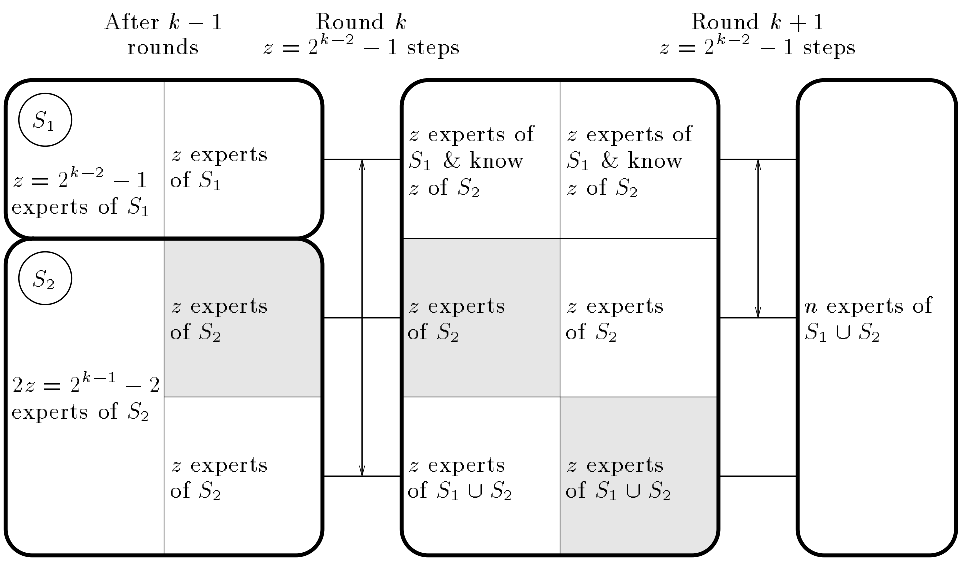

Figure 11: Gossip algorithm for $n = 2^{k-1} + 2^{k-2} - 3$

For the case $n = 2^{k-1} + 2^{k-2} - 3$ (Figure 11), $z = 2^{k-2} - 1$ and $y = 2^{k-2} - 2$, and we have chosen to have y steps in round $k-1$. So, the step sequence for the rounds is: $1\ 2\ 4\ 8\ \ldots\ 2^{k-3}\ y\ z\ z$. The subset S_1 has $z = 2^{k-2} - 1$ nodes and z is in the top half. By Theorem 3, we can gossip in S_1 in $k-1$ rounds with the step sequence, $1\ 2\ 4\ 8\ \ldots\ 2^{k-4}\ x\ x$, with $x = 2^{k-2} - 1 - 2^{k-3} = 2^{k-3} - 1$. Since x is less than the numbers of steps available in rounds $k-2$ and $k-1$ (which have 2^{k-3} and

$2^{k-2} - 2$, steps respectively), the gossiping in S_1 can be completed. The number of nodes in S_2 is even, so gossiping can be completed in $k-1$ rounds and the step sequence 1 2 4 8 ... 2^{k-3} y by Lemma 2.3 in [4].

The structures of the algorithms for the remaining values of n in the bottom half, $2^{k-1} + 1 \le n \le 2^{k-1} + 2^{k-2} - 5$, depend or whether y is even (and $z = y + 1$ is odd), or y is odd (and z is even). Figures 12 and 13 show the first $k-1$ rounds for y even and y odd respectively. Both algorithms use the fact, established in the previous section, that Property A_n holds for all odd n, and both algorithms use z steps in round $k-1$. In both cases, $2^{k-3} \le y \le 2^{k-2} - 3$ and $2^{k-3} + 1 \le z = y + 1 \le 2^{k-2} - 2$. Figures 12 and 13 should require no further explanation.

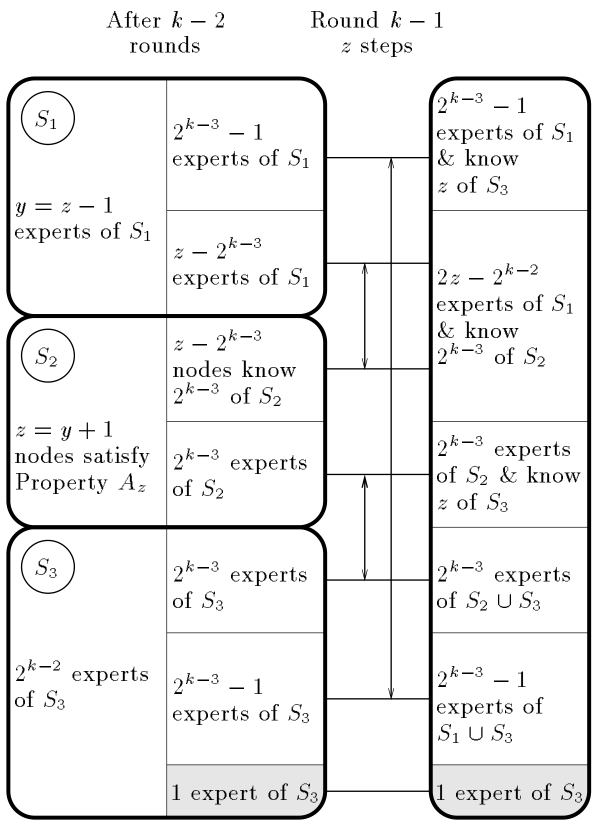

Figure 12: Round $k-1$ when y is even

The last two rounds k and $k+1$ depend on whether y is even or odd, and also on whether n is in the first quarter or the second quarter. Thus, there are four cases to consider for the last two rounds. The two algorithms for round $k-1$ shown in Figures 12 and 13 are common to the first and second quarters.

The last two rounds in the first quarter are shown for y even and y odd in Figures 14 and 15 respectively. In both cases $2^{k-1} + 1 \le n \le 2^{k-1} + 2^{k-3} - 1$, so $2^{k-3} \le y \le 3 \cdot 2^{k-4} - 1$ and $2^{k-3} + 1 \le z = y + 1 \le 3 \cdot 2^{k-4}$. Also, $|S_1| = y$, $|S_2| = z$, and $S_3 = 2^{k-2}$. Since $y = \lfloor \frac{n - 2^{k-2}}{2} \rfloor$ and $z = \lceil \frac{n-2^{k-2}}{2} \rceil$, we get $|S_3| < y + z$. With this information, it is not difficult to verify that all of the blocks shown in the diagrams contain positive numbers of nodes and that there are enough steps in rounds k and $k+1$ for each pair of communicating nodes to exchange all of the information specified. The only other observation that should be made concerns blocks of nodes such as the

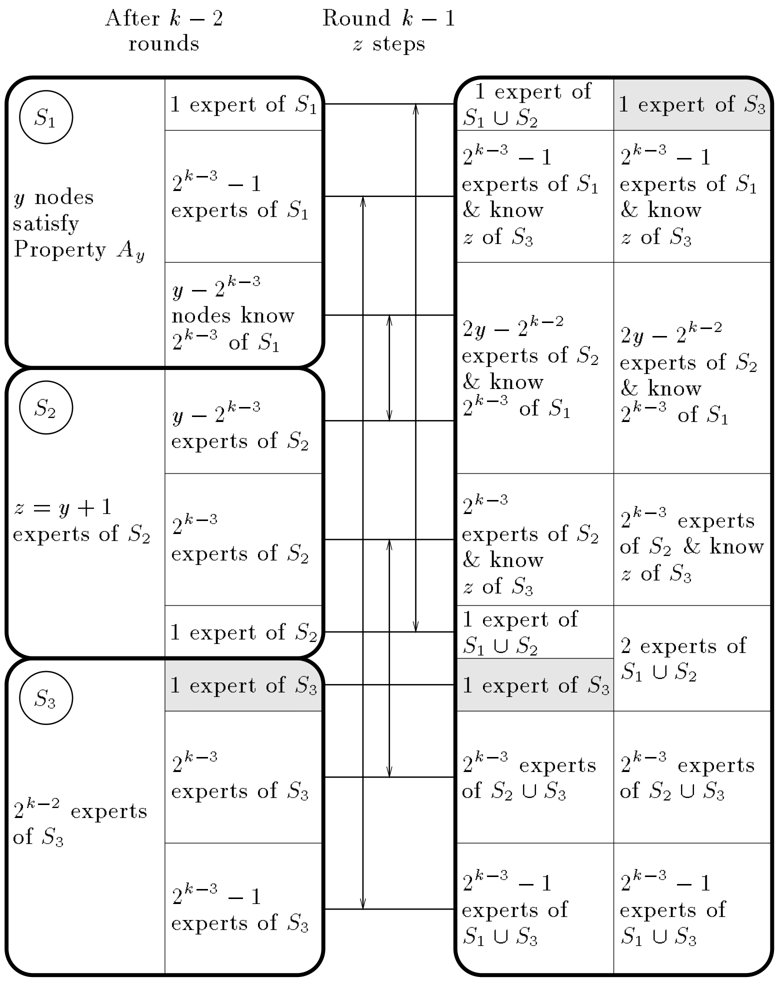

Figure 13: Round $k-1$ when y is odd

first and third blocks in Figure 14 which exchange information in round k. The nodes in these two blocks each know z pieces of information from S_3 after $k-1$ rounds and they must all be experts of S_3 after round k. Since $|S_3| < y+z$, there are enough steps in round k for each node to acquire the information that it needs, but care must be taken in round $k-1$ of Figure 12 that the nodes in the two blocks acquire complementary information from S_3. Similar comments apply to the second and fourth blocks in round k of Figure 15 and to other pairs of blocks in the two diagrams.

The last two rounds in the second quarter are shown for y even and y odd in Figures 16 and 17 respectively. In both cases $2^{k-1}+2^{k-3}+1 \leq n \leq 2^{k-1}+2^{k-2}-5$, so $3\cdot 2^{k-4} \leq y \leq 2^{k-2}-3$ and $3\cdot 2^{k-4}+1 \leq z = y+1 \leq 2^{k-2}-2$. As in the first quarter, $|S_1| = y$, $|S_2| = z$, and $S_3 = 2^{k-2} < y+z$. Observations similar to the observations for the first quarter also apply to the second quarter. The only additional observation for the second quarter is that the range of values of n, and consequently the ranges of y and z, are reduced because the two largest values of n in the second quarter, $n = 2^{k-1}+2^{k-2}-1$ and $n = 2^{k-1}+2^{k-2}-3$, have been handled as special cases. □

Theorem 3 and Theorem 4 together prove the truth of Conjecture 1 for every odd n.

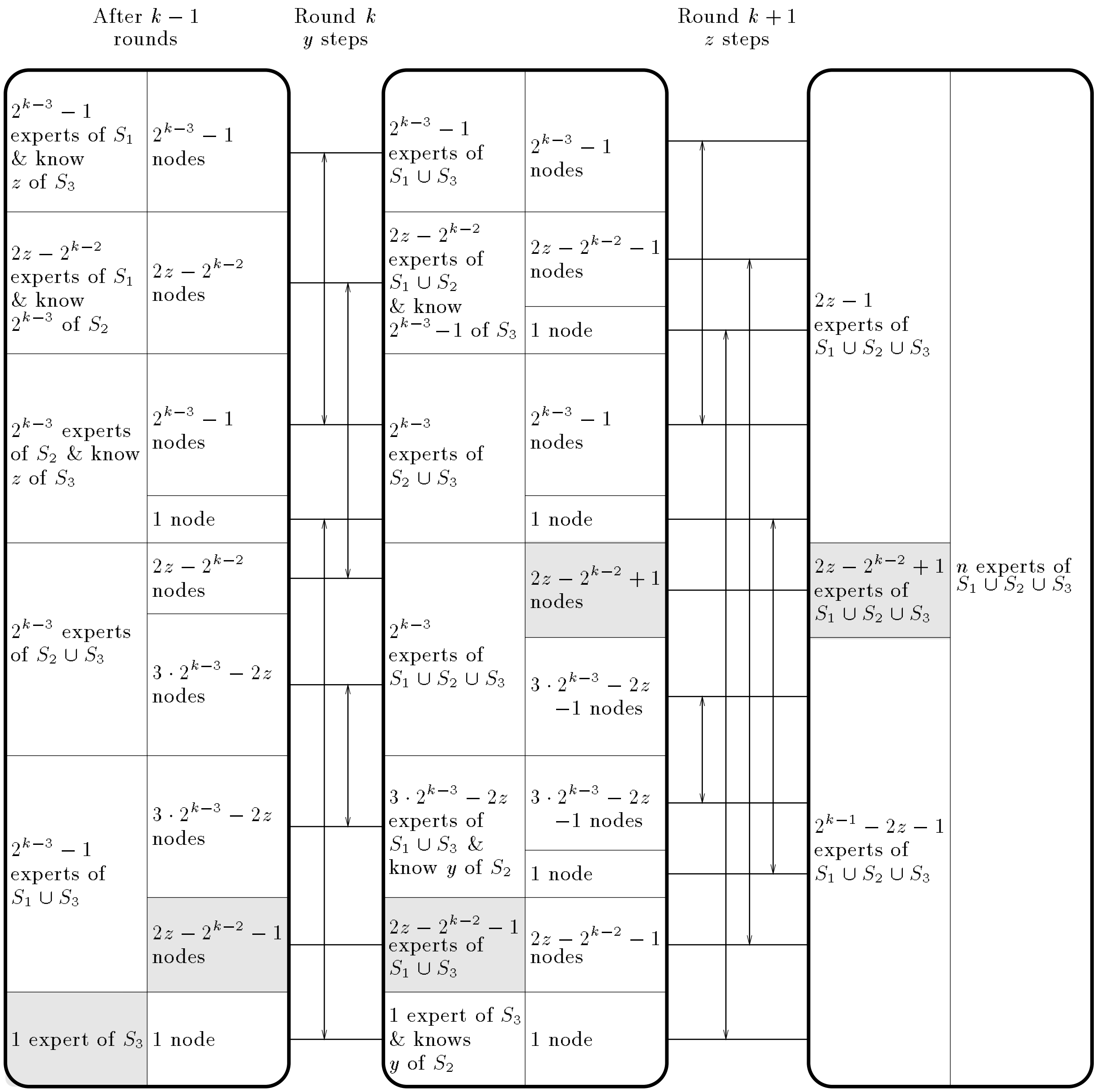

Figure 14: Last two rounds in first quarter when y is even

5 Asynchronous Gossiping

5.1 The Equal Exchange Principle

Any asynchronous gossip algorithm for n nodes with n odd takes time at least $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$ for any $\beta \geq 0$ and $\tau \geq 0$ [9]. We will derive several properties of asynchronous gossip algorithms that take time exactly $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$ and then derive a contradiction for the case $n = 2^k - 1$, and $\beta > 0$ and $\tau > 0$.

First, we need some terminology. We say that a node is *busy* at a particular time if it is in the

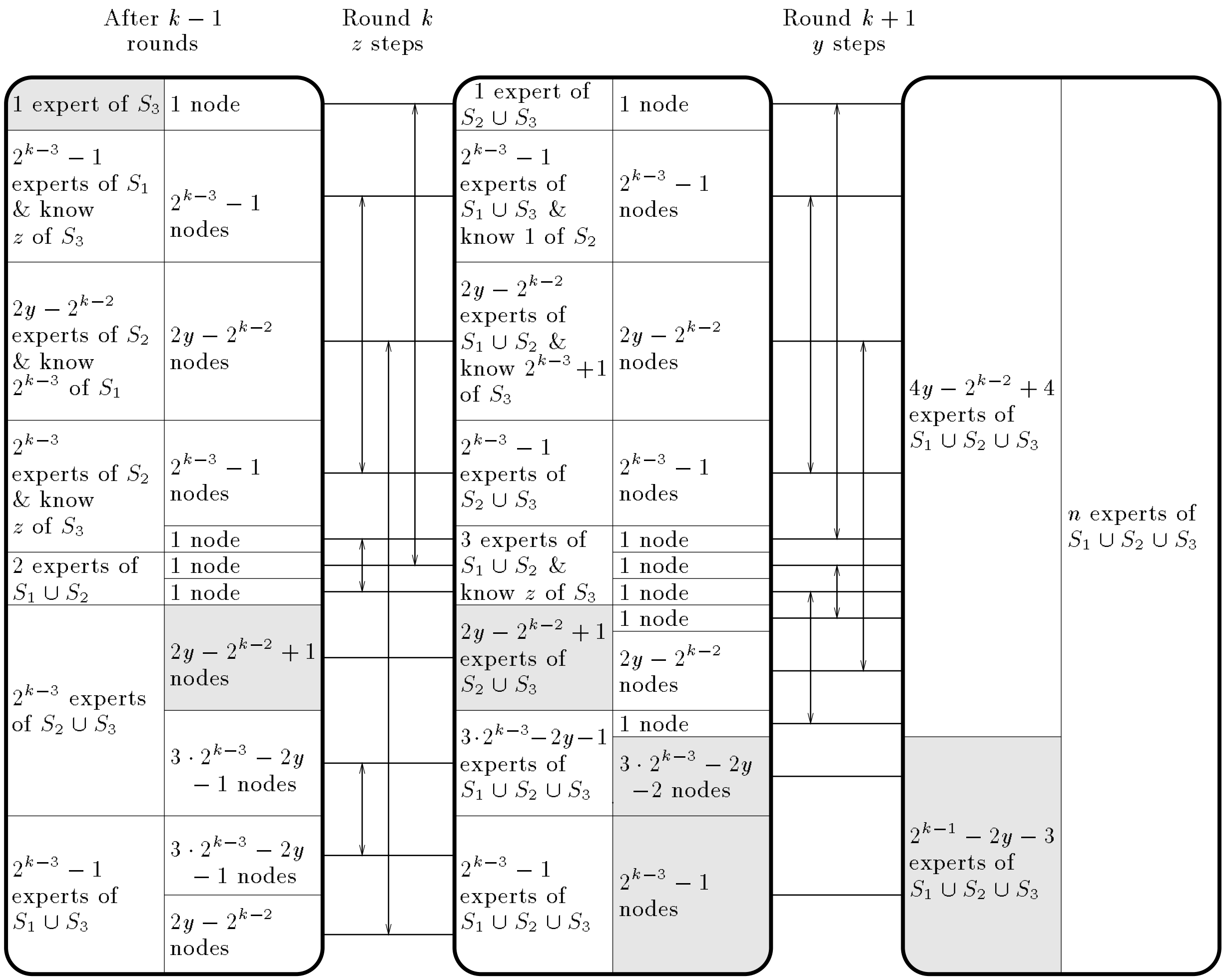

Figure 15: Last two rounds in first quarter when y is odd

start-up period of a call (which takes time β) or if it is receiving information (which takes time τ for each unit of information). A node is *idle* when it is not busy. Note that this definition of *idle* is not the same as the definition for synchronous algorithms. In the asynchronous case, there are two types of idle time. A node can be idle because it is not currently involved in a call. At any given time during an asynchronous algorithm, there is at least one node idle for this reason because communications are pair-wise. There may also be nodes which are idle waiting to start their next calls; a call cannot start until both of the nodes have completed their previous calls. This second type of idle time occurs when the two nodes involved in a call are exchanging different amounts of information. At the end of the call there will be a period when one node is considered to be idle because it is not receiving information even though it is still active sending information.

The total time of a gossip algorithm is the minimum time at which all nodes have received the information of all other nodes. When referring to the activities of any particular node, we will use the term *step* to refer a period during which the node is receiving a piece of information or the node is idle because it must wait while some other node is receiving a piece of information. A node can be idle during a step for any of the reasons described in the previous paragraph. Similarly, the term *start-up period* will refer to a period during which a node is in the start-up period of a call

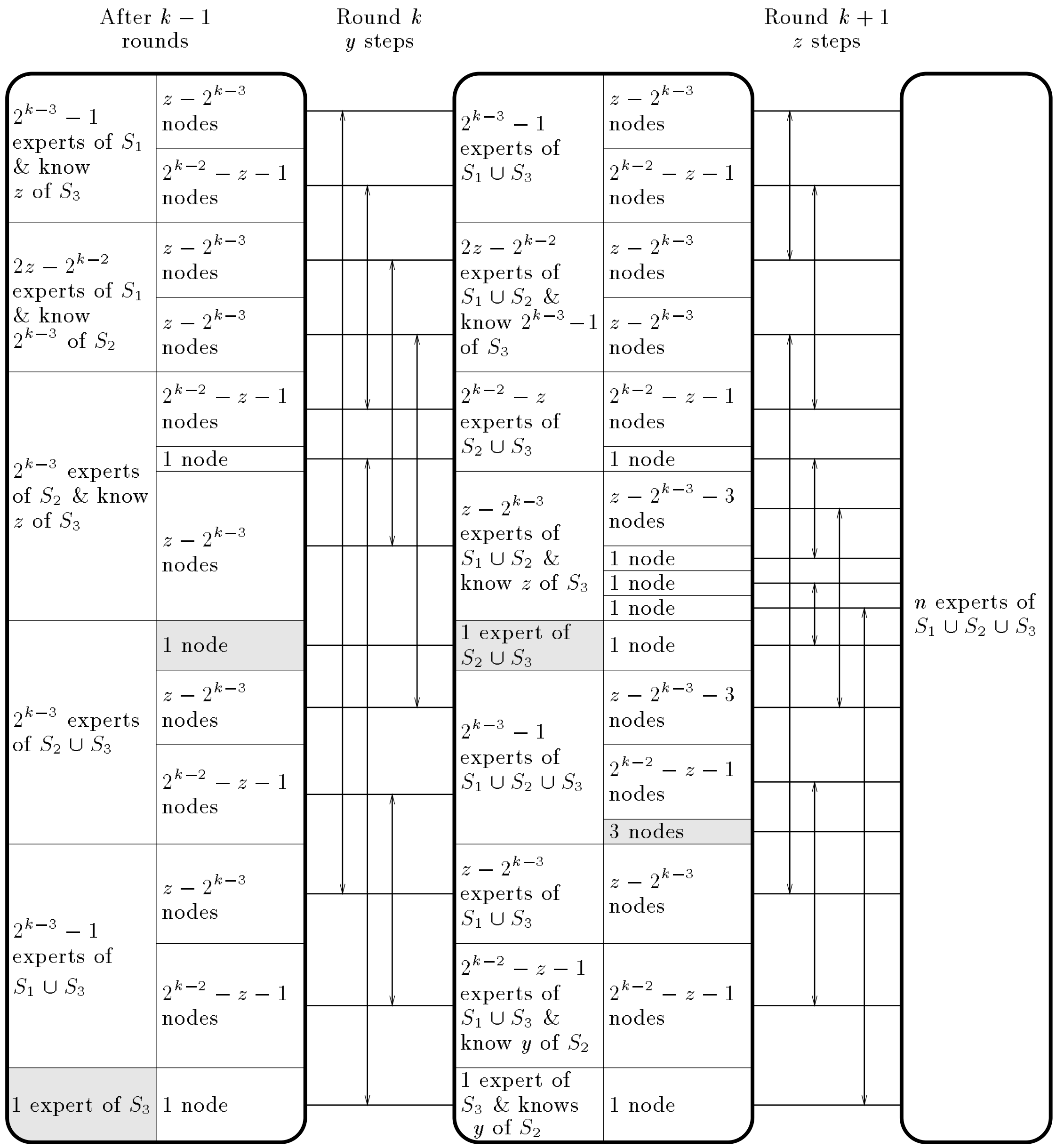

Figure 16: Last two rounds in second quarter when y is even

or a period that it is idle waiting for another node that is in the start-up period of a call. In an asynchronous algorithm, the steps and start-up periods of the nodes can occur at different times. However, we can number the steps and start-up periods of each node. In the following, when we talk about the i^{th} step of an algorithm, we are referring collectively to the i^{th} steps of all of the nodes even though these steps may occur at different times.

Now, we derive some properties of asynchronous gossip algorithms that take time exactly $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$.

Property 1 *No node can be idle during more than one step of an algorithm.*

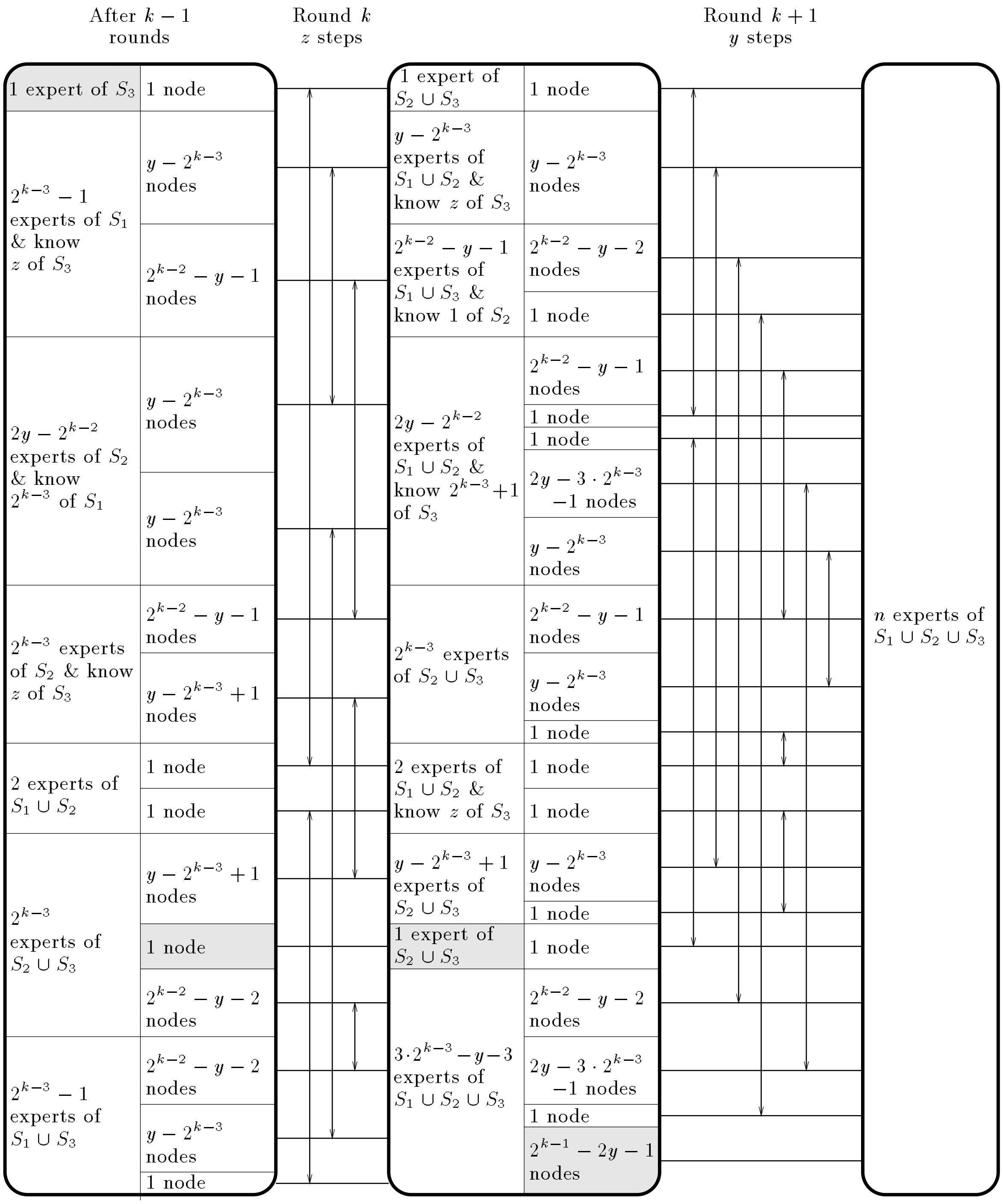

Figure 17: Last two rounds in second quarter when y is odd

Proof: Suppose some node u is idle during two or more steps. Since u needs $n-1$ steps to acquire the information of the other nodes, the total number of steps for u, and therefore for the algorithm, will be greater than n. □

Property 2 *Each node must be idle during at least one step of the algorithm.*

Proof: Suppose that some node u is never idle during a step of the algorithm. Since n is odd, and steps occur pairwise (because communications occur pairwise), there must be at least one idle node during each of the n steps of the algorithm. This means that some other node v must be idle during at least two steps, which contradicts Property 1. □

Property 3 *Two nodes cannot be idle during the same step of the algorithm.*

Proof: Each node must be idle during at least one step by Property 2 and each node needs $n-1$ (busy) steps to acquire the information of the other nodes. Summing the number of steps over all nodes gives a total requirement of n^2 steps. Since n is odd, there is at least one idle node during each step. If two nodes are idle during the same step, then the total number of steps is at least n^2+1 and this is not possible in an algorithm with n steps. □

Property 4 *Each node is idle during exactly one step, and these idle steps are distinct.*

Proof: This follows directly from the other three properties. □

Based on these properties, we get a short proof of the *Equal Exchange Principle* first proved in [9].

Theorem 5 (Equal Exchange Principle [9]) *Two nodes exchange the same amount of information when they communicate.*

Proof: Suppose two nodes u and v send different amounts of information to each other during a communication. Then one of these nodes, say u, is idle (i.e., not receiving information) during at least one step s while v is busy receiving information from u. Since the number of nodes is odd, and since communications occur between pairs of nodes, there must be another node w which is idle during the same step s. This contradicts Property 3. □

5.2 The Case $n = 2^k - 1$

The following theorem shows that the lower bound $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$ cannot be achieved by any gossip algorithm when $n = 2^k - 1$. A different proof of this result is given in [9]. The proof that we present here is much shorter and more intuitive.

Theorem 6 *Any gossip algorithm for $n = 2^k - 1$ nodes, $k \geq 3$, takes time strictly greater than $(\lceil \log_2(n) \rceil + 1)\beta + n\tau$, for all $\beta > 0$, $\tau > 0$.*

Proof: A gossip algorithm can be represented as an $n \times n$ grid. Each row represents a node and each column represents a step of the algorithm. An algorithm represented this way appears to be synchronous because the diagram does not show the start-up periods. However, the $i-th$ steps of different nodes need not occur at the same time.

By Property 4, each node must be idle during exactly one step, and these idle steps must be distinct. Without loss of generality, we can arrange the idle steps along a diagonal. Figure 18 shows the case $n = 15$ with the idle steps shown in dark gray.

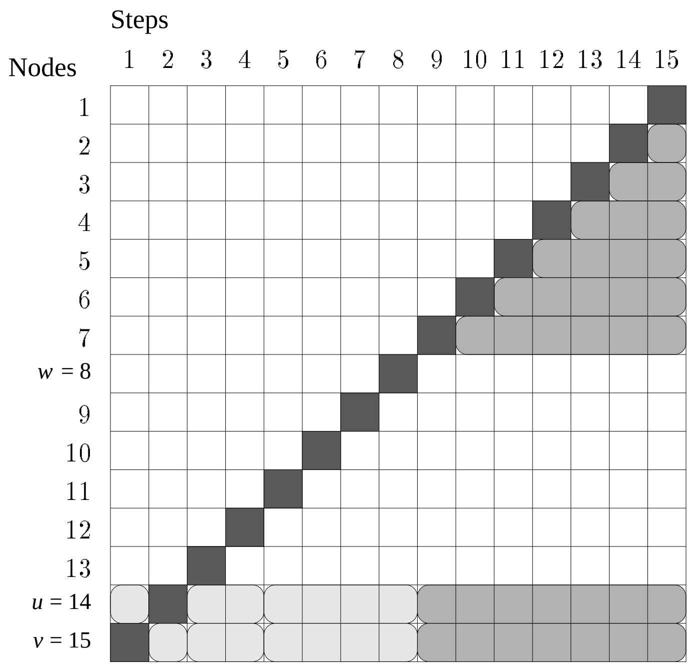

Figure 18: Asynchronous gossiping with 15 nodes

Consider the last two nodes, $u = 2^k - 2$ and $v = 2^k - 1$ (nodes 14 and 15 in Figure 18). Node v is idle during the first step while all other nodes are busy. When it starts its communication in step 2, it will inherit a delay of $\beta + \tau$ from the node with which it is communicating. So, v can only have k "active" calls and the numbers of steps in these calls must be $1, 2, 4, ..., 2^{k-2}, 2^{k-1} - 1$ as shown in Figure 18. Any increase in the number of steps in one of the first $k-1$ active calls would violate the equal exchange principle and any decrease would prevent v from acquiring enough information for its last call. The pattern for node u is the same as for node v by a similar argument.

Next, consider the $2^{k-1} - 2$ nodes labelled 2 to $2^{k-1} - 1$ (nodes 2 to 7 in Figure 18). None of these nodes is idle before step $2^{k-1} + 1$. Since the amount of information exchanged during the $i - th$ call of each of these nodes cannot be greater than 2^{i-1}, none of these nodes can start its call k later than step 2^{k-1}. Thus, these nodes have at most one call after their idle steps and each node i, $2 \leq i \leq 2^{k-1}$ must exchange exactly $i - 1$ pieces of information with another node during its last call. These communications are indicated by light gray rectangles in Figure 18. The only available nodes for these exchanges are the $2^{k-1} - 2$ nodes labelled 2^{k-1} to $2^k - 3$ (nodes 8 to 13). Therefore, exactly two of the nodes that are active during the last step must exchange i pieces of information during their last calls for each $i = 1, 2, ..., 2^{k-1} - 1$. We can show that this is impossible by examining node $w = 2^{k-1}$ (node 8). If node w has k calls before its idle step, then it must exchange $2^{k-1}-1$ pieces of information during a single call after its idle step and this gives three nodes (u, v, and w) exchanging $2^{k-1}-1$ pieces of information. If node w has $k-1$ calls before its idle step, then it can have calls k and $k+1$ after its idle step. During call k, node w cannot communicate with any of the nodes 1 through $2^{k-1}-1$ (nodes 1 to 7) because they must all start their call k no later than step 2^{k-1}. Node w cannot communicate with any of nodes $2^{k-1}+1$ to 2^k-3 (nodes 9 to 13) because this would leave three nodes with the same amount of information to exchange during their last calls (node w, the node with which w communicated in call k, and one of nodes 2 to $2^{k-1}-1$). □

6 Conclusion

We have shown that synchronous gossiping can be completed in time that matches the lower bounds for all odd values of n. This proves that the conjecture in [9] is true. We have also given a simple new method to prove the lower bound on asynchronous gossiping for $n = 2^k - 1$. The extension of this method to other values of n remains open. The trade-offs between the number of rounds and the number of steps for both synchronous and asynchronous gossiping also remain unexplored.

References

[1] Even, S. and Monien, B. On the number of rounds necessary to disseminate information, in *Proc. 1st ACM Symp. on Parallel Algorithms and Architectures*, Santa Fe, NM, June 1989, 318–327.

[2] Fertin, G. and Peters, J.G. Odd Gossiping in the Linear Cost Model. *Proc. Workshop on Communication - 23rd Int. Symp. on Mathematical Foundations of Computer Science*, Brno, Czech Republic, August 1998.

[3] Fraigniaud, P. and Lazard, E. Methods and problems of communication in usual networks. *Discrete Applied Math.* 53(1994), 79-133.

[4] Fraigniaud, P. and Peters, J.G. Minimum Linear Gossip Graphs and Maximal Linear (Δ, k)-Gossip Graphs. Technical Report 94-06, School of Computing Science, Simon Fraser Univ., 1994. (ftp://fas.sfu.ca/pub/cs/techreports/1994/CMPT94-06.ps.gz)

[5] Hedetniemi, S.M., Hedetniemi, S.T., and Liestman, A.L. A survey of gossiping and broadcasting in communication networks. *Networks* 18(1988), 319-349.

[6] Hromovič, J., Klasing, R., Monien, B., and Peine, R. Dissemination of information in interconnection networks (broadcasting and gossiping), in *Combinatorial Network Theory*, eds. D.-Z. Zhu and D.F. Hsu (Kluwer Academic Publishers, 1995) 125–212.

[7] Knödel, W. New Gossips and Telephones. *Discrete Mathematics* **13** (1975) 95.

[8] Krumme, D.W., Cybenko, G., and Venkataraman, K.N. Gossiping in minimal time. *SIAM J. Comput.* **21** (1992) 111–139.

[9] Peters, J.G. Raabe, L., and Xu, C. Odd Gossiping. Manuscript, 1996.