Multicasting in the Hypercube, Chord and Binomial Graphs

Christopher C. Cipriano and Teofilo F. Gonzalez

Department of Computer Science

University of California, Santa Barbara, CA, 93106

E-mail: {ccc,teo}@cs.ucsb.edu

March 10, 2009

Abstract

We discuss multicasting for the n-cube network and its close variants, the Chord and the Binomial Graph (BNG) Network. We present *simple* transformations and proofs that establish that the sp-multicast (shortest path) and Steiner tree problems for the n-cube, Chord and the BNG network are NP-Complete.

Keywords: NP-Complete, n-cube, binomial graphs, Chord, multicasting, Steiner trees.

1 Introduction

Multicasting is a communication primitive that allows a node in a network to send a message to multiple destination nodes. There are many ways in which multicasting can be implemented. For example, when an e-mail is sent to k destinations, e-mailing systems make k copies of the message and send each copy separately to the destinations (k unicasting operations). This is an efficient implementation when sending a message to neighbors. But when all the destinations are far away from the source, the implementation is not an efficient one. In this scenario, sending the message to one of the destinations and then forwarding it from there to the remaining $k-1$ destinations would be a more efficient solution, especially when the message being sent is large. The communication in this case is modeled by a *tree*. A tree connects (directly or indirectly) the source node to all the destination nodes, and may include other nodes in the network. There are many different multicasting trees and objective functions. In this paper we consider regular architectures without link weights (or costs). The first type of trees have the minimum number of links (edges). The problem of generating this type of trees is known as the *minimum Steiner tree (MST)* problem[1]. The second type is to minimize link usage provided that every path from the source node to a destination node is a shortest path in the original network. We refer to this problem as the *shortest path multicast(sp-multicast)* problem.

The decision version of these problems are formally defined below. The Steiner tree (ST) decision problem is given an undirected graph $G = (V, E)$, a subset of vertices, $K = \{u_0, u_1, \ldots, u_k\} \subseteq V$, and a positive integer r, find a subtree $T = (V_T, E_T)$ of G (i.e., $V_T \subseteq V$ and $E_T \subseteq E$) such that (a) $K \subseteq V_T$, and (b) the number of edges in E_T is at most r.

The sp-multicast tree decision problem is the Steiner tree decision problem with the added constraint $d_T(u_0, u_i) = d_G(u_0, u_i)$ for $1 \leq i \leq k$, where $d_T(a, b)$ and $d_G(a, b)$ is the number of edges in a shortest path from a to b in T and G, respectively.

Graham and Foulds [1] studied the MST problem for the n-cube in order to determine the possibility of computing specific biological sciences problems in reasonable time. Their work resulted in a complex proof for the NP-Completeness of the decision version of the Steiner tree problem for the n-cube. Later on, a complex transformation and proof was used to establish that the sp-multicast problem for the n-cube is NP-Complete [2, 3]. In this paper we present *simple* transformations and proofs that establish the NP-Completeness of these two problems.

An *n-cube* (*hypercube*) consists of 2^n vertices or processors. Every vertex in the n-cube is represented by an n-bit string and there is an edge between two vertices if their bit representation disagrees in exactly one bit.

For the n-cube graph we refer to the above problems as the n-cube Steiner tree problem and the n-cube sp-multicast tree problem. There is a trivial algorithm to implement optimum unicasting in the n-cube. Optimal polynomial time algorithms for unicasting have been developed for both the Chord and the binomial graph network [4, 5]; however, there has not been a lot of work on multicast in these topologies. It was conjectured

[1] Traditionally Steiner tree problems are defined for weighted graphs with the objective being to minimize the total weight of the edges in the tree.

that optimum sp-multicast trees for the binomial graph network can be constructed by simply using the unicast algorithm from the source to all destinations while choosing intermediate vertices that decrease network traffic [5]. While this explanation does describe a procedure to construct minimum sp-multicast trees, there is no known polynomial time algorithm that can implement it because there is no known efficient algorithm to choose intermediate vertices that decrease network traffic. We prove that no such polynomial time implementation exists if $P \neq NP$. In this paper we present proofs of NP-Completeness for the MST and sp-multicast tree for the Chord and the BNG by simple modifications of our new transformations for the n-cube.

BiNomial Graph (or *n-BNG*) networks consists of n vertices. The vertices are denoted $\{0, 1, \ldots, n-1\}$. Let k be the largest integer such that $2^k \leq n-1$. Every vertex i in the n-BNG network has (clockwise) edges to vertices $\{(i+2^0) \mod n, (i+2^1) \mod n, \ldots, (i+2^k) \mod n\}$ and (counterclockwise) edges to vertices $\{(i-1^0) \mod n, (i-2^1) \mod n, \ldots, (i-2^k) \mod n\}$. The n-BNG network is referred to as the *k-Chord* (or simply the *Chord*) when $n = 2^k$ for some integer $k \geq 1$.

It is simple to show that deleting some edges from an n-Chord results in an n-cube. Therefore, message communication in the n-Chord is more efficient than in the n-cube, but the number of edges (links) in the n-Chord is twice the number of edges in the n-cube and therefore more expensive to deploy. The BNG network has properties similar to the Chord.

Since any sp-multicast tree is a Steiner tree we know that an optimum solution to the n-cube Steiner tree has at most as many edges as an optimum n-cube sp-multicast tree. For some problem instances it has fewer. This also holds for the n-Chord and the n-BNG networks.

To prove our NP-Completeness results we use the vertex cover problem. The Vertex Cover (VC) decision problem is given an undirected graph $G = (V = \{1, 2, \ldots, n\}, E)$ and an integer c, find vertex cover V' with cardinality at most c, i.e., find a set of vertices V' such that $V' \subseteq V$ and every edge $e \in E$ is incident upon at least one vertex in V'.

2 NP-Completeness Results

In this section we establish our NP-Completeness results.

Theorem 2.1 *The n-cube sp-multicast tree decision problem is NP-Complete even when every vertex in $K/\{u_0\}$ is at a distance two from the source vertex u_0.*

Proof: Our polynomial time transformation from the VC decision problem is defined as follows. Let $G = (V, E)$ be an undirected graph and c a positive integer be any instance of the VC decision problem Let $n = |V|$ and $m = |E|$. We construct the instance $(K = \{u_0, u_1, \dots, u_k\}, r)$ of the n-cube sp-multicast tree decision problem as follows. The vertex u_0 is the vertex in the n-cube represented by the string of n 0-bits. For every edge $e_l = \{i, j\}$ in G we define the vertex u_l in the n-cube represented by the string of n 0-bits except for two bits that are 1-bits at positions i and j. Clearly, $k = m$ and let $r = c + k$.

We now prove our transformation is correct. Let integers $i_1, i_2, \dots, i_c$ represent the vertices in a vertex cover with cardinality c for G. Now lets define the set of vertices $\{j_1, j_2, \dots j_c\}$. Vertex j_l (in the n-cube) is represented by the string of n 0-bits except for a 1-bit at position i_l. Since every edge e_l is incident to at least one vertex in $\{i_1, i_2, \dots, i_c\}$, then vertex u_l is a neighbor of at least one vertex in $\{j_1, j_2, \dots, j_c\}$ in the n-cube. Define the sp-multicast tree MT by the set of vertices $K \cup \{j_1, j_2, \dots, j_c\}$ and the set of edges of the from $\{u_0, j_i\}$ plus one edge from each vertex u_l to a vertex in $\{j_1, j_2, \dots, j_c\}$. These edges exist in the n-cube as $\{i_1, i_2, \dots, i_c\}$ is a vertex cover for G. The number of edges in the tree is $r = k + c$. Therefore, (K, r) has an sp-multicast tree with at most r edges.

Conversely, let T be an sp-multicast tree with at most r edges for the instance (K, r). Clearly all the edges join a vertex with exactly one 1-bit to either a vertex with zero 1-bits (vertex u_0), or a vertex with exactly two 1-bits (u_l vertex). Therefore every vertex in $\{u_1, u_2, \dots, u_k\}$ has an edge to a vertex in the n-cube with exactly one 1-bit for a total of k edges. Let $\{j_1, j_2, \dots, j_f\}$ be the vertices with exactly one 1-bit in T. All of these vertices are neighbors of u_0 in the n-cube, so the f edges to join them to u_0 must be in T. In order for the tree to have at most r edges, it must be that $f \le c$. For $1 \le l \le f$ define $i_l = b$, where i_l has its 1-bit at position b. Clearly, the set $\{i_1, i_2, \dots i_f\}$ is a vertex cover for G with cardinality at most c.

□

Before we establish that the n-cube Steiner tree decision problem is NP-Complete, we establish the following two lemmas.

Lemma 2.1 *Let y be a vertex with exactly three 1-bits and let Γ be a non-empty subset of vertices each* *with exactly two 1-bits that are neighbors of y in the n-cube. Let γ be the number of vertices in Γ. Then, $1 \leq \gamma \leq 3$, and there is an sp-multicast tree rooted at $u_0 = 00\ldots0$ that includes all the vertices in Γ with at most $\lfloor 4\gamma/3 \rfloor + 1$ edges.*

Proof: Figure 1, after deleting all the vertices of the form $A1B*$, shows an sp-multicast tree for the case when γ equals to 3 with 5 edges which is $\lfloor 4\gamma/3 \rfloor + 1$. When γ equals two, the two elements in Γ must have a 1-bit at the same position. Therefore there is an sp-multicast tree with 3 edges, which is $\lfloor 4\gamma/3 \rfloor + 1$. When γ equals to one there is an sp-multicast tree with 2 edges, which is $\lfloor 4\gamma/3 \rfloor + 1$.

□

Lemma 2.2 *Let x be a vertex with exactly four 1-bits and let Γ be a non-empty subset of vertices each with exactly two 1-bits which are in common with the 1-bits in x. Let γ be the number of vertices in Γ. Then, $1 \leq \gamma \leq 6$ and there is an sp-multicast tree rooted at u_0 that includes all the vertices in Γ with at most $\gamma + 3$ edges.*

Proof: Figure 1 shows an sp-multicast tree for the case when γ equals to 6 with $\gamma + 3$ edges. When γ is less than six just delete from Figure 2 the vertices that are not in Γ and that is a multicast tree with at most $\gamma + 3$ edges.

□

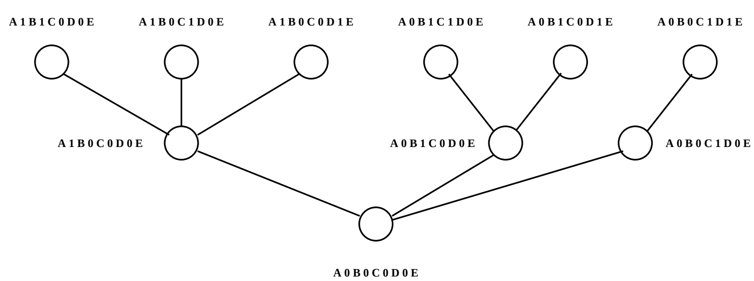

Figure 1: An sp-multicast tree for u_0 and six vertices each with two 1-bits in two of four possible positions. The symbols, A, B, C, D, and E represent strings of zeros.

Theorem 2.2 *The n-cube Steiner tree decision problem is NP-Complete.*

Proof: Our polynomial time transformation is the same one as the one used in Theorem 2.1. To establish that this is a valid transformation we use the proof of Theorem 2.1 and prove that if there is a Steiner tree with at most r edges, then there is also an sp-multicast tree with at most r edges.

Let $f(I)$ be any problem instance generated by the polynomial transformation. Let ST be a Steiner tree with at most r edges that is not an sp-multicasting tree. Assume without loss of generality that when viewing the tree ST as a tree rooted at u_0 all its leaves are elements of the set $\{u_1, u_2, \ldots, u_k\}$. We now show that instance $f(I)$ has an sp-multicast tree with at most r edges. Let $sp(ST)$ be the number of vertices in $\{u_1, u_2, \ldots, u_k\}$ that have a path in ST to u_0 with exactly two edges. Clearly $sp(ST) < k$. Our approach is to show that ST can be transformed into another Steiner tree ST' with at most r edges such that $sp(ST') > sp(ST)$. After applying this argument at most k times we know that instance $f(I)$ has an sp-multicast tree with at most r edges.

Let $u \in \{u_1, u_2, \ldots, u_k\}$ be a vertex whose shortest path in ST from u_0 to u is the largest. Let P be that shortest path in ST that starts at u_0 and ends at u. Clearly, path P has more than two edges. Let w, x and y be the last three vertices just before u in path P, i.e., the path from u_0 to u visits first vertex w, then it is followed by the edges $\{w, x\}$, $\{x, y\}$ and $\{y, u\}$, to reach vertices x, y and u in that order.

If the number of 1-bits of y is equal to 1, then define ST' as ST after deleting edge $\{x, y\}$ and adding edge $\{u_0, y\}$. Consider now the case when the number of 1-bits of y is equal to three. The subtree ST_w is defined as ST after deleting all the subpaths originating at vertex u_0 that do not include vertex w. It is convenient to visualize ST_w as a tree rooted at w. All the neighbors of w are said to be the children of w. A similar relationship holds for all the children of w and so on. Every leaf in ST_w is at a distance at most three from w and it is a vertex in $\{u_1, u_2, \ldots, u_k\}$. There are two cases depending on the number of 1-bits of x.

Case 1: The number of 1-bits of x is two. Let α be the number of children of y in ST_w. Since the parent of y and all the children of y in ST_w have exactly two 1-bits and y has three 1-bits, it must be that $1 \leq \alpha \leq 2$. By Lemma 1 we know there is an sp-multicast tree rooted at u_0 that includes all the children of y in ST_w with $\alpha + 1$ edges (as $\alpha \leq 2$). Define ST' as ST after deleting edges incident to vertex y and adding the sp-multicast tree just defined.

Case 2: The number of 1-bits of x is four. Let β be the number of children of x in ST_w Let α be the number of vertices in $\{u_1, u_2, \ldots, u_k\}$ that are descendants of x in the subtree ST_w. If $\beta = 1$, then α is at most three. By Lemma 1 we know there is an sp-multicast tree rooted at u_0 that includes all the children of y in ST_w with $\alpha+2$ edges. Define ST' as ST after deleting the edge $\{w,x\}$ as well as all the edges incident to vertex y and adding the sp-multicast tree just defined. On the other hand if $\beta > 1$, then by Lemma 2 we know there is an sp-multicast tree rooted at u_0 that includes all the leaves that are descendants of x in ST_w with $\alpha+3$ edges, as x has four 1-bits. Define ST' as ST after deleting the edge $\{w,x\}$ as well as all the edges from x to a descendant of x in ST_w and adding the sp-multicast tree just defined.

In all cases ST' does not have more edges than ST and $sp(ST') > sp(ST)$. Eventually $sp(ST')$ will be equal to k and ST' will be an sp-multicast tree. This concludes the proof of the theorem.

□

We now establish that the Chord sp-multicast tree decision problem is NP-Complete.

Theorem 2.3 *The Chord sp-multicast and Steiner tree decision problems are NP-Complete even when every vertex in $K/\{u_0\}$ is at a distance two from the source vertex u_0.*

Proof: The reductions are similar to the ones in the previous theorems. The difference is that between every pair of bits of the vertices in K in the previous reduction, which we call *box bits*, we add a bit pattern called the *signature*.

Let $G = (V,E)$ be an undirected graph and c a positive integer be any instance of the VC decision problem. Let $n = |V|$ and $m = |E|$. We construct the instance $(K = \{u_0, u_1, \ldots, u_k\}, r)$ of the t-Chord multicast tree decision problem as follows. Let $k = m$, $r = c+m$, and $t = n+(n-1)*(2r+3)$. Every vertex in K in our reduction consists of n box bits and $n-1$ signatures arranged in the order $b, s, b, s \ldots, b, s, b$, where b is a box bit, and s is the signature. The signature is the $2r+3$ bit pattern $0101\ldots010$. Vertex u_0 in the t-Chord has all the box bits equal to zero. For every edge $e_l = \{i,j\}$ in G we define the vertex u_l in the t-Chord with all the box bits equal to zero, except for the i^{th} and j^{th} box bits which are 1-bits. Therefore, k is equal to m.

The proof that the transformation is correct is based is based on Theorems 2.1 and 2.2, and the argument that in a tree with r edges one cannot change two or more box bits in one step. The reason for this is that in order to change two or more box bits one must make at least one of the signatures equal to all zeros or all ones. But each signature has $r+1$ 1-bit runs and at each step one can reduce the number of 1-bit runs in a signature by at most one. Since the whole tree has at most r edges, transforming one signature into all ones or all zeros is not possible as this would take more than r edges.

$\square$

3 Discussion

We presented simple proofs to establish that the Steiner and sp-multicast tree decision problems on the n-cube, Chord and BNG networks are NP-Complete. Our reductions and the ones in Refs. [1, 2, 3] define problem instances where the number of bits is the n-cube is proportional to size n of the NP-Complete problem being reduced. However this implies that the n-cube has 2^n vertices, though at most $O(n^2)$ vertices are used as input to the n-cube problem. An important open problem is to determine whether or not our problems remain NP-Complete when the reduction is for an n-cube with $O(P(n))$ vertices, where $P(n)$ is a polynomial on n. Gonzalez and Serena [6] have shown that some problems defined over the hypercube are NP-Complete even under this condition. Our reductions can be extended to the (k, n)-cube network, where links exist between vertices that differ in at most k bits in their binary representation.

References

[1] Graham, R. L., and Foulds, L. R., "Unlikelihood that Minimal Phylogenies for a Realistic Biological Study can be Constructed in Reasonable Computation Time", *Mathematical Biosciences,* 60, 133 – 142, 1982.

[2] Choi, H.-A., and Esfahanian, A. H., "On Complexity of a Message-Routing Strategy for Multicomputer Systems," *Lecture Notes in Computer Science*, R.H. Mohring (Ed.), 484, 170 – 181, Springer-Verlag, 1991.

[3] Choi, H.-A., Esfahanian, A. H., and Houck, B., "Optimal Communication Trees with Application to Hypercube Multiprocessors," *Graph Theory, Combinatorics, and Applications,* Alavi, Y, Chartrand, G., Oellermann, O. R., and Schwenk, A. J. (Eds.), 1, 245 – 264, 1991.

[4] Ganesan, P. and Manku G. S. "Optimal Routing in Chord," Proceeding of the ACM-SIAM Symposium on Distributed Algorithms (SODA), 2004.

[5] Angkun, T., Bosilca, G., Vander Zanden, B., and Dongarra, J., "Optimal Routing in Binomial Graph Networks," Proceedings of the Eight International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'07), 363 –369, 2007.

[6] Gonzalez, T. F. and Serena, D., Pairwise Disjoint Shortest Paths in the n-Cube and Related Problems, *Theoretical Computer Science,* Vol 369, No. 1, pp. 427 – 435, 2006.