A Distributed Algorithm for Min-Max Tree and Max-Min Cut Problems in Communication Networks

Song Guo, *Member, IEEE*, and Victor C. M. Leung, *Fellow, IEEE*

***Abstract*—We consider the problem of finding a multicast tree rooted at the source node and including all the destination nodes such that the maximum weight of the tree arcs is minimized. It is of paramount importance for many optimization problems, *e.g.,* the maximum-lifetime multicast problem in multihop wireless networks, in the data networking community. We explore some important properties of this problem from a graph theory perspective and obtain a min-max-tree max-min-cut theorem, which provides a unified explanation for some important while separated results in the recent literature. We also apply the theorem to derive an algorithm that can construct a global optimal min-max multicast tree in a distributed fashion. In random networks with n nodes and m arcs, our theoretical analysis shows that the expected communication complexity of our distributed algorithm is in the order of $O(m)$. Specifically, the average number of messages is $2(n-1-\gamma)-2\ln(n-1)+m$ at most, in which γ is the Euler constant. To our best knowledge, this is the first contribution that possesses the distributed and scalable properties for the min-max multicast problem and is especially desirable to the large-scale resource-limited multihop wireless networks, like sensor networks.**

***Index Terms*—Communication complexity, distributed algorithm, min-max tree, Steiner tree.**

I. Introduction

THE problem that this paper investigates—namely, the min-max Steiner tree problem—is one that arises naturally from several optimization problems in the data networking community. Given a weighted directed graph, a source node, and a set of destination nodes, the min-max Steiner tree problem is to determine a Steiner tree rooted at the source and including all the destination nodes such that the maximum weight of the tree arcs is minimized over all possible Steiner trees. In a wired network, this optimization problem would intend to find a maximum bandwidth tree, *e.g.,* the work [1], in which the arc-weight represents the negative of link bandwidth. The same problem in a wireless network received attention for finding a maximum-lifetime tree, *e.g.,* the works [2], [3], and [9]–[11], in which the arc-weight represents the reciprocal of lifetime of wireless communication link.

We consider this optimization problem in a general network that is modeled as a weighted graph with n nodes and m arcs. A special case of the problem is to determine the min-max path, which is to find a path from the source node to the destination node that minimizes the maximum arc cost on the path. In the literature, the min-max path problem arises in the implementation of Edmonds and Karp's "maximum capacity augmentation" version of the Ford–Fulkerson maximum network flow algorithm [4].

In an undirected graph, the min-max spanning (broadcast) and Steiner (multicast) tree problems can be solved by the algorithms proposed in [5] and [6], respectively, in a time complexity of $O(m)$. These results, however, cannot be simply extended to the directed graph scenarios, *i.e.*, the trees obtained from the above algorithms on a directed graph are not necessary to be optimal.

The algorithm discovered by Camerini in [5] is the first one that obtains the optimal solutions for the min-max spanning tree problem in a directed graph. It runs in $O(m\log n)$ time. This result has been improved by the algorithms proposed by Gabow and Tarjan [7] to an $O(\min(n\log n+m, m\log^* n))$ time, where $\log^*$ is recursively defined as $\log^{(0)} x = x$, $\log^{(i+1)} x = \log(\log^{(i)} x)$, and $\log^* x = \min\{i|\log^{(i)} x \le 1\}$. The proposed algorithms can be also extended to the Steiner tree case with the same time complexity. Recently, Georgiadis gave another optimal solution for the directed min-max Steiner tree problem based on a variant implementation of Dijkstra's algorithm. It runs in $O(T(m))$ time [8], in which $T(m)$ is the time needed to sort the arc costs. Note that the above results can be applied to the undirected graph case.

This min-max Steiner tree problem has been receiving new and tremendous attention recently in multihop wireless networks in forms of ad hoc network or sensor network. In such networks, energy resource supplied by batteries is likely to be scarce and even entirely nonrenewable in some applications. In order to extend the network operating lifetime, the maximum-lifetime multicast problem, or equivalently the directed min-max Steiner tree problem, is of paramount importance for the wide deployment of multihop wireless networks. In the recent literature, some algorithms have been proposed to maximize the lifetime for broadcasting (*e.g.,* [9], [10]) and multicasting (*e.g.,* [11]–[14]), respectively.

Most of the existing solutions for the min-max Steiner tree problem are centralized, meaning that the global network information must be propagated throughout the network to a certain node in order to construct a Steiner tree. Unfortunately, such solutions are not practical for some specific problems with certain constraints, *e.g.,* the lifetime optimization problem in large-scale multihop wireless networks in which each node has limited energy, bandwidth, memory, and computation capabilities. The only possible distributed solution is to use the distributed minimum spanning tree algorithm proposed in [15] and

Manuscript received November 01, 2007; revised September 17, 2008; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor Z.-L. Zhang.

S. Guo is with the School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu 965-8580, Japan (e-mail: sguo@ u-aizu.ac.jp).

V. C. M. Leung is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: vleung@ ece.ubc.ca).

Digital Object Identifier 10.1109/TNET.2009.2038998

[16] for the special case of finding min-max spanning tree in undirected graph. This conclusion is based on the well-known findings in [5] that the minimum spanning tree is a min-max spanning tree, but not vice versa, in undirected graphs. The distributed minimum spanning tree algorithms [15], [16] have the communication complexity of $O(n \log n+m)$, which is the best result so far.

This motivates us to reexamine the general problem, i.e., the directed min-max Steiner tree problem, and to design the distributed algorithm that can run on a resource-limited network. The desired distributed algorithm should be able to construct a global optimal directed min-max Steiner tree with a low communication complexity. In the rest of the paper, we only consider the general case, in which the term min-max tree implicitly refers to directed min-max Steiner tree. In order to refer to the special cases, we shall add determinatives like spanning, undirected, *etc*.

The contributions of this paper are summarized as follows.

1) We have discovered by the first time a min-max-tree max-min-cut theorem in directed graphs. It shows the equivalence of two optimization problems, *i.e.,* the *min-max tree problem* and the *max-min cut problem*. Some results with regard to the min-max tree problem that previously seem unrelated, *e.g.,* [8]–[14], now can be explained under the same philosophy given by this theorem. Furthermore, we have attempted to reveal the relationship of this theorem with the famous max-flow min-cut theorem and obtained some preliminary but important results.
2) Applying this theorem, we derive an optimal algorithm that can solve this optimization problem in a distributed fashion. It is especially favorable for a resource constrained network, like sensor networks.
3) We also present a theoretical analysis to show that the proposed distributed algorithm can achieve an expected linear message complexity $O(m)$. This not only appears to be the first result for the general min-max tree problem, but also improves the best performance $O(n \log n + m)$ [15], [16] known so far in the literature, in terms of average message complexity, for the special case of the undirected min-max tree problem.

The remainder of this paper is organized as follows. Section II gives the problem formulation. Section III derives a min-max-tree max-min-cut theorem and gives some applications of this theorem. Section IV then presents a distributed optimal min-max tree algorithm. Section V analyzes the communication complexity of the algorithm. Section VI summarizes our findings and points out future research problems.

II. Problem Statement

We model a general network as a simple directed graph $G(N, A, w)$ with a finite node set $N(n = |N|)$, an arc set $A(m = |A|)$, and a weight function w that assigns a nonnegative real number w_{vu} to each arc (v, u). Most communication networks support bidirectional links, resulting in the corresponding directed graphs *symmetric, i.e.,* for every arc that belongs to G, the corresponding inverted arc also belongs to G. Note that because the undirected graph is a special case of the symmetric directed graph when $w_{vu} = w_{uv}$ for each arc (v, u), the conclusions achieved in this paper can apply to the case of undirected graph directly. In the rest of the paper, we focus our study on the symmetric directed graph and finally extend the results to the general directed graph.

We consider a source-initiated multicast with multicast members $M = \{s\} \cup D$, where s is the source node and D is the set of destination nodes. All the nodes involved in the multicast form a multicast tree rooted at the node s, *i.e.,* a rooted tree T_s, with a tree node set $N(T_s)$ and a tree arc set $A(T_s)$. We define a rooted tree as a directed acyclic graph with a source node with no incoming arcs, and each other node v has exactly one incoming arc. A node with no outgoing arcs is called a leaf node, and all other nodes are internal nodes (also called relay nodes).

A directed path π_{vu} on graph G can be defined by a sequence of adjacent links from v to u. An important property of a rooted tree is that for any node v in the rooted tree T_s, there must exist a unique directed acyclic path π_{sv} in the tree. Let $\delta(T_s)$ denote the maximum weight of the tree links, *i.e.,*

$$\delta(T_s) \equiv \max\{w_{xy} | (x, y) \in A(T_s)\}. \tag{1}$$

A tree link (v, u) is called the bottleneck link of the tree T_s if $w_{vu} = \delta(T_s)$. Similarly, we use $\delta(\pi_{vu})$ to denote the maximum weight of the links on a directed path π_{vu}, *i.e.,*

$$\delta(\pi_{vu}) \equiv \max\{w_{xy} | (x, y) \in \pi_{vu}\}. \tag{2}$$

Now the formal definition of the min-max tree problem can be formulated as follows.

Definition 1: The *min-max tree problem* is to determine a directed multicast tree T_s^* rooted at the source s and including all the multicast members (*i.e.,* $M \subseteq N(T_s^*)$) such that $\delta(T_s^*)$ is minimized, *i.e.,*

$$\delta_{\min} \equiv \delta(T_s^*) = \min_{T_s} \delta(T_s). \tag{3}$$

III. A Min-Max-Tree Max-Min-Cut Theorem

Given a multicast request (s, D), we say the link set C_X, including all links crossing the node partition $(X, N - X)$, is a (s, D)-cut if the first node set X includes at least the source node s and the second node set $N - X$ includes at least one destination node, *i.e.,*

$$C_X \equiv \{(x, y) | x \in X \wedge y \in N - X \wedge s \in X \wedge D \not\subset X\}. \tag{4}$$

In the rest of the paper, the prefix (s, D) of the term (s, D)-cut is omitted for simplicity without confusion. Let $\psi(C_X)$ denote the minimum weight of the cut links, *i.e.,*

$$\psi(C_X) \equiv \min\{w_{xy} | (x, y) \in C_X\}. \tag{5}$$

We now introduce, by the first time, another problem that is closely related to the *min-max tree problem*.

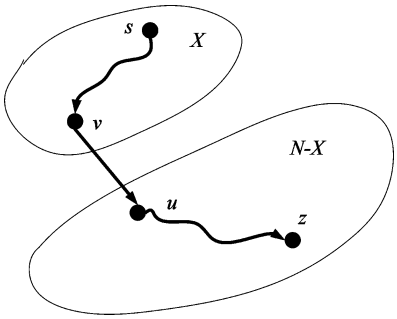

Fig. 1. Illustration of the proof for Lemma 1. The arrowed line denotes the directed tree link and arrowed curve denotes the directed tree path.

Definition 2: The *max-min cut problem* is to determine a cut C_X^* such that $\psi(C_X^*)$ is maximized, *i.e.,*

$$\psi_{\max} \equiv \psi(C_X^*) = \max_{C_X} \psi(C_X). \tag{6}$$

In the following, we provide a sufficient and necessary condition for the min-max tree problem, and then derive a min-max-tree max-min-cut theorem that eventually shows the equivalence of these two optimization problems.

Lemma 1: If there exists a directed multicast tree in G, then for any cut C_X

$$\delta_{\min} \geq \psi(C_X). \tag{7}$$

Proof: Note that there is at least one destination node $z (z \in D)$ belonging to the node set $N - X$, *i.e.,* $z \in D \cap (N - X)$, based on the cut definition. Let T_s^* be a min-max tree of G. There must exist an arc $(v, u) \in A(T_s^*)$ connecting the node sets X and $N - X$ (*i.e.,* $v \in X$ and $u \in N - X$) as shown in Fig. 1 in order to satisfy the requirement that is there exists a directed path from s to the destination node z in the tree. Therefore, we can obtain $\delta_{\min} = \delta(T_s^*) = \max\{w_{xy} | (x, y) \in A(T_s^*)\} \geq w_{vu} \geq \min\{w_{xy} | (x, y) \in C_X\} = \psi(C_X)$. ■

Because the inequality given in (7) is for any cut, it must hold for the max-min cut C_X^* as well. We then have the following conclusion.

Corollary 1:

$$\delta_{\min} \geq \psi_{\max}. \tag{8}$$

Theorem 1: The multicast tree T_s is a min-max tree if and only if there exists a cut C_X such that $\delta(T_s) \leq \psi(C_X)$.

Proof: We first prove the "if" case. Let T_s' be any multicast tree. For the same reason as stated in the proof of Lemma 1, there must exist at least one tree link $(v, u) \in A(T_s')$ connecting node sets X and $N - X$ (*i.e.,* $v \in X$ and $u \in N - X$). The definition of $\psi(C_X)$ given in (5) implies $\psi(C_X) \leq w_{vu}$. Furthermore, since $(v, u) \in A(T_s')$, its weight must not be greater than the weight of a bottleneck link, *i.e.,* $w_{vu} \leq \delta(T_s')$. Therefore, from the given condition in the theorem and the above derived inequalities, we have $\delta(T_s) \leq \psi(C_X) \leq w_{vu} \leq \delta(T_s')$ for any T_s', *i.e.,* T_s is a min-max tree.

We then prove the "only if" case. Let T_s be a min-max tree. We construct a cut C_X as follows:

$$X \equiv \{x | \exists \pi_{sx}, \delta(\pi_{sx}) < \delta(T_s)\}. \tag{9}$$

In the following, we first explain that cut C_X must satisfy the cut definition given by (4) as follows. The source s is included in X trivially. We show by contradiction that there must exist at least one destination node in the set $N - X$. If we assume $D \subseteq X$, then we can construct a multicast tree T_s' by superposing all the directed path π_{sx}, from the source s to each individual destination node x, that satisfies the condition $\delta(\pi_{sx}) < \delta(T_s)$, resulting in $\delta(T_s') < \delta(T_s)$. This contradicts the fact that T_s is a min-max tree.

Now, we prove that the constructed cut above satisfies the inequality in Theorem 1. Let $(v, u) \in C_X$ be the cut link with minimum weight, *i.e.,* $w_{vu} = \psi(C_X)$. It must be true that $w_{vu} \geq \delta(T_s)$. This can be explained by contradiction as follows. Since $v \in X$, there must exist a directed path π_{sv} that satisfies $\delta(\pi_{sv}) < \delta(T_s)$ based on the construction rule given in (9). If $w_{vu} < \delta(T_s)$, we can obtain a directed path π_{su} by appending the link (v, u) to the end of path π_{sv}, *i.e.,* $\pi_{su} = \pi_{sv} + (v, u)$. Therefore, node u satisfies the condition $\delta(\pi_{su}) < \delta(T_s)$ in (9) as well and should also be included in set X. This contradicts the observation of $v \in X$ and $u \in N - X$. Finally, we achieve that $\delta(T_s) \leq w_{vu} = \psi(C_X)$. ■

Corollary 2:

$$\delta_{\min} \leq \psi_{\max}. \tag{10}$$

Proof: Let T_s be a min-max tree. Equation (9) gives a way to construct X such that $\delta(T_s) \leq \psi(C_X)$. From the definition of the max-min cut, we then have $\delta_{\min} = \delta(T_s) \leq \psi(C_X) \leq \psi_{\max}$. ■

Corollary 3:

$$\delta_{\min} = \psi_{\max}. \tag{11}$$

Proof: It can be derived from Corollaries 1 and 2 directly. ■

Finally, we have the following min-max-tree max-min-cut theorem. To our best knowledge, this is the first theorem that shows the equivalence of the *min-max tree problem* and the *max-min cut problem*. We shall see later that it can be used to derive optimal algorithms for the min-max tree problem.

Theorem 2: (The Min-max-tree Max-min-cut Theorem) If there exists a multicast tree T_s and a cut C_X such that $\delta(T_s) = \psi(C_X)$, then T_s is a min-max tree and C_X is a max-min cut.

Proof: From Lemma 1, we have $\delta_{\min} \geq \psi(C_X)$. On the other hand, we have $\delta(T_s) \geq \delta_{\min}$ from (3). From the given condition $\psi(C_X) = \delta(T_s)$, we obtain $\delta(T_s) = \delta_{\min}$, *i.e.,* T_s is a min-max tree. Finally, we achieve the equation $\delta_{\min} = \delta(T_s) = \psi(C_X) = \psi_{\max}$ from Corollary 3 directly, *i.e.,* C_X is a max-min cut. ■

We shall have the immediate result that Prim's algorithm [17] can generate a directed min-max tree from the above theorem. Recall that the standard Prim's algorithm maintains throughout its execution a single tree rooted at the source node. Initially, the rooted tree includes the source node only. Subsequently, new nodes are added to the tree iteratively one at a time on a minimum weight basis until all nodes are included in the tree. Let T_s^i be the tree after the ith iteration of the tree incremental formation, where $i = 0, 1, \ldots, n-1$. The initial tree only includes the source, *i.e.,* $N\left(T_s^0\right) = \{s\}$ and $A\left(T_s^0\right) = \phi$.

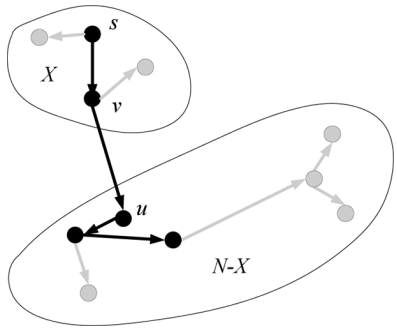

Fig. 2. Illustration of the proof for Theorem 3. The dark nodes and arcs indicate the multicast tree, and the light nodes and arcs indicate the branches pruned from the spanning tree T_s^{n-1}.

Theorem 3: The multicast tree obtained by running Prim's algorithm on a directed graph and pruning the resulting spanning tree is a min-max tree.

Proof: Let T_s be the final multicast tree pruned from the spanning tree T_s^{n-1} obtained by Prim's algorithm as shown in Fig. 2. We assume that (v,u) is a bottleneck link of T_s that is added into the tree just after the ith iteration. We define a node set

$$X \equiv N\left(T_s^i\right), \quad 0 \le i \le n-1 \tag{12}$$

such that its corresponding cut C_X obviously satisfies the cut definition given by (4). Note that link (v,u) is added to the tree on a minimum weight basis, *i.e.,* $w_{vu} = \psi(C_X)$. This leads to $\delta(T_s) = w_{vu} = \psi(C_X)$. From the min-max-tree max-min-cut theorem, we can conclude that T_s is a min-max tree. ■

Corollary 4: A pruned minimum spanning tree in undirected graph is an undirected min-max tree.

Proof: The result is achieved directly from Theorem 3 since the tree obtained from Prim's algorithm on an undirected graph is a minimum spanning tree. ■

The min-max-tree max-min-cut theorem achieves the same result for the revised Dijkstra's algorithm [8]. The standard Dijkstra's algorithm [18] works by keeping for each node v the cost $d[v]$ of the shortest path found so far between s and v. Initially, this value is 0 for the source node s ($d[s] = 0$) and infinity for all other nodes. Similar to Prim's algorithm, new nodes are added to the tree iteratively one at a time on a minimum cost basis until all nodes are included in the tree. After adding a new node, it is followed by the edge relaxation operation [18] for each node outside the tree.

The revised Dijkstra's algorithm [8] changes the relaxation operation given in (13) to the new one given in (14) as follows:

$$\text{if}(d[u] + w_{uv} < d[v])\ d[v] := d[u] + w_{uv} \tag{13}$$

$$\text{if}(\max(d[u], w_{uv}) < d[v])\ d[v] := \max(d[u], w_{uv}) \tag{14}$$

in which u is the new node included into the tree and v is any node outside the tree. Based on this revision, we have the following observation.

Observation 1: At the beginning of each ith iteration ($1 \le i \le n-1$), the following equations are always held:

$$d[x] = \delta(\pi_{sx}), \quad \forall\, x \in N\left(T_s^i\right) \tag{15}$$

$$d[y] = \min_{x \in N(T_s^i)} (\max(\delta(\pi_{sx}), w_{xy})), \quad \forall\, y \in N - N\left(T_s^i\right) \tag{16}$$

$$d[x] \le d[y], \quad \forall x \in N\left(T_s^i\right),\ \forall y \in N - N\left(T_s^i\right). \tag{17}$$

Theorem 4: The multicast tree obtained by running the revised Dijkstra's algorithm on a directed graph and pruning the resulting spanning tree is a min-max tree.

Proof: Let T_s be the final multicast tree pruned from the spanning tree T_s^{n-1} obtained by the revised Dijkstra's algorithm. We assume that (v,u) is a bottleneck link of T_s that is added into the tree just after the ith iteration. Note that the link (v,u) is added to the tree on a minimum cost basis, *i.e.,* $d[u] \le d[b]$ for any node $b \in N - N(T_s^i)$.

Let (a,b) be any link that connects node sets $X \equiv N\left(T_s^i\right)$ and $N - X$, *i.e.,* $(a,b) \in C_X$. Since node b is outside the tree T_s^i, by substituting b into (16), we thus obtain

$$d[b] = \min_{x \in N(T_s^i)} (\max(\delta(\pi_{sx}), w_{xb})) \le \max(\delta(\pi_{sa}), w_{ab}). \tag{18}$$

Therefore, we have

$$d[u] \le d[b] \le \max(\delta(\pi_{sa}), w_{ab}). \tag{19}$$

On the other hand, because node u is outside the tree T_s^i and node a is inside the tree T_s^i, we obtain the following inequality from (17) and (15) directly:

$$d[u] \ge d[a] = \delta(\pi_{sa}). \tag{20}$$

Considering both (19) and (20), we conclude that

$$d[u] \le w_{ab}. \tag{21}$$

Furthermore, because the link (v,u) is chosen to be added into the tree, it must hold that

$$d[u] = \delta(\pi_{su}) = \max(\delta(\pi_{sv}), w_{vu}) \ge w_{vu}. \tag{22}$$

Combining (21) and (22), we can obtain $w_{vu} \le d[u] \le w_{ab}$. Recalling that $(v,u) \in C_X$ and (a,b) is any cut link in C_X, the above inequality shows that $w_{vu} = \psi(C_X)$.

Finally, since (v,u) is a bottleneck link of T_s, we have $\delta(T_s) = w_{vu} = \psi(C_X)$. From the min-max-tree max-min-cut theorem, we can conclude that T_s is a min-max tree. ■

The above practice shows that our analysis on the min-max tree and max-min cut problems can actually provide the insights to explain some results that previously seem separate and different (*e.g.,* the work in [8]–[14] including the proposal in Section IV) under a unified theorem.

IV. Distributed Min-Max Tree Algorithm

The investigation of the min-max tree properties now would allow us to design a distributed min-max tree algorithm. It has an initial neighbor discovery process that allows each node v to be aware of the existence of all its one-hop neighbors N_v and the one-hop neighborhood knowledge $w_{vu}, u \in N_v$. In this section, we present a multicast routing protocol using the Search-Report and Grow-Request message forwarding mechanism to construct

Search-and-Grow procedure	
Search Phase:	find the minimum weight δ_{LB}^i of the links crossing node sets $N(T_s^{i-1})$ and $N - N(T_s^{i-1})$.
Grow Phase:	the tree T_s^{i-1} then grows to be T_s^i by absorbing as many links as possible such that any included link (v,u) satisfies $w_{vu} \leq \delta_{LB}^i$ and the resulting sub-graph still keeps a tree structure until no more such links can be found.

Fig. 3. The description of the ith round of Search-and-Grow procedure.

an optimal min-max tree. Such description is nonstrict but sufficient for us to study the communication complexity of the distributed algorithm in Section V.

Whenever there is a multicast session request (s, D) in the network but no route information is known, the source will initiate the distributed min-max tree algorithm by running the *Search-and-Grow* procedure iteratively. Let T_s^i be the intermediate tree constructed after the ith $(i \geq 1)$ round of Search-and-Grow procedure. The algorithm starts from an initial tree T_s^0, which contains the source node s only, and each iteration-i makes the tree grow from T_s^{i-1} to T_s^i with one or more nodes included. Eventually, such iterations terminate until the tree contains all the nodes in M and the final min-max tree is achieved by pruning all unnecessary links (the branches including nonmember nodes only). The pseudocode of the ith round of Search-and-Grow procedure is given in Fig. 3.

We use an example to illustrate the basic tree construction steps in the above algorithm. A 10-node graph is given in a 10×10 square, in which all nodes are multicast members and node 0 is the source. The link weight is defined as its Euclidean length.

The min-max tree is constructed step by step with three iterations as shown in Fig. 4(a)–(c), respectively.

Step 0: Initially, the tree consists of only the source node 0.

Step 1: In the first iteration, link (0, 4) crossing node sets $\{0\}$ and $\{1,2,3,4,5,6,7,8,9\}$ is found with minimum weight, and it is the only link added into the tree as shown the dark arc in Fig. 4(a).

Step 2: In the second iteration, link (0, 7) crossing node sets $\{0,4\}$ and $\{1,2,3,5,6,7,8,9\}$ is found with minimum weight and added into the tree. Link (7, 9) is also included because $w_{79} < w_{07}$ as shown in Fig. 4(b). We denote such subsequently included links as the light arcs.

Step 3: In the third iteration, link (9, 1) crossing node sets $\{0,4,7,9\}$ and $\{1,2,3,5,6,8\}$ is found with minimum weight and added into the tree. The tree then grows by including more links (1, 3), (1, 5), (1, 6), (3, 8), and (6, 2) since their weights are all less than w_{91}. The min-max tree is eventually obtained as shown in Fig. 4(c) with the bottleneck link (9, 1) that is found in the last iteration.

In the following, we describe how the Search-Report and Grow-Request messages are propagated in the ith round of Search-and-Grow procedure. In the Search phase, each tree node $v \in N\left(T_s^{i-1}\right)$ first calculates an upper bound of δ_{LB}^i locally as follows:

$$\begin{aligned}\psi_{UB}(v) &= \min\left\{w_{vu} | u \in N_v - N\left(T_s^{i-1}\right)\right\} \\ &\geq \min\left\{w_{xy} | x \in N\left(T_s^{i-1}\right), y \in N - N\left(T_s^{i-1}\right)\right\} \\ &\equiv \delta_{LB}^i. \end{aligned} \tag{23}$$

It then sends back a Search-Report message to its parent node (if $v \neq s$) with the parameter $\psi_{UB}(v)$ if v is a leaf node or, otherwise, the parameter $\min\left\{\psi_{UB}(v), \psi_{UB}(u) | u : (v,u) \in A\left(T_s^{i-1}\right)\right\}$ after collecting all the Search-Report messages from its child nodes. Furthermore, if v is a multicast member, it also attaches its own address in the Search-Report message to notify its attendance to the multicast. These messages propagating back to the source, shown as the dotted arrowed lines in Fig. 5, shall eventually allow the source to calculate a lower bound δ_{LB}^i of $\delta_{\min}$ as

$$\begin{aligned}\delta_{LB}^i &= \min\left\{\psi_{UB}(v) | v \in N\left(T_s^{i-1}\right)\right\} \\ &= \psi\left(C_{N(T_s^{i-1})}\right) \\ &\leq \delta_{\min}. \end{aligned} \tag{24}$$

If not all multicast members are included in the tree, the source initiates a Grow phase by flooding the Grow-Request messages with the parameter δ_{LB}^i over the tree T_s^{i-1}. Otherwise, the iterations of Search-and-Grow procedure terminate.

After the tree-flooding of the Grow-Request messages, shown as the solid arrowed lines within the area T_s^{i-1} in Fig. 5, each tree node v would further forward Grow-Request to a nontree node u if $w_{vu} \leq \delta_{LB}^i$. At each new included node, the node from which the first Grow-Request message is received will be set as the parent node, and all subsequent duplicate Grow-Request messages are simply dropped (*e.g.,* the message from x to y as shown in Fig. 5). Such message propagation, shown as the solid arrowed lines in the shaded area outside T_s^{i-1} in Fig. 5, will proceed until the incremental tree T_s^i ceases growing. When the Grow operation completes at node v, it then goes to the Search operation again as described earlier.

Eventually, a multicast forwarding tree is created by several Search-and-Grow cycles until all members join the tree. After that, the final min-max tree can be obtained straightforward by pruning all unnecessary links shown as the dotted lines in Fig. 5. Such post-procedure starts backwards from the nonmember leaf nodes and performs in a distributed manner.

Note that at each round of Search-and-Grow procedure, the tree grows by one more node at least. Therefore, we can assume that there are exactly k $(1 \leq k \leq n-1)$ rounds of Search-and-Grow procedure to achieve the final multicast tree. Finally, it remains to show that the multicast tree discovered by our distributed algorithm is a min-max tree. This is stipulated in Theorem 5 as follows.

Theorem 5: The distributed algorithm constructs a min-max tree.

Proof: Let T_s be the final tree obtained from the distributed algorithm. We observe that the sequence of the values δ_{LB}^i in the

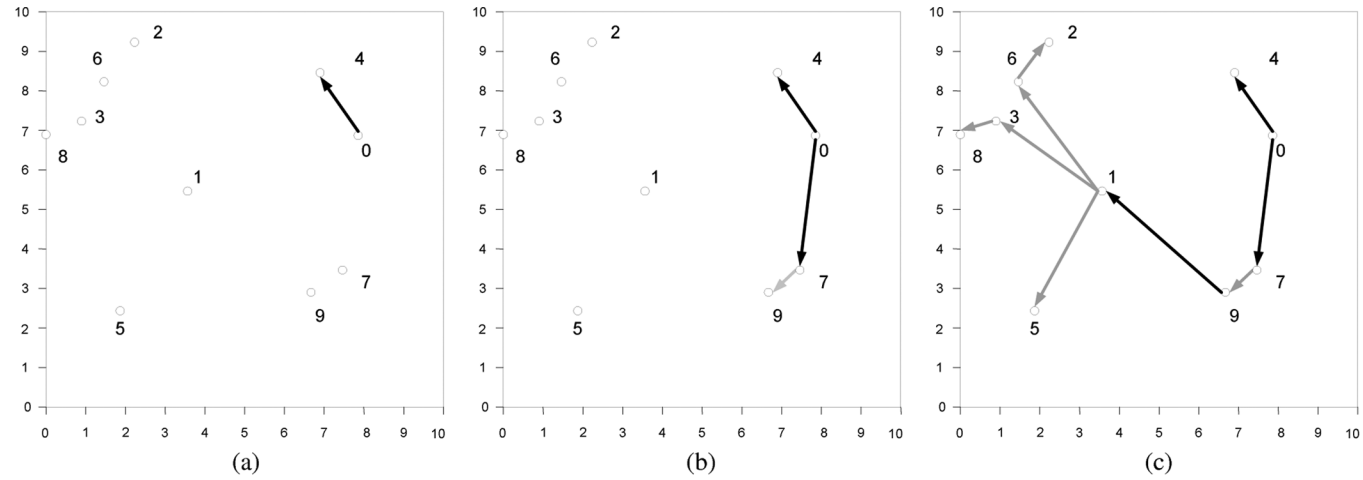

Fig. 4. Examples of min-max tree construction using the iterations of the Search-and-Grow procedure. (a) iteration-1. (b) iteration-2. (c) iteration-3.

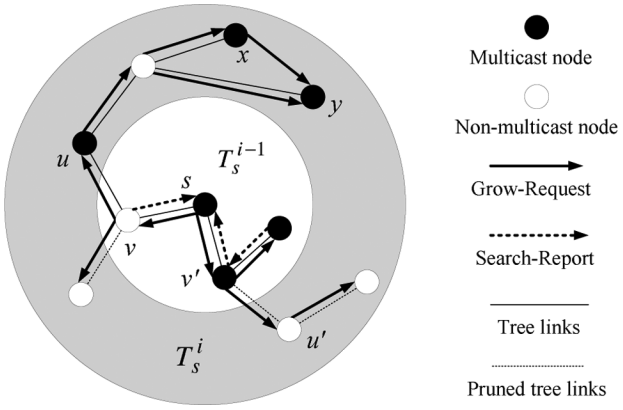

Fig. 5. Illustration of the message propagation at the ith round of Search-and-Grow procedure.

multicast tree formation is in an increasing order and the final one in the sequence is equal to $\delta(T_s)$, *i.e.,*

$$\delta^1_{LB} < \cdots < \delta^k_{LB} = \delta(T_s), \quad 1 \le k \le n-1. \tag{25}$$

Now, we can define $X \equiv N\left(T_s^{k-1}\right)$ and thus obtain $\delta(T_s) = \delta^k_{LB} = \psi(C_X)$ by substituting (24) into (25). From the min-max-tree max-min-cut theorem, we can conclude that the final tree T_s is a min-max tree. ■

V. Communication Complexity Analysis

In order to study the upper bound of message complexity of distributed algorithms for the min-max tree problem, we only need to consider the spanning tree case. The best results on the communication complexity is $O(n \log n + m)$ [15], [16] for the min-max spanning tree problem in an undirected graph. In this section, we shall analyze the communication complexity of our distributed algorithm for the same problem in a directed graph.

We consider the message interactions in the ith round of Search-and-Grow procedure. It can be considered to consist of two components: 1) the tree-flooding of the messages (Search-Report and Grow-Request) in the area T_s^{i-1}; and 2) the network-flooding of the messages (Grow-Request) in the area $T_s^i - T_s^{i-1}$ as shown in Fig. 5.

Let n_i and m_i be the number of messages propagated within the areas T_s^{i-1} and $T_s^i - T_s^{i-1}$, respectively. Thus, the communication complexity c (*i.e.,* the total number of Search-Report and Grow-Request messages) of the distributed algorithms using k rounds of Search-and-Grow procedure can be expressed as follows:

$$c = c_1 + c_2 \tag{26}$$

$$c_1 = \sum_{i=1}^{k} n_i = \sum_{i=1}^{k} 2\left|A\left(T_s^{i-1}\right)\right| \tag{27}$$

$$c_2 = \sum_{i=1}^{k} m_i. \tag{28}$$

Recall that in the network-flooding, the Grow-Request messages are delivered only to nontree nodes, which will join the tree and never receive such messages again in the later rounds. Therefore, throughout the whole distributed algorithm, the network-flooding Grow-Request message passes on each link at most once, resulting in

$$c_2 = \sum_{i=1}^{k} m_i \le m. \tag{29}$$

In the best case, only one round of Search-and-Grow procedure can span all the nodes, and thus the total number of messages is at most

$$c_{\text{best}} \le 2\left|A\left(T_s^0\right)\right| + m = m. \tag{30}$$

On the other hand, in the worst case, the final min-max spanning tree T_s^k needs exactly $k = n-1$ rounds, and each round only includes one more node, *i.e.,* $\left|A\left(T_s^{i-1}\right)\right| = i-1, 1 \le i \le$

k, which can be substituted into (26)–(29) to obtain the total number of messages as follows:

$$\begin{aligned} c_{\text{worst}} &= \sum_{i=1}^{k} 2\left|A\left(T_s^{i-1}\right)\right| + \sum_{i=1}^{k} m_i \\ &= \sum_{i=1}^{n-1} 2(i-1) + \sum_{i=1}^{k} m_i \\ &\le (n-1)(n-2) + m. \end{aligned} \tag{31}$$

We now turn our attention to the more interesting and difficult task on analyzing the complexity for the average case. Suppose that our distributed algorithm includes the links into the min-max spanning tree in the order of $(e_1, \ldots, e_{n-1})$, in which e_i is the ith included link with the weight w_i ranked as the r_ith $(1 \le i \le n-1)$ one in the corresponding weight list sorted in an increasing order. We consider there are exactly $i-1$ number of tree links already in the tree. A key observation made below allows us to derive the average message complexity in the rest of the analysis.

Observation 2: Link e_i is the first one chosen to be included in a certain Search-and-Grow procedure, if and only if $r_i > r_j$, $1 \le j \le i-1$.

If we define a random variable x_i $(1 \le i \le n-1)$ as follows:

$$x_i = \begin{cases} 1, & r_j < r_i, 1 \le j \le i-1 \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

then the communication complexity c_1 in (27) can be rewritten as

$$c_1 = \sum_{i=1}^{n-1} 2(i-1)x_i. \tag{33}$$

We first consider the case in which each value w_i is unique over the weight list $(w_1, \ldots, w_{n-1})$. The probability of the event $x_i = 1$, *i.e.,* $\Pr(x_i = 1)$, can be obtained under this assumption as shown in the following lemma.

Lemma 2:

$$\Pr(x_i = 1) = \frac{1}{i}. \tag{34}$$

Proof: See the Appendix for details. ∎

This result allows us to calculate the average number of Search-and-Grow iterations and eventually to achieve the average total number of messages, which are stipulated in the following Theorems 6 and 7, respectively.

Theorem 6: The average number of Search-and-Grow iterations is bounded by $\gamma + \ln n$, in which γ is the Euler constant $(\gamma = 0.5772\ldots)$, *i.e.,*

$$E\left[\sum_{i=1}^{n-1} x_i\right] \le \gamma + \ln n. \tag{35}$$

Proof: This result can be obtained immediately from Lemma 2 as follows:

$$\begin{aligned} E\left[\sum_{i=1}^{n-1} x_i\right] &= \sum_{i=1}^{n-1} E[x_i] = \sum_{i=1}^{n-1} \Pr(x_i = 1) \\ &= \sum_{i=1}^{n-1} \frac{1}{i} = H_{n-1} \\ &\le \gamma + \ln n. \end{aligned}$$

∎

Notice that we use the well-known inequality $H_{n-1} < \gamma + \ln n < H_n$ to achieve the final result, in which $H_n = 1 + (1/2) + \cdots + (1/n)$ is the nth harmonic number.

Theorem 7: The average total number of messages is bounded by $2(n-1-\gamma) - 2\ln(n-1) + m$, *i.e.,*

$$E[c] \le 2(n-1-\gamma) - 2\ln(n-1) + m. \tag{36}$$

Proof: Considering Equations (26), (33), (29) and (34), we have the following derivations:

$$\begin{aligned} E[c] &\le \sum_{i=1}^{n-1} E[2(i-1)\cdot x_i] + m \\ &= \sum_{i=1}^{n-1} 2(i-1)\cdot \Pr(x_i = 1) + m \\ &= \sum_{i=1}^{n-1} 2(i-1)\cdot \frac{1}{i} + m \\ &= 2(n-1) - 2H_{n-1} + m \\ &\le 2(n-1-\gamma) - 2\ln(n-1) + m. \end{aligned}$$

∎

Now, we consider the weight list with redundant elements. Let $W = (w_1, \ldots, w_i, \ldots, w_j, \ldots, w_{n-1})$ and $W' = (w'_1 = w_1, \ldots, w'_i = w_i, \ldots, w'_j = w_i, \ldots, w'_{n-1} = w_{n-1}$ be two weight lists, in which the ith and jth $(i < j)$ elements in W' have the same value w_i. By comparing W and W', we can conclude that

$$\begin{cases} x'_i = x_i, & i \ne j \\ x'_i = 0, & i = j \end{cases} \tag{37}$$

which achieves the relation of their communication complexities directly from (33) as follows:

$$c(W) \ge c(W') = \begin{cases} c(W), & x_j = 0 \\ c(W) - 2(j-1), & x_j = 1. \end{cases} \tag{38}$$

This result shows that the average communication complexity of our distributed algorithm to find the min-max spanning trees with redundant link weights should not exceed the upper bound $2(n-1-\gamma) - 2\ln(n-1) + m$ given in (36) as well.

To our best knowledge, this is the first theoretical analysis to show that a distributed algorithm can achieve an expected linear communication complexity $O(m)$ for the directed min-max tree problem. Since the undirected min-max tree problem can be

viewed as a special case, our algorithm can also apply to undirected graphs. The best distributed algorithms known so far to find an undirected min-max tree can be found in [15] and [16], which use the technique of parallelly merging minimum subtrees to achieve a message complexity at order of $O(n\log n + m)$. We thus can conclude that our distributed algorithm outperforms them in terms of average message complexity for such problems.

VI. Discussions

In this section, we would like to further discuss the results obtained so far with regard to several related problems in the literature and also to explore more general conclusions.

A. Minimum Steiner Tree versus Min-Max Steiner Tree

The minimum Steiner tree and min-max Steiner tree have been both received much attention in wireless networks for energy-aware multicast problems. When the objective is to minimize the total energy consumption of the multicast tree, the minimum Steiner tree algorithms are especially of interest. These algorithms are not optimal [22] since they tend to optimize link costs but not node costs, which can save energy further by exploiting the broadcast advantage property. However, the minimum Steiner tree is still a good approach because any Steiner tree algorithm with a constant-factor approximation ratio gives rise to a heuristic for the minimum energy multicast problem with a constant-factor approximation ratio as well [23]. On the contrary, the min-max Steiner tree is an optical approach when the objective is to maximize the lifetime of multicast tree. This is because such optimization problem can be modeled as a min-max tree problem [9]–[11] just based on the broadcast advantage property.

B. Symmetric Directed Graph versus Directed Graph

We observe that the results in Section III, including the min-max-tree max-min-cut Theorem and the centralized optimal algorithms, are also correct in general directed graphs because they are obtained without any dependency on the symmetric-arc property. On the other hand, the proposed distributed min-max tree algorithm exploits such property by allowing the Search-Report messages to propagate in a reverse direction of the tree arcs. Now, we consider extending the results in Sections IV and V to directed graph under the condition of strong connectivity, or a more relaxed condition in which for each arc (v, u) in G, there must exist a directed path from u to v in G as well. Note that the latter is the weakest requirement that must be satisfied in a communication network if the packet acknowledgement to senders should be supported.

We assume an underlying shortest-path unicast protocol for our distributed algorithm such that for any tree arc (v, u), node u can forward the Search-Report message to its parent node v through the route provided by the unicast protocol if the reverse arc (u, v) does not exist. Let h_{uv} be the hop number of directed shortest-path from u to v and ρ the radius of the network, *i.e.,* the maximum hop number of any shortest path in the network. In such network settings, the message component in (27) can be rewritten as

$$c_1 = \sum_{i=1}^{k} \left|A\left(T_s^{i-1}\right)\right| + \sum_{i=1}^{k} \sum_{(v,u)\in A\left(T_s^{i-1}\right)} h_{uv} \tag{39}$$

in which the first term is the number of Grow-Request messages propagating along the tree arcs and the second term is the number of Search-Report messages propagating backwards through unicast routes. Equation (39) can also be expressed in terms of the variable x_i, *i.e.,*

$$\begin{aligned} c_1 &\le \sum_{i=1}^{k} \left|A\left(T_s^{i-1}\right)\right| + \sum_{i=1}^{k} \sum_{(v,u)\in A\left(T_s^{i-1}\right)} \rho \\ &= (1+\rho)\sum_{i=1}^{k} \left|A\left(T_s^{i-1}\right)\right| \\ &= (1+\rho)\sum_{i=1}^{n-1} (i-1)x_i. \end{aligned} \tag{40}$$

Finally, the average message complexity is derived in a similar way as in Theorem 7.

$$\begin{aligned} E[c] &\le (1+\rho)\sum_{i=1}^{n-1} E[(i-1)\cdot x_i] + m \\ &\le (1+\rho)(n-1-\gamma) - (1+\rho)\ln(n-1) + m. \end{aligned} \tag{41}$$

C. Min-Max-Tree Max-Min-Cut Theorem versus Max-Flow Min-Cut Theorem

By reexamining the analysis in Section III, we find that the (42) implied by the min-max-tree max-min-cut theorem has a dual given in (43), which is referred as the *max-min-tree min-max-cut* theorem.

$$\min_{T_s} \max_{(v,u)\in A(T_s)} w_{vu} = \max_{C_X} \min_{(v,u)\in C_X} w_{vu} \tag{42}$$

$$\max_{T_s} \min_{(v,u)\in A(T_s)} w_{vu} = \min_{C_X} \max_{(v,u)\in C_X} w_{vu}. \tag{43}$$

The above *max-min-tree min-max-cut* theorem has a very close relationship to the traditional *max-flow min-cut* theorem [24] to be explained as follows.

Let weight w_{vu} be the link capacity of a directed graph G. In the case of single destination, *i.e.,* $D = \{d\}$, the maximum flow $f^{\max}_{\text{multi-path}}$ from source s to the destination d across multiple paths in G can be achieved in (44) directly from the *max-flow min-cut* theorem.

$$f^{\max}_{\text{multi-path}} = \min_{C_X} \sum_{(v,u)\in C_X} w_{vu}, \quad s \in X,\ D \not\subset X. \tag{44}$$

If we constrain the flow from source s to the destination d only along a single path, the value of a flow is exactly equal to the minimum link weight of this path $\min_{(v,u)\in A(T_s)} w_{vu}$, in which the multicast tree T_s degenerates into a directed path

from s to d. This shall result in the corresponding maximum flow $f^{\max}_{\text{single}-\text{path}}$ to be achieved in (45) from our *max-min-tree min-max-cut* theorem expressed in (43).

$$f^{\max}_{\text{single}-\text{path}} = \min_{C_X} \max_{(v,u)\in C_X} w_{vu}, \quad s \in X,\ D \not\subset X. \tag{45}$$

Equations (44) and (45) reveal the relationship of these two theorems in single-destination case. We now consider the multidestination case. The corresponding *max-flow* problem can be modeled as a *max-multicast-flow* problem in a linear programming form given in

$$\max: \quad f \tag{46}$$

$$s.t. \quad \sum_{u:(u,s)\in A} f^d_{us} - \sum_{u:(s,u)\in A} f^d_{su} = -f \quad \forall\, d \in D \tag{47}$$

$$\sum_{u:(u,v)\in A} f^d_{uv} - \sum_{u:(v,u)\in A} f^d_{vu} = 0 \quad \forall\, v \in N - \{s,d\},\ \forall\, d \in D \tag{48}$$

$$\sum_{u:(u,d)\in A} f^d_{ud} - \sum_{u:(d,u)\in A} f^d_{du} = f \quad \forall\, d \in D \tag{49}$$

$$f^d_{vu} \le w_{vu} \quad \forall\, (v,u) \in A,\ \forall\, d \in D. \tag{50}$$

Note that the capacity constraints in (50) are upon each individual flow f^d_{vu} from source s to a destination node d, not the accumulated flow

$$\sum_{d\in D} f^d_{vu} \le w_{vu} \quad \forall\, (v,u) \in A \tag{51}$$

as required in the traditional multicommodity flow problem. We observe that (45) is still true for the multidestination case, in which each individual flow f^d_{vu} is constrained on a single path from the source to the destination d. This is because the value of a multicast-flow f that satisfies constraints (47)–(50) is exactly equal to the minimum link weight of the tree, which is obtained by superposing all directed paths from source to each destination node. Equation (44) shall also hold, as conjecture, for the multidestination case, and we leave the strict proof in our future work.

Finally, a corollary from this conjecture as the multicast version of the Menger's theorem [24] is provided below. It can be easily obtained by applying (44) to a binary capacity network.

Corollary 5: Let G be a finite undirected graph and (s, D) the multicast with source s and nonempty destination node set D, $s \notin D$. The minimum number of edges whose removal disconnects s and at least one destination node in D is equal to the maximum number of edge-disjoint paths from s to *each* destination node in D.

VII. Conclusion

In this paper, we have systematically explored the properties of min-max tree and described how to use them for the design of a distributed algorithm. The scalability of the distributed algorithm in terms of average communication overhead has been also validated using theoretical analysis.

Recall that the communication complexity of the distributed algorithm in the worst case is $O(n^2)$, which is worthy of further exploration. One would be interested in improving this complexity to the order $O(n \log n + m)$, which is similar to the results in [15] and [16], or even to the same order as the average case $O(m)$. Due to the importance of the problem in multihop wireless networks, it is worth extending the results of this paper to the networks using directional antennas [19]–[21] as a possible new research direction.

Appendix
Proof of Lemma 2

In the case that each value w_i is unique over the weight list $(w_1, \ldots, w_{n-1})$, the statement $x_i = 1$ is true only if $r_i \in \{i, i+1, \ldots, n-1\}$. Thus, we can express the probability of $x_i = 1$ as follows:

$$\begin{aligned}\Pr(x_i = 1) &= \Pr(r_1 < r_i, \ldots, r_{i-1} < r_i) \\ &= \sum_{j=1}^{n-1} \Pr(r_1 < j, \ldots, r_{i-1} < j | r_i = j) \\ &\quad \cdot \Pr(r_i = j).\end{aligned} \tag{52}$$

We also observe that for a given link list $(e_1, \ldots, e_{n-1})$, the corresponding weight rank order $(r_1, \ldots, r_{n-1})$ is a permutation of $(1, \ldots, n-1)$. Under the condition $r_i = j$ $(i \le j \le n-1)$, the first $i-1$ elements in the list $(r_1, \ldots, r_{n-1})$ should be selected only from the set $\{1, \ldots, j-1\}$ with $j-1$ elements, instead of the set $\{1, \ldots, n-1\} - \{j\}$ with $n-2$ elements, to satisfy the statement $(r_1 < j, \ldots, r_{i-1} < j)$, resulting in the conditional probability $\Pr(r_1 < j, \ldots, r_{i-1} < j | r_i = j)$ to be

$$\Pr(r_1 < j, \ldots, r_{i-1} < j | r_i = j) = \frac{\binom{j-1}{i-1}}{\binom{n-2}{i-1}}. \tag{53}$$

We now can find the probability of $x_i = 1$ by substituting (53) and $\Pr(r_i = j) = 1/(n-1)$ into (52) as follows:

$$\begin{aligned}\Pr(x_i = 1) &= \sum_{j=i}^{n-1} \frac{\binom{j-1}{i-1}}{\binom{n-2}{i-1}} \cdot \frac{1}{n-1} \\ &= \frac{1}{(n-1)\cdot\binom{n-2}{i-1}} \\ &\quad \cdot \left(\binom{i-1}{i-1} + \binom{i}{i-1} + \cdots + \binom{n-2}{i-1} \right) \\ &= \frac{1}{(n-1)\cdot\binom{n-2}{i-1}} \\ &\quad \cdot \left(\binom{i}{i} + \binom{i}{i-1} + \cdots + \binom{n-2}{i-1} \right) \\ &= \frac{1}{(n-1)\cdot\binom{n-2}{i-1}} \cdot \binom{n-1}{i} = \frac{1}{i}.\end{aligned}$$

Notice that we use the well-known combinatorial identity $\binom{n}{j} = \binom{n-1}{j} + \binom{n-1}{j-1}$, $j < n$, to simplify the summation in the above derivation. ■

References

[1] R. Guerin and A. Orda, "Computing shortest paths for any number of hops," *IEEE/ACM Trans. Netw.*, vol. 10, no. 5, pp. 613–620, Oct. 2002.

[2] M. X. Cheng *et al.*, "Energy-efficient broadcast and multicast routing in ad hoc wireless networks," in *Proc. IEEE IPCCC*, Phoenix, AZ, Apr. 2003, pp. 87–94.
[3] I. Papadimitriou and L. Georgiadis, "Energy-aware broadcast trees in wireless networks," *Mobile Netw. Appl.*, vol. 9, no. 6, pp. 567–581, Dec. 2004.
[4] J. Edmonds and R. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, no. 2, pp. 248–264, Apr. 1972.
[5] P. M. Camerini, "The min-max spanning tree problem and some extensions," *Inf. Process. Lett.*, vol. 7, no. 1, pp. 10–14, Jan. 1978.
[6] C. W. Duin and A. Volgenant, "The partial sum criterion for Steiner trees in graphs and shortest paths," *Eur. J. Oper. Res.*, vol. 97, no. 1, pp. 172–182, Feb. 1997.
[7] H. N. Gabow and R. E. Tarjan, "Algorithms for two optimization problems," *J. Algor.*, vol. 9, no. 3, pp. 411–417, Sep. 1988.
[8] L. Georgiadis, "Bottleneck multicast trees in linear time," *IEEE Commun. Lett.*, vol. 7, no. 11, pp. 564–566, Nov. 2003.
[9] I. Kang and R. Poovendran, "Maximizing static network lifetime of wireless broadcast adhoc networks," in *Proc. IEEE ICC*, May 2003, pp. 2256–2261.
[10] A. K. Das, R. J. Marks, II, M. A. El-Sharkawi, P. Arabshahi, and A. Gray, "MDLT: A polynomial time optimal algorithm for maximization of time-to-first-failure in energy-constrained broadcast wireless networks," in *Proc. IEEE Globecom*, San Francisco, CA, Dec. 2003, pp. 362–366.
[11] S. Guo and O. Yang, "A framework for the multicast lifetime maximization problem in energy-constrained wireless ad-hoc networks," *ACM Wireless Netw. J.*, vol. 15, no. 3, pp. 313–332, Apr. 2009.
[12] B. Floréen *et al.*, "Multicast time maximization in energy constrained wireless networks," in *Proc. ACM Joint Workshop Found. Mobile Comput.*, San Diego, CA, Sep. 2003, pp. 50–58.
[13] S. Guo and O. Yang, "Multicast lifetime maximization for energy-constrained wireless ad-hoc networks with directional antennas," in *Proc. IEEE Globecom*, Dallas, TX, Dec. 2004, pp. 4120–4124.
[14] S. Guo, V. Leung, and O. Yang, "Improving scalability for longest-lived multicast using localized operations in WANETs," in *Proc. IEEE SECON*, San Diego, CA, Jun. 2007, pp. 243–252.
[15] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum weight spanning trees," *ACM Trans. Program. Lang. Syst.*, vol. 5, no. 1, pp. 66–77, 1983.
[16] M. Faloutsos and M. Molle, "A linear-time optimal-message distributed algorithm for minimum spanning trees," *Distrib. Comput.*, vol. 17, pp. 151–170, 2004.
[17] R. C. Prim, "Shortest connection networks and some generalisations," *Bell Syst. Tech. J.*, vol. 36, pp. 1389–1401, 1957.
[18] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
[19] S. Guo and O. Yang, "Formulation of optimal tree construction for maximum lifetime multicasting in wireless ad hoc networks with adaptive antennas," in *Proc. IEEE ICC*, Seoul, Korea, May 2005, pp. 3370–3374.
[20] Y. T. Hou, Y. Shi, H. D. Sherali, and J. E. Wieselthier, "Online lifetime-centric multicast routing for ad hoc networks with directional antennas," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 761–772.
[21] S. Guo, O. Yang, and V. Leung, "Approximation algorithms for longest-lived directional multicast communications in wanets," in *Proc. ACM MobiHoc*, Montreal, QC, Canada, Sep. 2007, pp. 190–198.
[22] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating optimal multicast trees in wireless multihop networks," in *Proc. IEEE ISCC*, Cartagena, Spain, Jun. 2005, pp. 686–691.
[23] P.-J. Wan, G. Calinescu, X.-Y. Li, and O. Frieder, "Minimum-energy multicast routing in static ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 507–514, Jun. 2004.
[24] B. Bollobas*, Graph Theory, an Introduction Course*. New York: Springer-Verlag, 1979.

Song Guo (M'05) received the Ph.D. degree in computer science from the University of Ottawa, Ottawa, Canada, in 2005.

Since then, he held a position with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, on a Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship. He is currently an Assistant Professor with the School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, Japan. His research interests are mainly in the areas of protocol design and performance analysis for computer and telecommunication networks, presently focusing on modeling, analysis, cross-layer optimization, and performance evaluation of wireless ad hoc and sensor networks for reliable, energy-efficient, and cost-effective communications.

Victor C. M. Leung (F'03) received the B.A.Sc. (Hons.) and Ph.D. degrees in electrical engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 1977 and 1981, respectively.

From 1981 to 1987, Dr. Leung was a Senior Member of Technical Staff and Satellite Systems Specialist at MPR Teltech Ltd., Burnaby, BC, Canada. In 1988, he was a Lecturer in Electronics at the Chinese University of Hong Kong, Shatin, Hong Kong. He returned to UBC as a faculty member in 1989, where he is now a Professor and the TELUS Mobility Research Chair in Advanced Telecommunications Engineering in the Department of Electrical and Computer Engineering. His research interests are in mobile systems and wireless networks.

Dr. Leung is a Voting Member of the Association for Computing Machinery (ACM). He was the recipient of many academic awards, including the APEBC Gold Medal as the head of the 1977 graduating class in the Faculty of Applied Science, UBC, and the NSERC Postgraduate Scholarship. He is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and an Editor of the *International Journal of Sensor Networks*.