Routing, Anycast, and Multicast for Mesh and Sensor Networks

Roland Flury
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
Email: rflury@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
Email: wattenhofer@tik.ee.ethz.ch

***Abstract*— This paper studies routing schemes and their distributed construction in limited wireless networks, such as sensor or mesh networks. We argue that the connectivity of such networks is well captured by a constant doubling metric and present a constant stretch multicast algorithm through which any network node u can send messages to an arbitrary receiver set U. In other words, we describe a distributed approximation algorithm which is only a constant factor off the NP-hard Minimum Steiner Tree on $u \cup U$. As a building block for the multicasting, we construct a $1+\varepsilon$ stretch labeled routing scheme with label size $O(\log \Theta)$ and storage overhead $O(1/\varepsilon)^{\alpha}(\log \Theta)(O(\alpha)+\log \Delta)$, where Θ is the diameter of the network, Δ the maximum degree of any network node, and α a constant representing the doubling dimension of the network. In addition to unicast and multicast, we present a constant approximation for anycasting on the basis of $\sqrt{6}$-approximate distance queries. We provide a distributed algorithm to construct the required labeling and routing tables.**

I. Introduction

Sensor networks and wireless mesh networks in general have received a lot of attention lately, last but not least because of their countless applications. Examples include wild life observation, fire detection, battlefield support, environment monitoring and diverse distributed applications that require communication. Most use cases have in common that the quality of the network improves when more network nodes are deployed. On the one hand, this calls for less expensive hardware, which mostly comes along with limited capabilities, such as small storage, slow processors, and limited energy resources. On the other hand, the drift towards larger networks also requires appropriate programming of the devices using scalable algorithms that cope with large networks.

In this paper, we consider one of the most fundamental building blocks of such limited networks: the exchange of information between network nodes. In most cases, the only means to divulge information is by routing, either as unicast, anycast, multicast, or broadcast. Whereas there are standard approaches for the latter, it is much more challenging to implement *efficient* single destination routing, anycasting, and multicasting. The efficiency is not only concerned with how fast a message is delivered, but also how much overhead the algorithm generates. For example, optimal routing is straightforward if each node stores the optimal path to every other node in the network. But this comes at a cost of $O(n)$ routing entries, which easily exceeds the memory capacity of hardware limited network devices when the network grows.

We want to point out that there exist routing protocols that require no routing tables at all. For instance, geographic routing [5], [2], [10] routes messages solely based on the position information of the nodes: every node knows the position of itself and its neighbors, and greedily forwards a message to its neighbor closest to the destination. A first drawback of georouting is that the length of its paths may be up to the square of the optimal path length. Also, the algorithms presume precise position information, which is very hard to obtain [12]. Furthermore, these algorithms have been studied on the oversimplified unit disk graph (UDG) connectivity model where all transmission ranges are assumed to be perfectly circular and of equal radius.

The labeled routing scheme presented in this paper overcomes the above issues: With small routing tables of bounded size, we provide close to optimal unicasting, and constant approximations to anycast and multicast. The proposed algorithms are analyzed on a connectivity model that can be applied to virtually any wireless network. In addition, our assumption about the capabilities of the nodes is minimal: we only require that each node has a unique ID and that it can communicate with its direct neighbors.

The multicasting feature may be extremely helpful to execute a query only on a subset of nodes. E.g. suppose a fire detection system where a coordinator first determines the set of nodes measuring a critical state. In subsequent rounds, it may only need to survey these nodes. Talking to each of them individually introduces a much higher message overhead than efficiently multicasting a single message, cf. Figure 1. The gain of multicasting becomes even more substantial when the receiver set tends to be clustered. In contrast to multicasting, anycasting becomes interesting when a node needs to send some data to any node of a given set. In the example of the fire detection network, this may arise when a node measuring a critical state wants to inform one of several coordinators. The anycasting algorithm picks the coordinator as to minimize transmission cost.

We have already noted that the often studied UDG connectivity model is not appropriate for real wireless networks because it does not consider perturbations of the wireless medium, e.g. caused by obstacles. However, to ensure applicability of the proposed algorithms, they should be analyzed on a connectivity model that characterizes real wireless networks as

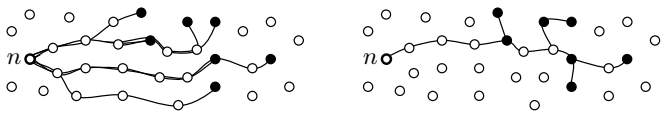

Fig. 1. Support for efficient multicasting is crucial. If node n sends a separate message to all destination nodes marked with solid circles (left image), it causes a much higher message overhead than if it utilizes an efficient multicast algorithm (right image).

accurately as possible. Clearly, any network connectivity can be described using a general graph. But taking into account the limited range of wireless devices, we observe that nodes are mostly connected with other nodes in their proximity, which results in a far more regular connectivity graph than a general graph. Although close-by nodes may be out of communication range due to obstacles, a node is typically highly connected to nodes in its surroundings. In other words, even in environments with many obstacles, the total number of mutually independent[1] neighbors of a node is likely to be small [14].

We exploit the slightly more general formulation of this property that takes into account the multi-hop nature of the network: Given a h-hop neighborhood of a node v, it is possible to cover all these nodes with only a small number of $\frac{h}{2}$-hop neighborhoods. Similar to before, this property states that in any h-hop neighborhood, one can pick only a relatively small subset of nodes with pairwise distance larger than $\frac{h}{2}$. Figure 2 shows the deployment of 39 sensor nodes in our office building and how the nodes are connected [4]. In this network, the 2-hop neighborhood of *any* node can be covered by at most five 1-hop neighborhoods. The figure shows the 2-hop neighborhood of node v, which is covered by four 1-hop neighborhoods.

These observations motivate to characterize the network connectivity by $\mathcal{A}$, the maximum number of half-sized neighborhoods that are necessary to cover any given neighborhood. This number is expected to be quite small for most wireless networks, e.g. a small two-digit number. Being the only parameter to describe the network, we can adjust $\mathcal{A}$ to apply our analysis to virtually any wireless network topology, including obstacle-rich deployments in 3D and worst case scenarios. Furthermore, because the presented routing algorithms are self-adaptive and need *not* to know the value of $\mathcal{A}$, they are good for any network. It is only the performance of the routing algorithms that depends on the value of $\mathcal{A}$.

In the sequel we describe the network connectivity by a weighted graph $G = (V,E)$ where each network node is mapped to a node of the graph, and a weighted edge is present between any two nodes within mutual transmission range. We observe that the coverability property described above matches the definition of a constant doubling metric[2] on G (see Definition 1.2). The distance metric $\mathcal{M}$ associated with G corresponds to the metrization of G using the cost-function

[1]Two nodes are independent if they are not (direct) neighbors.

[2]A metric assigns to each node-pair $(a,b) \in V \times V$ a cost satisfying non-negativity, identity of indiscernibles, symmetry and triangle inequality.

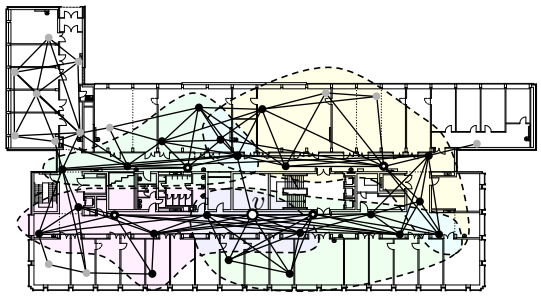

Fig. 2. The figure shows a deployment of 39 nodes in an office building and how they are connected. The nodes in black denote a 2-hop neighborhood of node v. These nodes can be covered by four 1-hop neighborhoods indicated with the dashed lines and rooted at the nodes with the small white dot.

$\mathrm{d}_{\mathcal{M}}(a,b)$ which assigns to each pair $(a,b) \in V \times V$ the cost of the least cost path between a and b. Our analysis holds for a more general class of networks, where each link may be assigned a cost, e.g. the number of retransmissions needed to send a message over the link. Setting all link costs to 1, we obtain the hop-metric discussed above.

Definition 1.1 (Ball): Given a node $v \in V$, the ball $\mathcal{B}_v(r)$ with radius r denotes the set of nodes with distance at most r from v: $\mathcal{B}_v(r) = \{u | \mathrm{d}_{\mathcal{M}}(v,u) \leq r\}$.

Definition 1.2 (Constant Doubling Metric): A graph $G = (V,E)$ fulfills the doubling metric property if any ball $\mathcal{B}_v(r)$ can be covered[3] by a constant number of balls $\mathcal{B}(\frac{r}{2})$ with half the radius: For $r \geq 0$ and $\forall v \in V$: $\exists U \subseteq \mathcal{B}_v(r)$ such that $|U| = \mathrm{O}(1)$ and $\mathcal{B}_v(r) \subseteq \bigcup_{u \in U} \mathcal{B}_u(\frac{r}{2})$. If $|U|$ is bounded by 2^{α} for a constant α, we say that the metric associated with G has *doubling dimension* α.

We point out that the value of $\alpha = \lceil \log_2 \mathcal{A} \rceil$ is quite small for most wireless networks, e.g. around 3 or 4.

II. Related Work

There is a huge body of work regarding routing on wireless networks consisting of hardware limited participants. In this paper, we are particularly interested in *scalable* routing algorithms that consider the limited hardware capabilities, but still achieve excellent routing performance. One milestone in this area was laid by Peleg and Upfal in [13], where they examine the trade-off between the efficiency of a routing algorithm and its space requirements. They present a stretch-k routing algorithm for general graphs with an average routing table size of $\mathrm{O}\left(k^3 n^{1/k} \log n\right)$ bits and $\mathrm{O}(\log n)$ bit labels. The renaming (labeling) of the network nodes is a widely used technique to reduce the routing table size. In fact, any routing algorithm that does not rename the nodes and requires a stretch below 3 may need routing tables of $\Omega(n)$ bits [6]. For constant doubling metrics, we know that the stretch is above $9-\varepsilon$ if the routing tables size is $\mathrm{o}\left(n^{(\varepsilon/60)^2}\right)$ [7].

In his recent work [16], Talwar described a $(1+\varepsilon)$ stretch routing scheme for α-doubling metrics with routing tables of

[3]A ball B is *covered* by a set of balls $\{b_1, \ldots, b_n\}$ if $\forall u \in B : \exists i$ such that $u \in b_i$.

$O(1/(\varepsilon\alpha))^{\alpha}\log^{2+\alpha}\Theta$ bits and label-size $O(\alpha\log\Theta)$ bits. This work was improved by Chan et al. in [3] by reducing the storage overhead to $(\alpha/\varepsilon)^{O(\alpha)}(\log\Theta)(\log\Delta)$ bits per node and a label size of $O\big(\alpha\log(\varepsilon^{-1})\big)\log\Theta$ bits.

Slivkins presented two improved compact routing schemes in [15]. The first uses $(\varepsilon)^{-O(\alpha)}(\log\Theta)(\log\Delta)$ bits per routing table and $O\big(\alpha\log(\varepsilon^{-1})\big)\log\Theta$ bits per label, whereas the second scheme uses $\varepsilon^{-O(\alpha)}(\log\Theta)(\log\log\Theta)(\log n)$ routing table bits and $2^{O(\alpha)}(\log n)\log(\varepsilon^{-1}\log\Theta)$ bits for each label.

In their seminal work [1], Abraham et al. presented the first compact routing scheme with $\lceil\log n\rceil$ bit labels and routing tables of $\varepsilon^{-O(\alpha)}\log n\log(\min(\Theta,n))$ bits. This work emphasized on scale-freedom, i.e. independence of Θ. Dropping this constraint, their technique easily yields $O(\Theta)$ bit labels. In addition, the authors provide a scale-free name-independent routing scheme, including the matching lower bounds.

Our work does not quite achieve the bounds of [1], our routing tables are of size $O\big(\frac{1}{\varepsilon}\big)^{\alpha}(\log\Theta)(O(\alpha)+\log\Delta)$ and the routing labels require $2\alpha\lceil\log\Theta\rceil+\lceil\log n\rceil$ bits[4]. This small overhead allows us to build an *all-in-one* routing scheme that not only supports unicasting (Section V), but also provides constant approximations to anycasting (Section VII) and multicasting (Section VI). Furthermore, we present a distributed algorithm to construct the labeling and routing tables in Section VIII.

III. Definitions and Preliminaries

We first define some further terms and state properties of doubling metrics that will turn out to be handy. We start with the definition of a $\rho-$net, which is closely related to maximal independent sets and dominating sets.

Definition 3.1 ($\rho-$net): A subset U of the node-set V of a graph $G=(V,E)$ is a $\rho-$net if each node in V has distance at most ρ to at least one node in U, and the mutual distance between any two nodes in U is strictly larger than ρ. Formally, a set $U\subseteq V$ is a $\rho-$net of $G=(V,E)$ if

$\forall v\in V:\exists u\in U:\mathrm{d}_{\mathcal{M}}(v,u)\leq\rho$ and

$\forall u_1,u_2\in U:\mathrm{d}_{\mathcal{M}}(u_1,u_2)>\rho$.

We say that a node $u\in\rho-$net covers the nodes contained in $\mathcal{B}_u(\rho)$. Thus, each node of the network is covered by at least one $u\in\rho-$net. In the sequel, we denote the nodes of a $\rho-$net as **net-centers** of the $\rho-$net.

Given a constant doubling metric, we know that for every ball $\mathcal{B}_v(r)$, it is possible to cover all nodes in $\mathcal{B}_v(r)$ with 2^{α} balls of half the radius. However, this is only a feasibility statement and does not give an upper bound on the number of $\mathcal{B}(\frac{r}{2})$ that may be deployed to cover $\mathcal{B}_v(r)$. The following property gives an upper bound on the number net-centers of $\frac{r}{2}-$net a ball $\mathcal{B}_v(r)$ may cover.

Property 3.1 (Sparseness): For $x\in\mathbb{N}^0$, each ball $\mathcal{B}_v(2^x\rho)$ covers at most $2^{(1+x)\alpha}$ nodes from an arbitrary $\rho-$net on the same graph with constant doubling dimension α.

[4]Note that even for a huge network of 10^6 nodes, diameter 10^4, and $\alpha=4$, the routing labels are still below the size of an IPv6 address and easily fit in the tiny messages of today's sensor nodes.

Proof: By recursively applying the definition of the doubling metric, we observe that it is possible to cover $\mathcal{B}_v(2^x\rho)$ with at most $2^{(1+x)\alpha}$ balls of radius $\frac{\rho}{2}$. Assume we know such a covering C with $|C|\leq 2^{(1+x)\alpha}$.

In order to see that $\mathcal{B}_v(2^x\rho)$ covers at most $2^{(1+x)\alpha}$ net-centers $U=\{u_1,u_2,\ldots,u_{|U|}\}$ from any $\rho-$net, we first observe that the distance between any two distinct nodes in U exceeds ρ. Secondly, by the definition of a cover, each $u_j\in U$ is covered by at least one ball from the cover C. Because the balls in C have radius $\frac{\rho}{2}$ and the distance between any two u_j is more than ρ, any $c\in C$ covers at most one $u_j\in U$. Thus, $|U|\leq|C|\leq 2^{(1+x)\alpha}$, which completes the proof. □

Along with this upper bound on the number of net-centers covered by a ball, we can give an upper bound on the number of net-centers of a given $\rho-$net that may cover any given node v. In particular, we are interested in the maximum number of net-centers of a $\rho-$net that cover a node v, if each net-center u covers an extended ball $\mathcal{B}_u(2^x\rho)$.

Property 3.2 (Dominance): Given a $\rho-$net on a α-doubling metric represented by the graph $G=(V,E)$ where each net-center u covers $\mathcal{B}_u(2^x\rho)$ with $x\in\mathbb{N}^0$, then any node $v\in V$ is covered by at most $2^{(x+1)\alpha}$ net-centers from the $\rho-$net.

Proof: If v is covered by a net-center u, then u is at most $2^x\rho$ away from v. Therefore, it is sufficient to show that $\mathcal{B}_v(2^x\rho)$ contains at most $2^{(x+1)\alpha}$ $\rho-$net-centers, which follows from Property 3.1. □

Corollary 3.1 (Dominance): Given a $\rho-$net on a α-doubling metric represented by the graph $G=(V,E)$ where each net-center u covers $\mathcal{B}_u(i\rho)$ with $i\in\mathbb{R}^+, i>1$, then any node $v\in V$ is covered by at most $2^{2\alpha}i^{\alpha}$ net-centers from the $\rho-$net.

IV. Dominance Net

Due to our minimal assumptions about the node's capabilities (unique ID and communication with direct neighbors), our routing algorithms need some means to characterize the network.

We propose to obtain this information through a *dominance net*, which is a hierarchic locality-preserving decomposition of the network. In a nutshell, the dominance net is built of several layers of $\rho-$nets with exponentially increasing radius ρ, each of which covers the entire network. In the sequel, we will show how to build a dominance net with the following properties:

1) The number of layers is at most $1+\lceil\log\Theta\rceil$, where Θ is the diameter of the network.[5]
2) Each node is dominated by at most $O(\log\Theta)$ nodes.
3) The parent-tree induced by the dominance net allows for an unique distance labeling with label-size $O(\log\Theta)$ and stretch at most $\sqrt{6}$.
4) Adding routing tables of $O(1/\varepsilon)^{\alpha}\log\Delta\log\Theta$ bits to the nodes allows for a routing scheme with stretch $1+\varepsilon$. (Δ is the maximum degree of any node.)

[5]Throughout the paper, log stands for the binary logarithm.

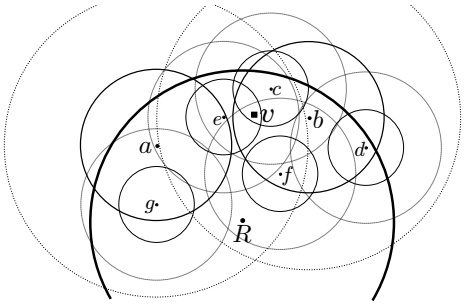

Fig. 3. A small dominance net with only three levels. R is the root and therefore the only net-center of level-2. It covers all nodes of the network. The net-centers of level-1 are $\Gamma_1 = \{a,b,R\}$ and $\Gamma_0 = \{c,d,e,f,g,a,b,R\}$ for level-0. (Note that some nodes are net-center on several levels.) The solid circle around each net-center indicates its coverage area, whereas the dashed circle indicates the extended coverage area. Note that the coverage area may have arbitrary shape – the circular coverage area is only used for this schematic representation.

For the rest of the paper, we assume w.l.o.g. that the smallest distance between any two nodes is 1, that the diameter of the network is given by Θ, and $\vartheta = 1 + \lceil \log \Theta \rceil$. Furthermore, we use $G = (V,E)$ to denote the graph induced by the network, whose metrization has constant doubling dimension α.

A. Dominance

For building the dominance net, we construct a hierarchy of $\rho-$nets on G with $\rho = 2^i$, where i is chosen from the range $\{0,1,2,3,...,\vartheta-1\}$. In the following, we call the the $(2^i)-$net the **level-*i*** of the hierarchy, and we denote its net-centers by Γ_i. Note that on level-$(\vartheta-1)$, a single (arbitrary) node of the network becomes net-center. We call this node the **root** of the hierarchy. An algorithm to construct these $\rho-$nets in a distributed manner is presented in Section VIII.

We define the dominance on this $\rho-$net hierarchy in the following way: A net-center $\gamma_i \in \Gamma_i$ is dominated by a net-center $\gamma \in \Gamma_{i+1}$ iff $\gamma_i \in \mathcal{B}_\gamma(2^{i+1})$. Thus, each net-center (except the root) is dominated by at least one net-center of the next higher level.

B. Naming Scheme

Given the dominance-net, we name the net-centers in the following way: Each net-center v except for the root node selects exactly one[6] of its dominators to be its *parent* $\mathcal{P}(v)$. This results in a dominance-tree with depth $\vartheta-1$. Each parent enumerates its children and informs each of them about the assigned enumeration value. The naming scheme is defined recursively: The root has an empty name, while any other net-center obtains its name by appending its enumeration value to its parents name.

Figure 3 shows a sample dominance net with only three levels. The left picture of Figure 4 shows the dominance-tree with the assigned enumeration values. The name of a net-center is obtained by walking from the corresponding node of the dominance-tree towards the root and concatenating the

[6]For performance reasons, each net-center may choose the closest net-center of the next higher level.

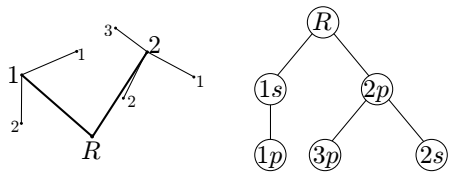

Fig. 4. The left picture shows the dominance-tree corresponding to the dominance-net of Figure 3. The tree on the right is the cover-tree of node v, a tree-representation of v's label. At depth i, this tree contains the enumeration-values of the net-centers of level-$(\vartheta-i-1)$ which cover v. The letter p indicates a primary net-center, s a secondary net-center.

enumeration values of the visited nodes. E.g. the name of net-center g is `R:1:2` whereas the name of c is `R:2:3`. The uniqueness of the names is guaranteed due to the tree structure, where each net-center is identified through its parent and the enumeration value assigned by the parent.

Because any net-center may be parent of at most $2^{2\alpha}$ net-centers (Property 3.1), the enumeration values can be represented with at most 2α bits. Thus, the name may grow by 2α bits on each level, which results in a maximum name size of $2\alpha\lceil \log \Theta \rceil = O(\log \Theta)$ bits for net-centers on level-0.

C. Dominance Labeling

Based on the dominance-net and the corresponding naming scheme, each node v of the graph G assigns itself a label $\mathcal{L}(v)$ of size $O(\log \Theta)$ which allows for efficient routing, multicasting, anycasting, and distance labeling. In the remainder of this section, we show the following theorem:

Theorem 4.1 (Compact Labels): The nodes of G can be assigned unique labels of size $O(\log \Theta)$ allowing approximate distance queries of stretch $\sqrt{6}$ between any two nodes.

We use an *extended dominance net* for the labeling, where a net-center $\gamma \in \Gamma_i$ covers all nodes in $\mathcal{B}_\gamma(2^{i+1})$, cf. Figure 3. This extension results in an overlap of the net-covers, which allows to rule out borderline effects where two close-by neighbors are not covered by common net-centers. Given this extended dominance set, each node v determines the net-centers by which it is covered. Essentially, the label of v contains the name of all net-centers that cover v.

In order to obtain the desired label size for a node v, we cannot simply store all names of the covering net-centers: By construction, there are at most ϑ levels, each of which has at most $2^{2\alpha}$ net-centers covering v (Property 3.2). From the previous section, we know that the naming for the net-centers uses at most $2\alpha(\vartheta-i-1)$ bits on level-i, which bounds the maximum label length to $\sum_{i=0}^{\vartheta} 2^{2\alpha} 2\alpha(\vartheta-i-1) = O(\vartheta^2) = O(\log^2 \Theta)$. To receive the $O(\log \Theta)$ label size, we make use of the following lemma which shows that only partial net-center names need to be stored for each covering net-center.

Lemma 4.2 (Parental Cover): If a net-center $c \in \Gamma_i$ covers a node v in the extended dominance net, then its parent $\mathcal{P}(c) \in \Gamma_{i+1}$ covers v as well.

Proof: The net-center c on level-i only covers nodes in $\mathcal{B}_c(2^{i+1})$ in the extended dominance net. Therefore, $d_{\mathcal{M}}(v,c) \leq$

2^{i+1}. Because the parent $\mathcal{P}(c)$ of c is chosen from the ordinary[7] dominance net and $\mathcal{P}(c) \in \Gamma_{i+1}$, $d_{\mathcal{M}}(c, \mathcal{P}(c)) \leq 2^{i+1}$. Using the triangle inequality we receive that $d_{\mathcal{M}}(v, \mathcal{P}(c)) \leq d_{\mathcal{M}}(v,c) + d_{\mathcal{M}}(c, \mathcal{P}(c)) \leq 2^{i+2}$. The claim follows because $\mathcal{P}(c)$ covers $\mathcal{B}_{\mathcal{P}(c)}(2^{i+2})$ in the extended dominance net, which includes v. □

D. Cover Tree

The label $\mathcal{L}(v)$ of a node v is the flat representation of a cover tree that efficiently describes all net-centers that cover v. This tree is rooted at the root node. The body of the cover tree is defined recursively: At depth i, the tree contains the net-centers $\gamma \in \Gamma_{\vartheta-i-1}$ that cover v. Each γ in the cover tree is connected to its parent $\mathcal{P}(\gamma)$. This is possible because we know from Lemma 4.2 that if γ covers v, then also $\mathcal{P}(\gamma)$ covers v and therefore must be present in the cover tree.

Because the name of a net-center is defined as the recursive concatenation of the enumeration values of its parents in the dominance net (see Section IV-B) and the same parenthood persists in the cover tree, each node of the cover tree only needs to store the enumeration value of its corresponding net-center. The name of a net-center c in the cover tree can be obtained by prefixing the enumeration value of c with the name of $\mathcal{P}(c)$, which is obtained recursively. Therefore, each node of the cover tree carries at most 2α bits, see Figure 4 for an example. We also know from Property 3.2 that there are at most $2^{2\alpha}$ net-centers per level covering node v. Because the depth of the tree is at most $\vartheta - 1$, we deduce that the cover tree holds at most $\alpha 2^{2\alpha+1}\vartheta$ bits. The serialization of the cover tree to a flat data structure is straightforward and can be done introducing only two additional bits per net-center, which leads to a label size of $(2\alpha+2)2^{2\alpha}\vartheta = O(\log \Theta)$ bits.

In the remainder of this paper, we rely on a slightly enhanced node labeling where each node v not only stores the net-centers by which it is covered, but also indicates their type. We distinguish two types of net-centers depending on their distance to v: A net-center $c \in \Gamma_i$ is called **primary** if $v \in \mathcal{B}_c(2^i)$. Otherwise, the net-center is called **secondary**, i.e. a secondary net-center covers v only in the extended dominance net, whereas a primary net-center covers v already in the ordinary dominance net. This additional bit per net-center does not significantly increase the size of the cover tree, which becomes at most $(2\alpha+3)2^{2\alpha}\vartheta = O(\log \Theta)$ bits. This proves the first part of Theorem 4.1.

E. Distance Approximation

To obtain the distance between two nodes a and b given their labels $\mathcal{L}(a)$ and $\mathcal{L}(b)$, we determine the smallest level-i for which a and b have at least one common net-center. Let $\rho = 2^i$ be the ordinary coverage radius of level-i and $C \subseteq \Gamma_i$ the set of common net-centers on level-i.

We immediately get a lower bound on the distance between a and b by observing that if $d_{\mathcal{M}}(a,b) \leq \rho/2$, there exists a net-center $c \in \Gamma_{i-1}$ that covers both a and b. Therefore, $d_{\mathcal{M}}(a,b) > \rho/2$. For the remainder of this proof, we need to consider the following three cases only:

1) If there is a net-center $c \in C$ s.t. c is primary for both a and b, then $d_{\mathcal{M}}(a,b) \leq 2\rho$ and therefore $d_{\mathcal{M}}(a,b) \in (\rho/2, 2\rho]$. This property holds due to the triangle inequality: Because a and b are primary, $d_{\mathcal{M}}(a,c) \leq \rho$ and $d_{\mathcal{M}}(b,c) \leq \rho$, which implies that $d_{\mathcal{M}}(a,b) \leq d_{\mathcal{M}}(a,c) + d_{\mathcal{M}}(c,b) \leq 2\rho$.
2) If all net-centers $c \in C$ are secondary for a and b, $d_{\mathcal{M}}(a,b) \in (\rho, 4\rho]$. Again, the upper bound is given by the triangle inequality: $d_{\mathcal{M}}(a,c) \leq 2\rho$ and $d_{\mathcal{M}}(b,c) \leq 2\rho$, and therefore $d_{\mathcal{M}}(a,b) \leq 4\rho$. The lower bound stems from the fact that any primary net-center of a would cover b if $d_{\mathcal{M}}(a,b) \leq \rho$, and vice versa.
3) Finally, if there is at least one net-center $c \in C$ s.t. c is primary for either a xor b, then $d_{\mathcal{M}}(a,b) \in (\rho/2, 3\rho]$. W.l.o.g. assume that c is primary for node a. Then $d_{\mathcal{M}}(a,c) \leq \rho$ and $d_{\mathcal{M}}(c,b) \leq 2\rho$. Therefore, $d_{\mathcal{M}}(a,b) \leq d_{\mathcal{M}}(a,c) + d_{\mathcal{M}}(c,b) \leq 3\rho$.
Note: If for either a xor b there is no $c \in C$ s.t. c is primary for the node, then $d_{\mathcal{M}}(a,b) \in (\rho, 3\rho]$. The increase in the lower bound holds because no primary net-center of a xor b covers the other node, which is only possible if $d_{\mathcal{M}}(a,b) > \rho$.

Given the interval $(r_1, r_2]$ for the possible values of $d_{\mathcal{M}}(a,b)$, we set $d_{\mathcal{M}}(a,b) = \sqrt{r_1 r_2}$, the geometric mean of the two bounds. The maximum factor by which the approximation is off from the actual distance is $\sqrt{r_2/r_1}$. Therefore, our labeling scheme suffers from a maximum stretch in the third case, which is at most $\sqrt{6}$. This concludes the proof of Theorem 4.1.

V. Routing

We present a single destination routing algorithm (*SDR*) which will be used as a building block for multicasting and anycasting. In SDR, a message needs to be forwarded from a sender node s to a single target node t. The stretch $S_{\mathcal{A}}$ of a routing algorithm $\mathcal{A}$ is defined as $S_{\mathcal{A}} = \max_{s,t \in V} \frac{d_{\mathcal{A}}(s,t)}{d_{\mathcal{M}}(s,t)}$ where $d_{\mathcal{A}}(s,t)$ is the length of the path found by the routing algorithm. Clearly, we desire the stretch to be as small as possible, but this comes at a certain cost. In our approach, the stretch is coupled with the routing table size, i.e. lowering the stretch induces bigger routing tables. Our unicast routing result is summarized in the following theorem, where Θ is the diameter of the network, and Δ stands for the maximum degree of a node.

Theorem 5.1: For a fixed ε with $0 < \varepsilon \leq 2$, SDR routes messages with stretch $(1+\varepsilon)$ such that the chosen path by SDR is at most $(1+\varepsilon)d_{\mathcal{M}}(s,t)$. The header size of each message is at most $2\alpha\lceil \log \Theta \rceil + \lceil \log n \rceil$ bits, and the routing tables stored at each node require only $O\left(\frac{1}{\varepsilon}\right)^{\alpha} (\log \Theta)(O(\alpha) + \log \Delta)$ bits.

The routing scheme is based on the node labeling introduced in Section IV-D and is quite simple: The sender node s extracts from $\mathcal{L}(t)$ the name $\mathcal{N}$ of an arbitrary primary net-center $\gamma \in \Gamma_0$

[7] *Ordinary* means not in the extended dominance net.

that covers the target node t. This name $\mathcal{N}$ and the ID of t serve as header information in the message, which uses at most $2\alpha\lceil\log\Theta\rceil+\lceil\log n\rceil$ bits (cf. Section IV-B). Remember that $\mathcal{N}$ is a concatenation of the enumeration values of γ and its ancestors in the dominance tree. Therefore, $\mathcal{N}$ encodes the names of exactly one net-center (primary or secondary) per level that covers t (Lemma 4.2). Let us denote these net-centers by $\{c^0,c^1,c^2,c^3,\ldots,2^{\vartheta-2}\}$, where c^i is the net-center on level-i. The forwarding mechanism works as follows: Each node that receives a message finds the net-center c^i on the lowest level for which it has routing information and forwards the message in this direction. This is repeated until the message hits a direct neighbor of t which sends the message directly to t based on its neighborhood list.

In order to support this forwarding scheme, each node needs to store how to reach some of the net-centers in its surroundings. For this purpose, each net-center $\gamma\in\Gamma_i$ advertises itself to $\mathcal{B}_\gamma(\eta 2^i)$ with $\eta=\frac{8}{\varepsilon}+6$. I.e. every node in this ball stores how to reach γ on an optimal path. In fact, it suffices to store the neighbor node lying on the optimal path. This corresponds to the shortest path problem (in a restricted area) and can be obtained by distributed versions of the Dijkstra algorithm. Thus, the routing table of each node v stores how to reach any net-center $\gamma\in\Gamma_i$ iff $\mathrm{d}_{\mathcal{M}}(v,\gamma)\le\eta 2^i$. This can be seen as a mapping from the *name* of the net-center γ to the neighbor of v which lies on the shortest path to γ. Before we show in Section V-B that the routing information can be stored efficiently, we prove that SDR finds good routing paths.

A. $(1+\varepsilon)$-*Stretch Routing*

We first observe that once the message has reached a net-center covering t, it can be forwarded to a net-center covering t on a lower level:

Lemma 5.2 (Net-Center Hopping): Given a net-center $c^i\in\Gamma_i$ that covers the destination node t, the routing table of c^i contains entries for all net-centers $c^{i-x}\in\Gamma_{i-x}$ covering t. This holds for $x\le\lfloor\log_2\frac{\eta-2}{2}\rfloor$.

Proof: Suppose that $c^j\in\Gamma_j$ with $j<i$ is a net-center of t. Due to the routing table construction, all nodes in $\mathcal{B}_{c^j}(\eta 2^j)$ have routing table entries to c^j. Using the triangle inequality, we know that $\mathrm{d}_{\mathcal{M}}(c^j,c^i)\le 2^{j+1}+2^{i+1}$. Therefore, c^i is sure to have a routing table entry if $\mathrm{d}_{\mathcal{M}}(c^j,c^i)\le\eta 2^j$, which holds if $x=i-j\le\lfloor\log_2\frac{\eta-2}{2}\rfloor$. □

The following proof of the routing stretch describes the worst case scenario where the message visits the net-centers $\{c^i, c^{i-x}, c^{i-2x}, \ldots\}$, where c^i is the net-center towards which s forwards the message, cf. Figure 5.

Proof: (*Routing Stretch*) Let c^i be the net-center on the lowest level for which s has routing information. Following the routing algorithm, the message is first forwarded to c^i, from where it can be forwarded to c^{i-x} with $x=\lfloor\log_2\frac{\eta-2}{2}\rfloor$ (Lemma 5.2). This step is repeated until the message reaches a net-center on level-0, which directly delivers the message.

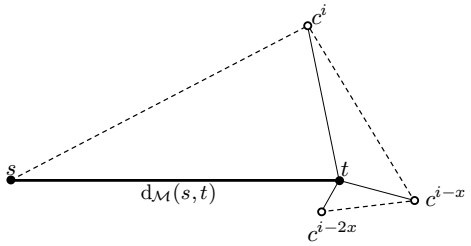

Fig. 5. Schematic illustration (out of scale) of the worst case scenario where a message to be sent from s to t first visits c^i, then c^{i-x} and so on, until it reaches a level-0 net-center covering t which directly delivers the message.

Thus, the total distance $\mathrm{d}_{\mathrm{SDR}}(s,t)$ of SDR is bounded by

$$\mathrm{d}_{\mathcal{M}}(s,c^i)+\mathrm{d}_{\mathcal{M}}(c^i,c^{i-x})+\mathrm{d}_{\mathcal{M}}(c^{i-x},c^{i-2x})+\cdots+1$$

Using the triangle inequality, we obtain that $\mathrm{d}_{\mathcal{M}}(s,c^i)\le\mathrm{d}_{\mathcal{M}}(s,t)+2^{i+1}$ and for $j\ge 0:\mathrm{d}_{\mathcal{M}}\left(c^{i-jx},c^{i-(j+1)x}\right)\le 2\cdot 2^{i-jx}+2\cdot 2^{i-(j+1)x}$. Therefore,

$$\begin{aligned}
\mathrm{d}_{\mathrm{SDR}}(s,t) &\le \mathrm{d}_{\mathcal{M}}(s,t)+4\sum_{j=0}^{\infty}2^{i-jx} && (1)\\
&= \mathrm{d}_{\mathcal{M}}(s,t)+4\sum_{j=0}^{\infty}\frac{2^i}{2^{\lfloor\log_2\frac{\eta-2}{2}\rfloor j}} && (2)\\
&\le \mathrm{d}_{\mathcal{M}}(s,t)+2^{i+2}\sum_{j=0}^{\infty}\left(\frac{4}{\eta-2}\right)^j && (3)\\
&\le \mathrm{d}_{\mathcal{M}}(s,t)+2^{i+2}\frac{\eta-2}{\eta-6} && (4)
\end{aligned}$$

We obtain (3) from (2) by observing that $2^{\lfloor\log_2\frac{\eta-2}{2}\rfloor j}\ge 2^{j\log_2\frac{\eta-2}{4}}=((\eta-2)/4)^j$. To obtain (4), note that the sum in (3) sums the elements of a geometric series with factor $\frac{4}{\eta-2}=\frac{\varepsilon}{2+\varepsilon}<1$.

Because s does not have a routing table entry for c^j with $j<i$, we deduce that $\mathrm{d}_{\mathcal{M}}(s,t)>(\eta-2)2^{i-1}$. This holds because $\mathrm{d}_{\mathcal{M}}(c^{i-1},t)\le 2\cdot 2^{i-1}$ and c^{i-1} advertises itself to $\mathcal{B}_{c^{i-1}}(\eta 2^{i-1})$. Therefore, s has a routing entry to c^{i-1} if $\mathrm{d}_{\mathcal{M}}(s,t)\le(\eta-2)2^{i-1}$.

Putting together the two results, we obtain that the routing stretch of SDR is

$$\begin{aligned}
S_{\mathrm{SDR}} &\le \frac{\mathrm{d}_{\mathrm{SDR}}(s,t)}{\mathrm{d}_{\mathcal{M}}(s,t)}\le 1+\frac{2^{i+2}\frac{\eta-2}{\eta-6}}{2^{i-1}(\eta-2)}\\
&= 1+\frac{8}{\eta-6}=1+\varepsilon.
\end{aligned}$$

This constitutes the proof for the routing stretch statement in Theorem 5.1. □

Throughout the proof, we assumed that the message visits the net-centers $\{c^i, c^{i-x}, c^{i-2x}, \ldots\}$, which is the worst case scenario. A considerable performance boost can be achieved in the average case if the routing algorithm tests in each step for a closer net-center. The key idea is that while the message is being routed towards c^j, it tests after each hop whether the

current node can route towards a closer net-center c^k with $k < j$. If this is the case, the message immediately routes towards c^k. This produces a shortcut towards t, reducing the routing path and therefore the stretch.

B. Compact Tables

The construction of the routing table is similar to the cover tree presented in Section IV-D with the main difference that each net-center $\gamma \in \Gamma_i$ covers an area which depends on ε, namely $\mathcal{B}_\gamma((\frac{8}{\varepsilon}+6)2^i)$. Because of this mutable coverage radius, we need to restate Lemma 4.2:

Lemma 5.3 (Parental Cover II): If a node v has routing information about net-center $\gamma \in \Gamma_i$, then v also has a routing entry to $\mathcal{P}(\gamma)$ unless γ is the root node.

Proof: Node v has routing information about $\gamma \in \Gamma_i$ iff $v \in \mathcal{B}_\gamma(\eta 2^i)$ with $\eta = \frac{8}{\varepsilon}+6$. Because $\mathrm{d}_\mathcal{M}(\gamma, \mathcal{P}(\gamma)) \leq 2^{i+1}$ and $\mathrm{d}_\mathcal{M}(\gamma, v) \leq \eta 2^i$, $\mathrm{d}_\mathcal{M}(v, \mathcal{P}(\gamma)) \leq (\eta+2)2^i$ using the triangle inequality. Therefore, it is sufficient to show that $(\eta+2)2^i \leq \eta 2^{i+1}$ such that $v \in \mathcal{B}_{\mathcal{P}(\gamma)}(\eta 2^{i+1})$. The inequality holds because $\eta = \frac{8}{\varepsilon}+6 \geq 2$. □

Using the same arguments as in Section IV-D, we can show that all net-centers for which a node v needs to store routing information can be be stored in a tree, where each tree node corresponds to one net-center. Each node of the tree only holds the enumeration value of the corresponding net-center γ and the routing information on how to reach γ. Recall that the enumeration value fits in 2α bits and the routing information is the neighbor node which lies on the optimal path between v and the corresponding net-center. Thus, the routing information uses at most $\log\Delta$ bits, where Δ is the maximum degree of a node.

Each node v needs to keep routing information for at most $2^{2\alpha}\left(\frac{8}{\varepsilon}+6\right)^\alpha$ net-centers per level (Corollary 3.1). Because there are at most ϑ levels, the tree has at most $2^{2\alpha}\left(\frac{8}{\varepsilon}+6\right)^\alpha\vartheta$ nodes, each of which needs to store $2\alpha+\log_2\Delta$ bits. We have already noted that such a tree can be stored in a flat data structure adding only two bits per node, which results in a total routing table size of $(2\alpha+\log_2\Delta+2)2^{2\alpha}\left(\frac{8}{\varepsilon}+6\right)^\alpha\vartheta = \mathrm{O}\left(\frac{1}{\varepsilon}\right)^\alpha(\log\Theta)(\mathrm{O}(\alpha)+\log\Delta)$ bits. This concludes the proof of Theorem 5.1.

VI. Multicasting

We now have developed all tools that allow for efficient multicasting from a sender node s to a set U of receivers. In a nutshell, the sender s approximates a minimum spanning tree (MST) on the set $s \cup U$ using Kruskal's algorithm and then routes the message along this tree towards all receivers. Note that in contrast to the centralized multicasting presented in [17], our approach is distributed. We show the following result:

Theorem 6.1: Consider a network $G = (V,E)$ on which a dominance net with the associated labeling and routing tables was created. Then, any sender node $s \in V$ can multicast messages to any set $U \subset V$ with constant stretch. The cost associated with the multicasting is at most $12(1+\varepsilon)$ times the cost of an optimal multicasting algorithm, which knows the entire network topology.

Proof: We need to show that the path along the MST approximation is at most $12(1+\varepsilon)$ longer than the optimal path, which is given by a minimum Steiner tree (MStT) on the set $s \cup U$ and the remaining nodes as Steiner points. The stretch is composed of three parts: First, a MST on $s \cup U$ is a 2-approximation of the corresponding MStT. This result was shown by Kou et al. in [8]. The $(1+\varepsilon)$ part is caused by the SDR routing scheme, which is responsible to forward the message along the tree. Lastly, the construction of the MST is based on the stretch-$\sqrt{6}$ distance labeling. As a result, Kruskal's algorithm may not choose the shortest, but up to a factor 6 longer edges in each step, which results in a MST approximation at most 6 times longer than the MST. □

VII. Anycast

As for the special case of anycasting, where a message has to be routed to exactly one node of a given node set U, we provide a constant stretch algorithm to deliver the message:

Theorem 7.1: The node labeling from Section IV-D and SDR from Section V allow for a $6(1+\varepsilon)$-approximation to anycast.

Proof: Based on the distance labeling, pick the node $u \in U$ which seems closest to the sender node s and send the message to u. Because of the stretch of the distance labeling, this approach may pick a receiver that is up to 6 times further away from s than the optimal receiver. The $(1+\varepsilon)$ factor stems from the SDR routing scheme. □

VIII. Distributed Dominance Net Construction

In this last section, we describe an efficient distributed algorithm to build the $\rho-$nets that constitute the dominance net. Recall from Section IV-A that the dominance net consists of ϑ $\rho-$nets with exponentially increasing ρ chosen from $\{1,2,4,\ldots,2^{\vartheta-1}\}$.

To start, we would like to point out that a *centralized* algorithm to build a $\rho-$net on $G = (V,E)$ is straightforward: Greedily pick an arbitrary $v \in V$, add v to Γ and remove all nodes in $\mathcal{B}_v(\rho)$ from V. Repeat until V is the empty set. By construction, the distance between any two net-centers is longer than ρ and each node $v \in V$ is covered by at least one net-center.

For our *distributed* algorithm, we exploit the fact that a $\rho-$net is a maximal independent set (MIS) for the ρ metric closure[8] $G^\rho = (V,E^\rho)$. This immediately leads to a simple distributed algorithm to create a $\rho-$net: create a ρ metric closure where each node $v \in V$ has all nodes in $\mathcal{B}_v(\rho)$ as direct neighbors. Then, run a distributed MIS algorithm on the closure and pick the nodes in the MIS to be the net-centers of the $\rho-$net. There exists a broad assortment of distributed

[8]The *n metric closure* of a graph $G = (V,E)$ is the graph $G^n = (V,E^n)$ with $E^n = \{(u,v) | u,v \in V \wedge u \neq v \wedge \mathrm{d}_\mathcal{M}(u,v) \leq n\}$

Algorithm 1: Dominance Net (Code for node v)

```
1  state = active
2  Main()
3    N = nodes in B_v(1)
4    Build MIS with neighbor set N
5    if v ∉ MIS then
6        state = passive
7    else
8        Inform B_v(1)                          v is a net-center of level-0
9        Add routing entries to v in B_v(η)     η = 4/ε + 3
10       JoinMIS(1)
11   end
12 JoinMIS(i)
13   ρ = 2^i
14   N = active nodes at most ρ away
15   Build MIS with neighbor set N
16   if v ∉ MIS then
17       state = passive
             v is not covered if ∄ net-center u of level-i s.t. v ∈ B_u(ρ)
18       while ∃w ∈ B_v(ρ/2) s.t. w not covered do
19           P = arbitrary uncovered node in B_v(ρ/2)
20           Send EXC(i) to P and wait for answer
21       end
22       Send ACK(i) to N
23   else
24       Inform B_v(ρ)                          v is a net-center of level-i
25       Add routing entries to v in B_v(ηρ)
26       Collect ACK(i) from all neighbors N
27       JoinMIS(i+1)
28   end
29 ReceiveMessage(EXC(i) from u)
30   state = excited, ρ = 2^i
31   Add temporary routing entries to v in B_v(ρ)
32   N = excited nodes at most ρ away
33   Build MIS with neighbor set N
34   if v ∈ MIS then
35       state = active
36       Inform B_v(ρ)                          v is a net-center of level-i
37       Add/validate routing entries to v in B_v(ηρ)
38       JoinMIS(i+1)
39   else
40       state = passive
41       Remove temporary routing entries to v in B_v(ρ)
42   end
43   Send ACK(state) to u
```

MIS algorithms, e.g. there is an elegant randomized algorithm with expected running time in $O(\log n)$ by Luby [11]. More recently, Kuhn et al. [9] described a deterministic MIS construction for bounded independence graphs with running time $O(\log\Delta\log^* n)$.

The above algorithm is realistic for small values of ρ. However, when ρ approaches Θ, the nodes tend to have huge neighborhoods, consisting of nearly all other nodes of the network. This may quickly exceed the memory capabilities of simple network nodes, especially when considering large networks. To overcome this issue, we present not only a relatively fast, but also *memory conservative* algorithm to construct the $\rho-$net hierarchy.

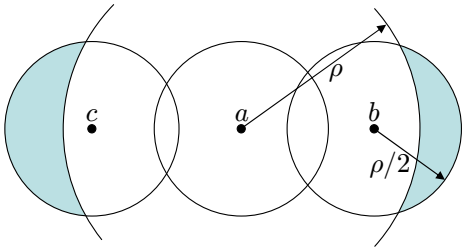

Fig. 6. A MIS on the ρ metric closure on the net-centers of a $\frac{\rho}{2}-$net does not necessarily cover all nodes of the network: Suppose that the $\frac{\rho}{2}-$net consists of $\{a,b,c\}$. One possible MIS on these 3 nodes is to pick a, which covers b and c, but not the shaded areas. Note that coverage areas may have arbitrary shape and need not be circular, as drawn in this example.

A. *Sequential* $\rho-$nets

Algorithm 1 describes in high-level pseudocode the steps each node v of the network performs in order to create the dominance nets. At the end of the algorithm, each node knows for which $\rho-$nets it is a net-center, and it already holds the complete routing table for the SDR routing scheme presented in Section V. In a nutshell, the algorithm is recursive and performs the following two steps to build the $\rho-$net of level-i.

1) Approximate a $\rho-$net given the $\frac{\rho}{2}-$net, i.e. build a MIS on the ρ metric closure of Γ_{i-1}, the independent nodes join Γ_i. Note that this MIS only guarantees that the net-centers of the $\frac{\rho}{2}-$net are covered, but it does not ensure coverage of all nodes of the network.
2) Add additional net-centers to Γ_i until all nodes $v \in V$ are covered by at least one $\gamma \in \Gamma_i$.

We obtain the desired memory relaxation by building the $\rho-$nets sequentially on top of each other. Initially, all nodes participate in a MIS construction on a 1 metric closure of the network, which results in the $1-$net (lines 2–4). Then, for any level-i, the ρ metric closure is constructed only with the net-centers Γ_{i-1} and induces at most $2^{2\alpha}$ neighbors per node, independent of the network size (Property 3.1). The downside of this approach is a longer running time and that the MIS on the net-centers of level-$(i-1)$ does not necessarily cover all nodes of the network, see Figure 6 for an example. This requires a second phase, where additional net-centers are added until full coverage is obtained.

The recursive call to join the next higher MIS is handled on the lines 13–15. Note that only nodes that are net-centers on level-$(i-1)$ participate in the MIS construction for level-i. If node v does not make it into the MIS of level-i, it becomes passive (line 17) and is responsible that all nodes in $\mathcal{B}_v(2^{i-1})$ are covered (lines 18–22). This covering algorithm works in several rounds, where v participates as long as $\mathcal{B}_v(2^{i-1})$ contains uncovered nodes. In each round, each net-center whose $\mathcal{B}(2^{i-1})$ contains uncovered nodes picks an uncovered node and sets its state to *excited* by sending an EXC message. The set of excited nodes build a MIS on their ρ metric closure (lines 32–33), and the independent excited nodes join the $(2^i)-$net (lines 35–38). The temporary routing entries added on line 31 and, if the node does not make it into the MIS,

removed on line 41 enable the MIS algorithm to exchange messages on the optimal path between any two neighbors in the corresponding metric closure.

Due to space constraints, we did not address the following issues in the pseudocode of Algorithm 1.

- If a node v receives several EXC requests in the same round, it executes lines 30–42 only once, but finally acknowledges on line 43 to all nodes that sent the request.
- The net-centers need to obtain their enumeration values (see Section IV-B) in order to construct their labels. Because these labels are incomplete until all $\rho-$nets are constructed, the routing table entries (constructed on lines 9, 25, 31, 37) are built using a partial labeling.
- The algorithm requires a rough synchronization between consecutive MIS constructions, such that a MIS construction only starts when the previous one finished. This can be obtained during the construction of the metric closure for the next MIS (lines 14 and 32), where any node can ask the others to wait. To support this synchronization, each node that joined the $\rho-$net waits until all of its neighbors (in the metric closure) have terminated the coverage procedure (line 26).
- Whenever $N = \emptyset$ on line 14, node v trivially joins the MIS and tests whether it covers all nodes of the network. If this is the case, v becomes the root of the network and stops the recursive call to JoinMIS().
- Before the first MIS can be constructed, the nodes need to determine the lowest cost associated with any edge of the network, as to determine the scaling factor that scales this cost to 1.

Theorem 8.1: Algorithm 1 constructs a dominance net as described in Section IV-A. Furthermore, the algorithm requires to build a total of at most $\vartheta(1+2^{3\alpha}) = O(\log\Theta)$ MIS, which gives an upper bound on its running time.

Proof: We show the first property by induction on the levels: Initially, all nodes participate in a MIS election on the 1-metric-closure, which corresponds to building a MIS on the network where only the edges with cost 1 are considered. By construction, the resulting MIS is a $1-$net.

Given the points Γ_{i-1} of a $2^{i-1}-$net, the algorithm first approximates Γ_i by a MIS on the 2^i metric closure of Γ_i. By construction, $a,b \in \Gamma_i \implies d_{\mathcal{M}}(a,b) > 2^i$. Because only nodes that are not covered by Γ_i may become excited (line 19), the addition of any excited node to Γ_i does not break this property. Furthermore, the excited nodes added to Γ_i in one round are independent w.r.t. the 2^i metric closure on the set of excited nodes (lines 32–34). Therefore, the minimum distance between net-centers of level-i is preserved. Because every $\gamma \in \Gamma_{i-1}$ repeats the completion process until all nodes in $\mathcal{B}_\gamma(2^{i-1})$ are covered (line 18), Γ_i finally covers all nodes and is a valid 2^i-net.

As for the running time, we show that the algorithm may have to build at most $1+2^{3\alpha}$ MIS per level: The first MIS is built on line 15, the remaining ones on line 33. Therefore, we need to show that a node v may execute the while-loop of line 18 at most $2^{3\alpha}$ times until all nodes in $\mathcal{B}_v(2^{i-1})$ are covered. We first note that an excited node P (line 19) may not join the MIS if another excited node Q with $d_{\mathcal{M}}(P,Q) \leq 2^i$ joins. Therefore, any excited node Q that may interfere with P must lie in $\mathcal{B}_v(3\cdot 2^{i-1})$, and the net-center that excited Q in $\mathcal{B}_v(2^{i+1})$. By Property 3.2, there are at most $2^{3\alpha}$ net-centers of level-$i-1$ in $\mathcal{B}_v(2^{i+1})$ and therefore at most $2^{3\alpha}-1$ excited nodes that may interfere with P. Just looking at P, we know that either P or one of its interfering nodes is selected in each round of the while loop. Thus, after at most $2^{3\alpha}-1$ rounds, P can join the MIS, which shows that at most $1+2^{3\alpha}$ MIS are necessary per level. Because there are at most ϑ levels, the number of MIS to construct is bounded by $\vartheta(1+2^{3\alpha})$. □

IX. Conclusion

This paper discussed an effective all-in-one solution for unicasting, anycasting, and multicasting in static wireless sensor and mesh networks. Being the probably most important building block, we feel that support for all of these routing schemes is crucial as to allow for performance-oriented applications building on top. Just as with IP-routing in the Internet, this routing scheme may be extended with security, support for mobility, fault tolerance and many other features, which we leave open for future work.

References

[1] I. Abraham, C. Gavoille, A. Goldberg, and D. Malkhi. Routing in Networks with Low Doubling Dimension. In *ICDCS*, 2006.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing With Guaranteed Delivery in Ad Hoc Wireless Networks. In *DIAL-M*, pages 48–55, 1999.

[3] H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou. On Hierarchical Routing in Doubling Metrics. In *SODA*, pages 762–771, 2005.

[4] M. Dyer, J. Beutel, and L. Thiele. S-XTC: A Signal-Strength Based Topology Control Algorithm for Seonsor Networks. In *2nd Int. Workshop on Ad Hoc, Sensor, and P2P Networks*, March 2007.

[5] Q. Fang, J. Gao, and L. J. Guibas. Locating and Bypassing Routing Holes in Sensor Networks. In *INFOCOM*, 2004.

[6] C. Gavoille and M. Gengler. Space-Efficiency for Routing Schemes of Stretch Factor Three. *J. Parallel Distrib. Comput.*, 61(5):679–687, 2001.

[7] G. Konjevod, A. W. Richa, and D. Xia. Optimal-Stretch Name-Independent Compact Routing in Doubling Metrics. In *PODC*, pages 198–207, 2006.

[8] L. Kou, G. Markowsky, and L. Berman. A Fast Algorithm for Steiner Trees. *Acta Informatica*, 15:141–145, 1981.

[9] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs. In *DISC*, September 2005.

[10] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric Ad-Hoc Routing: of Theory and Practice. In *PODC*, pages 63–72, 2003.

[11] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.

[12] M. O'Dell, R. O'Dell, M. Wattenhofer, and R. Wattenhofer. Lost in Space Or Positioning in Sensor Networks. In *REALWSN*, June 2005.

[13] D. Peleg and E. Upfal. A Trade-Off between Space and Efficiency for Routing Tables. *J. ACM*, 36(3):510–530, 1989.

[14] S. Schmid and R. Wattenhofer. Algorithmic Models for Sensor Networks. In *WPDRTS*, April 2006.

[15] A. Slivkins. Distance Estimation and Object Location via Rings of Neighbors. In *PODC*, pages 41–50, 2005.

[16] K. Talwar. Bypassing the Embedding: Algorithms for Low Dimensional Metrics. In *STOC*, pages 281–290, 2004.

[17] P.-J. Wan, G. Calinescu, and C.-W. Yi. Minimum-Power Multicast Routing in Static Ad Hoc Wireless Networks. *IEEE/ACM Trans. Netw.*, 12(3):507–514, 2004.