The Construction of Distributed Multicast Tree in Mobile Bitcoin Network

Yichi Zhang

Abstract

Blockchain technology aplied in Mobile Bitcoin Networks(MBNs) brings high security performance but leads to a long time delay at the same time. In this paper, we design and analyze a algorithm to build multicast trees in MBNs so as to reduce the cost and time delay while verifying the authenticity of the transaction. Because the network is a distributed one, the construction of a multicast tree in it is clearly a NP-hard problem. We analyze the efficiency of our algorithm by two performances including tree length and time complexity. We demonstrate the total tree length is upper bounded by $O(\sqrt{n})$ which is in the same order as Steiner tree. We prove that the running time of the construction is $O(\sqrt{nlogn})$, also the best among all solutions by now. We will also prove our algorithm by simulation and real data of the bitcoin network.

1. INTRODUCTION

In block chain modal, a transaction has to be buried into a block waiting for a kind of special nodes called miner to dig it out. By solving some difficult math problems, a miner can dig the block out and get its bonus. If a block contained thousands of transaction is digged out by a miner, the miner has to broadcast it to the entire network so that all the other nodes can varify the authenticity of the transactions in this block. This process is very time conssuming because the number of nodes in the network is really big and the process of virification brings long delay. As a result, both parties of the transaction have to wait for a long time until the varification process is finished.

Mobile Bitcoin Network(MBN) is a network of wireless nodes, in which every node is a bitcoin user's mobile terminal. That's to say the cost of connecting two nodes is directly influenced by the distance between them. This network is also distributed because it is decentralized and each node only has a limited knowledge of network. Multicast tree is topological structure of a one-to-many group communication where data transmission is addressed to a group of destinations simulaneously. In this paper, the message is multicasted by miner which is also the source node of the multicast tree to multiple receivers in MBNs.

A number of algorithms of the construction of multicast tree have already be designed by researchers so as to minimize the routing and time complexity. However, none of these algorithms fits the environment of MBNs based on the fact that they didn't consider the influences of block chain characteristics. Furthermore, we reduce the total tree length by nearly a half of what it was with the help of polar coordinate calculation.

Steiner Tree Problem aims to find a tree in a given network that spans a set of terminals with the least total weight. This problrm has been proved to be NP-hard A new algorithm inspríred by taking the advantage of the Steiner tree property called Toward Source Tree (TST) [1] is proposed recently to construct multicast tree in Wireless Sensor Networks. The tree length of this algorithm is proved to be proportional to $\sqrt{m}$ while m is the number of receivers in the network. And the running time is $O(\sqrt{nlogn})$ which is the best of all.

In this paper, we design a new algorithm based on TST to construct an approximate minimun-length multicast tree in Mobile Bitcoin network to reduce the cost and delay of the conmmunication among the nodes. In this way, we can approximately minimize the delay of the transaction while using bitcoin. We demonstrate that the total tree length of our algorithm is in the same order as Steiner tree and TST, but we reduce the coefficient by nearly a half. And the running time of it can be proven to also be $O(\sqrt{nlogn})$, the best among all existing solutions for large multicast groups.

In the rest of this paper, we introduce our network model in section 2. In section 3, we discuss our algorithm in details. We analyze the total tree length of our algorithm in section 4 and compare it with existing algorithms. In section 5, we continue to analyze the ruuning time coplexity. Finally, we conclude this paper and discuss the future works in section 6.

2. NETWORK MODEL

We abstract the problem of the construction of the tree under bitcoin network into a mathematical model through some assumptions. Firstly, we assume that every node in the network has a coordinate (r, θ) which is used to calculate the distance between it and the source. The coordinate is determined by the geographic location of the node. For ease of calculation, we consider the coordinate of the source(S) is $(0,0)$ and the entire area is normalized, which means the farest node from S has a coordinate $r = 1$. In this way, every node can be represented by a unique coordinate. Because these coordinates are only an abstraction of the geographical locations of nodes, they do not destroy the anonymity of the entire network.

Then we assume that each node has a weight $(0 \leq W_i \leq 1)$ to indicate the authoritativeness of this node when confirming the correctness of messages. This weight is determined by the historical accuracy of the node verification. For example, a node with $W_i = 1$ means this node i never made any mistake while verifying the block. However, a node with $W_j = 0$ means this node j is an interference node because it cannot correct judgment at all.

In fact, we don't need to broadcast the block to every node in the network in order to minimize the delay of verification. The number of nodes that S(miner who digs to the block firstly) broadcasts the block to is determined by how safe the parties to the transaction want to insure. Before S broadcast the block into the network, parties of transactions in this block will announce a goal weight W_0. If and only if the sum of weights of the nodes that the source choose to verify the block is larger than W_0, the verification process is finished and the transactions in this block are verified. In order to reduce the cost of tree construction, we select only a part of the nodes whose weights are larger in contrast as our receivers. This is because it doesn't worth to contact if a node has a weight of only 0.1 for example due to its low contribution to the sum of weights but may lead to a significant increase in tree length. Therefore, we only choose nodes with a weight larger that W_c. W_c is the critical value of weight which is determined by the distribution function of nodes weight. We assume the total number of nodes is n while the number of receivers is m. Therefore, the termination function of the tree construction should be:

$$\sum_{N_i \in \{T\}} W_i \geq W_0$$

Table 1. Notations and Definitions

n	the total number of nodes in the network
m	the number of receivers
N_i	node i
$\{T\}$	the collection of all nodes of the nulticast tree
W_i	weight of node i
W_0	goal weight of the tree construction

3. ALGORITHM

In this section, we discuss our algorithm in details. The algorithm can be divided into four phases basicly. In the first phase the source node(miner) broadcast a message in an area with radius R_1 and the first batch of receivers are chosen in this phase. In the second phase, every receiver choose another receiver that is closer to S to connect to, and a temperary tree is formed. In phase three, we eliminate cycles which may be formed in the second phase in order to avoid redundant transmissions and assure the topology is a tree. Finally, we calculate the total weight of the nodes in R_1 area and compare it with W_0. If $W_1 \geq W_0$ the process is finished, if not, we expand the search radius to R_2 and repeat the first three phases, until the k^{th} search satisfy the termination function.

As mentioned before, our algorithm is based on an algorithm called Toward Source Tree(TST). In fact, in every search process, we use TST to construct a multicast tree first. However, because the number of receivers in our algorithm is unknown at the begining, we can only expand the radius in turn until we find a sufficient number of nodes.

A. Phase 1: Searching and Identifying Receivers

In this phase, the source broadcast its coordinate and the cricial weight W_c in an area of radius R_1. Every node in this area received this message will firstly comfirm if its weight is larger than W_c, if so it is chosen as a receiver. This phase is necessary because before it nodes don't know whether they are receivers or not. The broadcast information will notify the nodes who are selected into the multicast group.

B. Phase 2: Building of temporary tree

In this phase, each receiver choose another receiver that has a shorter Euclidean distances to the source than itself to connect with. If no such receiver can be found, the receiver connects with the source directly. Then a temporary tree is formed which contains all the receivers in this area. Next we find the minimum-hop path between each pair of members that are directly connected in the temporary tree. Finally, the construction of multicast tree in this area is almost finished.(We will guarantee the tree structure by eliminating cycles in phase three.)

The node contact with each other by sending request message to ask for connection and respond message to comfirm the connection. For energy saving, each node only sends request message in an area of radius r_c. The coverage range r_c sets the range within which the multicast member searches for its neighboring members. The Euclidean distance between the sender and current node can be calculated with sender location, and messages will be discarded if the distance is larger than rc. If no effectve relay can be found within r_c, the node will expand the searching radius so as to find a suitable relay to connect with.

C. Phase 3: Eliminating Cycles

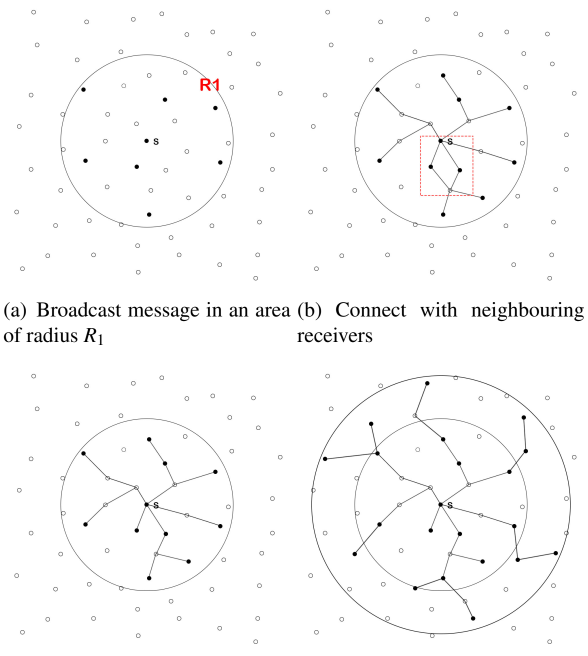
When a node received more than one request messages, cycles might be formed. Therefore, we can not garantee the multicast tree constructed in phase 2 has a topologic of tree. As a result, checking the existence of cycles and eliminating them is necessary.

Suppose a node u acts as a relay for $k(k \geq 1)$ pairs of receivers in the multicast group, which are directly connected in the temporary tree, denoted as $(R_{11};R_{12}),(R_{21};R_{22}),...,(R_{k1};R_{k2})$. Let us assume that in each pair, R_{i1} is closer to source than $R_{i2}(1ik)$. A relay stores its previous and the next hop of the path from R_{i1} to R_{i2}, and they are denoted as PH_i and NH_i respectively. Then it chooses one pair randomly, $(R_{j1};R_{j2})$ and keeps the information: $(R_{j1},R_{j2},PH_j,NH_j)$. For other pairs $(R_{i1};R_{i2})$ where $R_{i1} \neq R_{j1}$, the relay modifies their information as $(R_{j1},R_{i2},PH_j,NH_i)$. And then the cycle is eliminated.

D. Phase 4: Weight Calculation and Expanding Search Area

Because the number of receivers is determined by the goal weight W_0 and the distribution function of the node weight in the network. Therefore, we may have to repeat the first three phases several times until the termination function is satisfied.

In this phase, we calculate the sum of weights of all the members in multicast group and compare it with W_0. If the sum is larger, the construction of the multicast tree is finished. If not, however, we expand the search radius and repeat the first three phases until the k^{th} search satisfy the termination function. Nevertheless, after the search area is expanded, the construction has a little different. Because the construction of multicast tree is already finished within R_{i-1}, the nodes in the area between R_{i-1} and R_i only have connect with any node in the area of R_{i-1}. So the we choose the leaf nodes within R_{i-1} as the source node to connect with, because this is clearly the most cost-saving solution.



(a) Broadcast message in an area of radius R_1
(b) Connect with neighbouring receivers
(c) Eliminating circles
(d) Expand search radius and repeat phases 1 to 3

Figure 1. Four phases of the algorithm

4. PERFORMANCE ANALYSIS

The previous secton described our algorithm in details. In this section, we analyze the performance of our algorithm in order to prove that the algorithm we proposed has its advantages. And we mainly analyze the total tree length and time complexity of the algorithm.

4.1. Length Analysis

The analysis of the total tree length is of great significance because it influences the cost of costructing the multicast tree directly. Furthermore, the total tree length also has connection with the time delay while multicasting messages in the tree. Firstlt, we calculate the expected value of the temporary tree length under the uniform distribution of nodes. Then we generalize our results to the situation that nodes in the network is distributed generally following a distribution function $f(r,\theta)$.

A. Temporary Tree in Uniform Distribution

We fisrtly calaulate the temporary tree length in an area with radius R_1. And we assume the number of receivers in this area is $m1$. We divide the circle a into m equal area rings. The i^{th} ring's radius is R_i. Then:

$$\pi R_i^2 = \frac{\pi R_1^2}{m_1} i \Rightarrow R_i = \sqrt{\frac{i}{m_1}} R_1$$

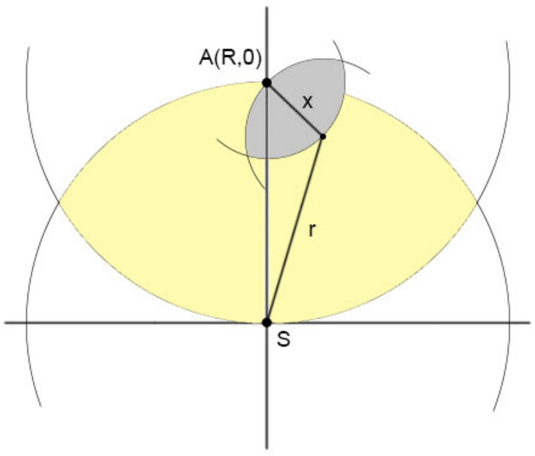


Figure 2. The average length of a receiver with coordinate (R_i, θ) connecting with its neighbour receiver

Because the nodes are uniformly distributed in a circle area, the direction angle θ has nothing to do with the average tree length. So the probability that a node is in the i^{th} ring is $\frac{1}{m_1}$. As we can see from Figure 2, the neighbouring receiver which could be chosen as the relay of node $A(R_i, 0)$ must be located in the yellow area. So we calculate the average of the tree length by sum all the expected length between node A and the neighbour node in yello area. We assume a node is chosen as receiver if and only if there is no node in the grey area in Figure 2. p_j is the probability that a node in circle $j (j \le i)$ is chose as relay.

$$p_j \le (1 - \frac{aR_j^2}{\pi R_1^2})^{m_1 - 1}$$

Where $a = \frac{\pi}{3} - \frac{\sqrt{3}}{4}$. Then we can calculate the expected length as following:

$$E_1\{L_v\} = m_1 \cdot \frac{1}{m_1} \sum_{i=1}^{m_1} \sum_{j=1}^{i} [1 - (1 - \frac{1}{2m_1})^{m_1 - 1}] p_j \cdot R_j$$

Because m_1 is a large number.

$$\begin{aligned} E_1\{L_v\} &= \sum_{i=1}^{m_1} \sum_{j=1}^{i} (1 - e^{\frac{m_1 - 1}{2m_1}}) \cdot e^{-\frac{aj(m_1 - 1)}{\pi m_1}} \cdot \sqrt{\frac{j}{m_1}} R_1 \\ &= \frac{(1 - e^{-\frac{1}{2}}) R_1}{\sqrt{m_1}} \sum_{i=1}^{m_1} \sum_{j=1}^{i} e^{-\frac{aj}{\pi}} \cdot \sqrt{j} \end{aligned}$$

Then we want to show the upper bound of $E_1\{L_v\}$. In order to get that, we first consider the sum:$\sum_{j=1}^{i} e^{-\frac{aj}{\pi}} \cdot \sqrt{j}$ with the function $f(x) = e^{-\frac{ax}{\pi}} \cdot \sqrt{x}$.

$$f'(x) = \frac{1}{2} \cdot e^{-\frac{ax}{\pi}} \cdot x^{-\frac{1}{2}} \cdot (1 - \frac{2ax}{\pi}) \tag{1}$$

$$f''(x) = -((\frac{1}{4} x^{-\frac{3}{2}} + \frac{a}{2\pi} x^{-\frac{1}{2}})(1 - \frac{2a}{\pi} x) + \frac{a}{\pi} x^{-\frac{1}{2}}) \cdot e^{-\frac{ax}{\pi}} \tag{2}$$

Then we get that $f(x)$ is monotone decreasing as $x \in (0, \frac{\pi}{2a})$ and is monotone increasing as $x \in (\frac{\pi}{2a}, +\infty)$. Moreover we know $f(x)$ is a concave function as $x \in (0, \frac{\pi}{2a})$.($f'(x)$ and $f''(x)$ represent the first order derivative and the second derivative of the function f respectively in the equation (1) and (2).) Now we can get the upper bound of $\sum_{j=1}^{i} e^{-\frac{aj}{\pi}} \cdot \sqrt{j}$ as following:

$$\begin{aligned} \sum_{j=1}^{i} e^{-\frac{aj}{\pi}} \cdot \sqrt{j} &= \sum_{j=1}^{i} e^{-\frac{aj}{\pi}} \cdot \sqrt{j} \cdot 1 \\ &< \int_0^j f(x)dx + \frac{1}{2} \cdot \int_0^{\frac{\pi}{2a}} f(\frac{\pi}{2a})dx \\ &< \int_0^\infty f(x)dx + \frac{1}{2} \cdot \int_0^{\frac{\pi}{2a}} f(\frac{\pi}{2a})dx \\ &= \int_0^\infty e^{-\frac{ax}{\pi}} \cdot \sqrt{x}dx + \frac{1}{2} \cdot \int_0^{\frac{\pi}{2a}} f(\frac{\pi}{2a})dx \\ &= \frac{\pi}{a} \cdot \sqrt{\frac{\pi}{a}} \int_0^\infty e^{-t} \cdot \sqrt{t}dt + \frac{\pi}{4a} \cdot \sqrt{\frac{\pi}{2a}} \cdot e^{-\frac{\pi}{2a}} \\ &= \frac{\pi}{a} \cdot \sqrt{\frac{\pi}{a}} \Gamma(\frac{3}{2}) + \frac{\pi}{4a} \cdot \sqrt{\frac{\pi}{2a}} \cdot e^{-\frac{\pi}{2a}} \\ &= \frac{\pi}{2a} \cdot \sqrt{\frac{\pi}{a}} \Gamma(\frac{1}{2}) + \frac{\pi}{4a} \cdot \sqrt{\frac{\pi}{2a}} \cdot e^{-\frac{\pi}{2a}} \\ &= \frac{\pi^2}{2a\sqrt{a}} + \frac{\pi}{4a} \cdot \sqrt{\frac{\pi}{2a}} \cdot e^{-\frac{\pi}{2a}} \end{aligned} \tag{3}$$

In the process (3), we use the proporties of Γ function: (i)$\Gamma(s) = \int_0^\infty e^{-t} \cdot t^{(s-1)} dt$ (ii) $\Gamma(s+1) = s \cdot \Gamma(s)$ (iii) $\Gamma(\frac{1}{2}) = \sqrt{\pi}$.

Then it is easy for us to get the upper bound of $E_1\{L_v\}$. In fact:

$$\begin{aligned}E_1\{L_v\} &= \frac{(1-e^{-\frac{1}{2}})R_1}{\sqrt{m_1}}\sum_{i=1}^{m_1}\sum_{j=1}^{i} e^{-\frac{aj}{\pi}}\cdot\sqrt{j} \\ &< \frac{(1-e^{-\frac{1}{2}})R_1}{\sqrt{m_1}}\sum_{i=1}^{m_1}(\frac{\pi^2}{2a\sqrt{a}}+\frac{\pi}{4a}\cdot\sqrt{\frac{\pi}{2a}}\cdot e^{-\frac{\pi}{2a}}) \\ &= \frac{(1-e^{-\frac{1}{2}})R_1 m_1}{\sqrt{m_1}}(\frac{\pi^2}{2a\sqrt{a}}+\frac{\pi}{4a}\cdot\sqrt{\frac{\pi}{2a}}\cdot e^{-\frac{\pi}{2a}}) \\ &< C\cdot R_1\sqrt{m_1}\end{aligned}$$

where $C = 4.096$ is a constant. When we normalize the area, $R_1 = \frac{1}{\sqrt{\pi}}$. The expected tree length in the first search process is upper bounded by

$$E_1\{L_v\} \le 2.311\sqrt{m}$$

Since building a tree under multiple search processes is a Markov process, that is to say, the expected tree length while building a tree under multiple search processes within a radius R is actually the same as building a tree directly within this area. As a result, if the area is a unit circle, phase 4 won't increase the expected tree length at all. That's to say, the expected tree length of the whole process is also upper bounded by

$$E\{L_v\} \le 2.311\sqrt{m}$$

B.Temporary Tree in General Distribution

We divided the uniform circle into pieces small enough each with an area of $r\cdot dr\cdot d\theta$. As a result, in each small area we can consider the nodes are uniformly distributed. Then we can calculate the expected value od total tree length by integeal r and θ over the whole region. Here we assume the density function of the nodes is $f(r,\theta)$ which is a function of redius r and the direction angle θ.

$$\begin{aligned}E\{L_v\} &= \iint[\frac{d\theta}{2\pi}\cdot C\cdot r\cdot\sqrt{f(r,\theta)\pi r^2}- \\ &\quad \frac{d\theta}{2\pi}\cdot C\cdot(r-dr)\cdot\sqrt{f(r,\theta)\pi(r-dr)^2}] \\ &= \frac{C}{\sqrt{\pi}}\iint[r^2-(r-dr)^2]\cdot\sqrt{f(r,\theta)}\cdot dr\cdot d\theta \\ &= \frac{C}{\sqrt{\pi}}\int_0^{2\pi}d\theta\int_0^{R}[r\cdot\sqrt{f(r,\theta)}]dr\end{aligned}$$

While $C = 4.096$ which is equal to the conclusion in uniform distribution.

4.2. Running Time Analysis

Time efficiency is also important when evaluate the quality of multicast routing algorithm. The lower the time complexity, the shorter the delay in building this tree. The time analysis of our algorithm is discussed below.

The running time of one search process(phase 1 to 3) is proven in TST algorithm to be:

$$O(\sqrt{\frac{n}{logn}}) \le E(t) \le O(\sqrt{nlogn})$$

Therefore, we can focus on phase 4 while calculating the running time. In phase 4, the search process is repeated k times until the last time of search the termination function is satisfied. So the upper bound of the running time can be calculated as:

$$E(t) \le O(\sum_{i=1}^{k}\sqrt{n_i logn_i})$$

Where k is a small quantity compared with n. Specially k is constant when the nodes are uniformly diatributed because the density of the nodes can be infered by the first search process. And n_i satisfy the following function:

$$\sum_{i=1}^{k} n_i = n$$

Then we have:

$$\begin{aligned}\sum_{i=1}^{k}\sqrt{n_i logn_i} &\le \sqrt{k(\sum_{i=1}^{k}n_i logn_i)} \\ &\le \sqrt{k\cdot nlog(\frac{n}{k})}\end{aligned}$$

Because k can be seen as a constant, the total running time is also in the same order of TST, which is one of the best among all the algorithms. The total running time is:

$$E(t) \le O(\sqrt{nlogn})$$

5. CONCLUSIONS & FUTURE WORK

In this paper, we propose a new algorithm to construct distributed multicast tree in Mobile Bitcoin Networks. We analyze the performance of it to make sure that this algorithm minimizes the time and cost while multicasting a message. In this way, we prove that our algorithm can reduce the delay while verifying the authenticity of the block, which is one of the biggest weakness of block chain model. We also prove that the expected tree length of our algorithm is better than that of TST, while we remain the time complexity at the same order.

For future research, several directions worthy dissused are listed as follows.

- We only focus on tree length and running time analysis in this paper. However, it is also interesting to pay attention to the energy efficiency of the algorithm. Because the mobile terminals like mobile phones are supported by battery. The reduction in energy consumption is of great significance.

- Real data of the bitcoin network is needed to verify the correctness of the theory proposed in this paper. However, the real data of a mobile bitcoin network seems hard to be found based on the fact that the network is completely anonymized.

- The relationship between the distribution function of node weight and the critical weight W_c is still unclear. We only know that there must be some kind of connection between the them, but this connection has not been described quantitatively.

References

[1] H. Gong, L. Fu, X. Fu, L. Zhao, K. Wang and X. Wang. Distributed Multicast Tree Construction in Wireless Sensor Networks. In Proc. of IEEE, 2015.

[2] M. Segal, J. Crowcroft and L. Levin. Improved structures for data collection in wireless sensor networks. In Proc. of IEEE INFOCOM, pp. 13751383, 2014.

[3] H. R. Sadjadpour, Z. Wang and J. J. Garcia-Luna-Aceves. A unifying perspective on the capacity of wireless ad hoc networks. In Proc. of IEEE INFOCOM, 2008

[4] C. Oliveira, P. Pardalos, M. Resende. Optimization Problems in Multicast Tree Construction. DOI: 10.1007/978-0-387-30165-5-25, Januaty 2006

[5] A. Sapirshtein, Y. Sompolinsky, A. Zohar. Optimal selfish Mining Strategies in bitcoin. In Proc. of IEEE

[6] C. Bo, X-Y. Li, T. Jung, X. Mao, Y. Tao and L. Yao. SmartLoc: push the limit of the inertial sensor based metropolitan localization using smartphone In Proc. of ACM MobiCom, 2013.

[7] D. Koutsonikolas, S.M. Das, Y. C. Hu and I. Stojmenovic. Hierarchical geographic multicast routing for wireless sensor networks. In Wireless networks, Vol. 16, No. 2, pp. 449466, 2010.

[8] T. Rothvoss, J. Byrka, F. Grandoni and L. Sanita. Steiner tree approximation via iterative randomized rounding. In Journal of the ACM, Vol. 60, No. 1, pp. 6, 2013.

[9] R. M. Karp. Reducibility among Combinatorial Problems. Springer, 1972.

[10] P. M. Ruiz, J. A. Sanchez and I. Stojmenovic. Gmr: Geographic multicast routing for wireless sensor networks. In Proc. of IEEE SECON, pp. 2029, 2006