(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 March 2006 (02.03.2006)

PCT

(10) International Publication Number
WO 2006/023282 A2

(51) International Patent Classification:
H04L 12/28 (2006.01)

(21) International Application Number:
PCT/US2005/027687

(22) International Filing Date: 3 August 2005 (03.08.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/919,139 16 August 2004 (16.08.2004) US

(71) Applicant *(for all designated States except US)*: **MOTOROLA, INC.** [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **SHI, Qicai,** [CN/US]; 5462 NW 122 Drive, Coral Springs, FL 33076 (US). **HUANG, Jian,** [US/US]; 431 NW 118 Way, Coral Springs, FL 33071 (US). **HUANG, Yan,** [CN/US]; 879 Hawthorn Terrace, Weston, FL 33327 (US).

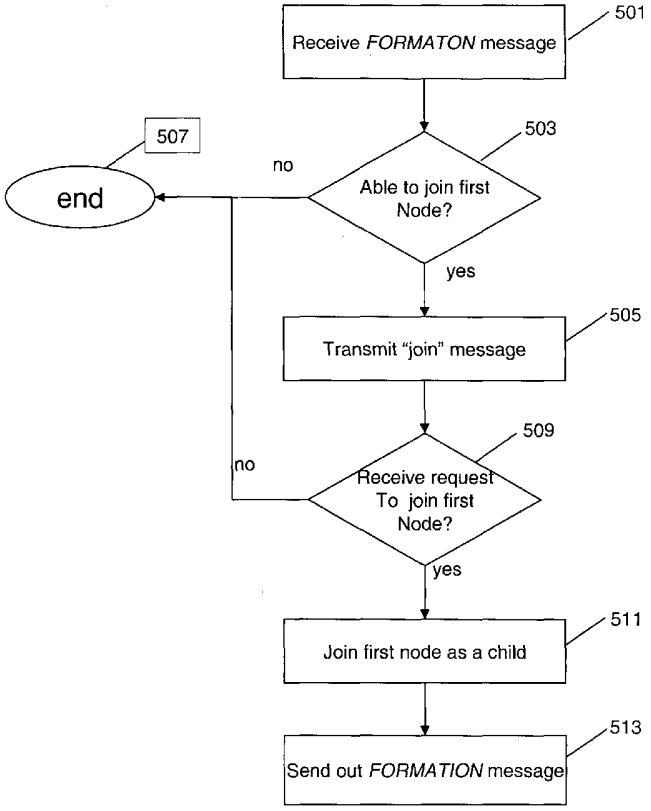
(74) Agent: **HAAS, Kenneth A.**,; 1303 East Algonquin Road, Schaumburg, IL 60196, (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR OPERATING AN AD-HOC COMMUNICATION SYSTEM



(57) Abstract: An ad-hoc communication system is established where each node within the system may have multiple children nodes and parent nodes. An address scheme is proposed where each node may comprise more than one address, each address being based upon one of the parent's address. Because each node may have more than one parent, a bridge, or shortcut may exist between network branches.

WO 2006/023282 A2

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR OPERATING AN AD-HOC COMMUNICATION SYSTEM

Field of the Invention

The present invention relates generally to communication systems and in particular, to a method and apparatus for operating an ad-hoc communication system.

Background of the Invention

Wireless ad hoc communication systems allow a number of devices (nodes) to communicate with each other without a need to pass communications through any particular node, or infrastructure. Data communication between devices occurs by firstly discovering intervening nodes that exist between a source and a destination node, and then relaying the data from the source node, through the intervening nodes, to the destination node. As is evident, many routes may exist between the source and the destination node, and the discovery of the most optimal route may be difficult for any network to achieve.

Obviously, for any ad-hoc communication system to function properly, a method and apparatus for operating the ad-hoc communication system must exist that discovers efficient routes between source and destination nodes. Therefore a need exists for a method and apparatus for operating an ad-hoc communication system that lends itself to easy route discovery, while assuring that any route discovered is an efficient route between devices.

Brief Description of the Drawings

FIG. 1 is a block diagram of a prior-art ad-hoc hierarchy tree network.

FIG. 2 illustrates network formation having "shortcuts".

FIG. 3 illustrates a prior-art network.

FIG. 4 is a block diagram of a node of FIG. 1.

FIG. 5 is a flow chart sowing the operation of a node of FIG. 1 when forming an ad-hoc network.

FIG. 6 is a flow chart showing operation of a node of FIG. 1 during route discovery

Detailed Description of the Drawings

To address the above-mentioned need method and apparatus for operating an ad-hoc communication system is provided herein. More particularly, an ad-hoc communication system is established where each node within the system may have multiple children nodes and multiple parent nodes. An address scheme is proposed where each node may comprise more than one address, each address being based upon one of the parent's address. Because each node may have more than one parent, a bridge, or shortcut may exist between network branches. This is in contrast to prior-art techniques for routing in which a single node may have only a single parent.

The present invention encompasses a method for operating a node within an ad-hoc communication system. The method comprising the steps of associating with a first parent node, receiving a first address from the first parent node, associating with a second parent node, and receiving a second address from the second parent node.

The present invention additionally encompasses a method for finding a best route to a destination node. The method comprises the steps of determining a number of hops to the destination node through a first parent node, determining a number of hops to the destination node through a second parent node, determining a number of hops to the destination node through a first child node, and determining a number of hops to the destination node through a second child node. A node having a shortest number of hops to the destination node is determined and route information is provided for routing data through the node having the shortest number of hops.

The present invention additionally encompasses a node existing within an ad-hoc communication system. The node comprises logic circuitry associating the node with a first and a second parent node and a receiver, receiving a first address from the first parent node and receiving a second address from the second parent node.

The present invention additionally encompasses an apparatus for determining a best route to a destination node. The apparatus comprising logic circuitry for determining a number of hops to the destination node through a first parent node, determining a number of hops to the destination node through a second parent node,

determining a number of hops to the destination node through a first child node, determining a number of hops to the destination node through a second child node, and determining a node having a shortest number of hops to the destination node. The apparatus additionally encompasses a transmitter for providing route information in order to route data through the node having the shortest number of hops.

Turning now to the drawings, wherein like numerals designate like components, FIG. 1 is a block diagram of network 100. As is evident, network 100 comprises a plurality of network nodes 101-107 in communication with each other. Network 100 preferably utilizes a modified neuRFon™ system protocol as described in US Patent Application Serial No. **09/803259**. As one of ordinary skill in the art will recognize, within the neuRFon™ system protocol single root node 104 serves as a group leader (GL), other nodes 103, 105 (referred to as "children nodes") form a direct link to group leader 104, with up to a maximum number (C_m) of children nodes under any given node. In a similar manner, each child node 101-103 and 105-107 can serve as its own group leader and have up to C_m child nodes themselves. Thus, in FIG. 1, where $C_m = 2$, group leader 104 has two child nodes 103 and 105. In a similar manner, child nodes 103 and 105 serve as their own group leader and each have two child nodes in direct communication with them. Particularly, child node 103 has nodes 101 and 102 in direct communication with it, while child node 105 has nodes 106 and 107 in direct communication with it.

Each node 101-107 within communication system 100 is assigned an address that indicates how many hops in communication the node is from root node 104. More generally, every node (except the root node) is assigned a "first" and a "last" (second) name, with the "last" name being equal to a parent's whole name (i.e., the parents first and last name). Thus, for example, root node 104 is at level 1 and assigned a whole name of "1". Nodes 103 and 105, being the first two nodes under node 104 are assigned a first name of "1" and "2" respectively, with each having a last name of 1. Thus, node 103 is named "11" with node 105 being named "21". In a similar manner, with nodes 101 and 102 being the first and the second nodes under node 103, node 101 is given a first name of "1" and node 102 is given a first name of "2". Each node has a last name of "11" (the parent node's whole name). Thus, node 101 is named "111" while node 102 is named "211".

Extrapolating this nomenclature to a system where $C_m = 4$ and a group leader's whole name is "31", then:

- If there is only one group member, then the name of it is"131".
- If there are two group members, then the names of the group members are "131", and "231".
- If there are three group members, then the names of the group members are "131", "231" and "331"
- If there are four group members, then the names of the group members are "131", "231", "331", and "431".

Therefore, a particular node within communication system 100 will always obtain an address *from* its parent node, and will always provide addresses *to* its child nodes. It should be noted that although FIG. 1 shows each child having a single parent, in the preferred embodiment of the present invention, each child may have up to C_p parents, with C_p preferably being equal to 2. If a device has more than one parent, it will receive one name from each of its parents. Thus it will have more than one name (address). For example, a device having one name "131" that is assigned by GL (parent) "31" may later receive another name "2411" that is assigned by GL (parent) "411". This is illustrated in FIG. 2.

As is evident, node 201 has two parents, namely a first parent with address "31" and a second parent with address "411". Thus, node 201 has two addresses, namely "131" and "2411". Because of this, bridge 202, or shortcut 202 is formed between branch "11" and branch "31". As mentioned, a device can be recruited to join C_p different parents, with C_p preferably being a small number, such as 2, to avoid too many shortcuts. As illustrated in FIG. 3, this is in stark contrast to prior-art techniques for routing in which a single node may have only a single parent. More particularly, FIG. 3 shows a prior-art Parent/Child tree structure for C_m=4. As is evident in FIG. 3, each node may have up to C_m=4 children, however, only a single parent is allowed, and thus no shortcuts exist between network branches.

Network Formation

During network formation, all the nodes are turned on waiting for a *FORMATION* message. The *FORMATION* message typically occurs after all nodes are deployed by manually triggering a specific device to send out the message. Once the *FORMATION* message has been sent out, the network formation process starts with the specific device acting as an initial group leader or root, sending out the message, and being assigned an address of "1".

As part of the network formation, all the nodes within communication range of the root node (node 1) will receive the *FORMATION* message. Once a neighbor node receives the *FORMATION* message, an acknowledgment is sent back to the root node. Based on the following rules, the group leader selects its group members:

- The group leader determines a RSSI (Receiving Signal Strength Information) and/or Link-Quality for potential child nodes, and only nodes whose RSSI (Receiving Signal Strength Information) and/or Link-Quality are above a threshold are qualified as group members;
- If there are more than C_m nodes whose RSSI is greater than the threshold, then only the C_m nodes with the strongest RSSI are selected as group members. Thus, no more than a predetermined number (i.e., C_m) of child nodes can be associated with any node.

Once a device is selected as a group member, a name (or logical address) will be assigned to it by the group leader. From now on, the node qualifies to be a new group leader and it will start to recruit its own children. If a node does not receive a name from the group leader, it will continually listen and wait for FORMATION message until a predefined time-out period.

As discussed, once the node receives a name (address), it then starts to recruit its own members/children forming a spanning-tree. The recruiting process is the similar to the process followed by the root node, but has slightly different rules, namely:

- Only nodes whose RSSI is gr eater than a threshold are qualified as group members;
- A node may recruit nodes that have already been recruited by another node, however, in order to avoid too many unnecessary short-cut, the direct relatives

of the recruiter are not qualified as group members. As an example, a node's children cannot have the nodes parent as their parent;

- A node that has more than C_p parents is not qualified as group member;
- In order to avoid too many short-cuts within one group, and to effectively expand the network, a group leader shall not have more than C_c ($1 \leq C_c < C_m$) members that have been recruited by other nodes;
- If there are more than C_m nodes that satisfy the above rules, then nodes with the highest RSSI are chosen as children.

Network Routing

Routing within network 100 is based on a concept of address prefix matching. Suffix matching is used to pass packets in an ascending order. The distance between two nodes is defined as the number of hops that are needed to navigate from one node to the other node. The distance or number of hops between two nodes can easily be determined as follows: Assume L_a is the length of the name of the node-a, L_b is the length of the name of node-b, L is the length of the shared last name by the two nodes. Then the distance D between node-a and node-b can be written as: $D = (L_a - L) + (L_b - L)$. For example: To calculate the distance between node "*321*" and node "*44221*". First, determine $L_a = 3$, and $L_b = 5$. Then, $L = 2$. Finally, the distance can be calculated as $D = (L_a - L) + (L_b - L) = (3-2) + (5-2) = 4$.

For routing between a source node and a destination node, the goal is to find a shortest route for sending the message from the source node to the destination node having a least amount of intervening nodes, i.e., one that minimizes D. The following algorithm accomplishes this task:

- Step 1: Begin with the source node and define the source node as the *current node*.
- Step 2: Calculate the distance between the *current node* and the *destination node*. If the distance is zero, then stop. Otherwise, go to step 3.
- Step 3: Find all the children and parents of the current node and calculate the distances between them and the destination node.

- Step 4: Identify a child or parent that if utilized in routing, yields the minimum distance.
- Step 5: Add the identified node the routing table. (Note: If there is more than one node with the same minimum distance, then pick the node that has the strongest RSSI).
- Step 6: Set the identified node as current node, and go to Step 2.

It should be noted that since we may have more than one address for any node, more than one distance between two nodes may be obtained. For these situations the address is used, that if utilized, has the shortest distance to the destination node.

Network Operation

FIG. 4 is a high-level block diagram of node 400 in accordance with the preferred embodiment of the present invention. In the preferred embodiment of the present invention all nodes within ad-hoc communication system 100 contain the elements shown in node 400. As shown, node 400 comprises transmit circuitry 401, receive circuitry 402, and logic circuitry 403. Logic circuitry 403 preferably comprises a microprocessor controller, such as, but not limited to a Motorola PowerPC microprocessor. In the preferred embodiment of the present invention logic circuitry 403 serves as means for controlling node 400, and as means for analyzing message content to determine any actions needed. Additionally receive and transmit circuitry 401 and 402 are common circuitry known in the art for communication utilizing a well known communication protocol, and serve as means for transmitting and receiving messages. For example, for nodes within communication system 100, receiver 402 and transmitter 401 are well known neuRFon™ transmitters that utilize a modified neuRFon™ communication system protocol. Other possible transmitters and receivers include, but are not limited to transceivers utilizing Bluetooth, IEEE 802.11, or HyperLAN protocols.

FIG. 5 is a flow chart showing operation of node 400 when forming an ad-hoc network. The logic flow begins at step 501 where receiver 402 receives a *FORMATION* message from a first node. At step 503 the *FORMATION* message is passed to microprocessor 403 where microprocessor 403 determines if it can join the first node,

being a child of the first node. As discussed above, any particular node can have no more than a predetermined number (i.e., C_p) parents. If, at step 503, it is determined at node 400 can join the first node as a child, the logic flow continues to step 505, otherwise the logic flow ends at step 507.

At step 505 transmitters 401 transmits a "join" message to the first node to join as a child of the first node. The logic flow continues to step 509 where microprocessor 403 determines if a request to join the first node was received. As discussed above, there may be situations where even though a node has asked to join another as its child, the parent node may decline. More particularly,

- A parent will only accept nodes whose RSSI are greater than a threshold;
- A parent may not recruit the siblings (siblings: the nodes share parents), parents;
- A parent may not recruit node that already has more than C_p parents;
- A parent may not recruit more than C_c nodes that have already been recruited by another node.
- A parent may not recruit more than C_m nodes.

If, at step 509 it is determined that a request to join the first node was received, then at step 511 logic circuitry 403 associates the node with the first node as a child, receiving an address from the first node, otherwise the logic flow ends at step 507. At step 513, the child node send out a *FORMATION* message of its own, and recruits up to C_m child nodes.

It should be noted that the above procedure takes place every time a node hears a formation message. Thus, logic circuitry 403 may instruct node 400 to join up to C_p parents, receiving an address from each parent. Thus, any given node may associate with a first parent, receiving a first address, and then associate with at least a second parent, receiving a second address. As discussed above, a first and a second address portion are received from each parent. Additionally, during the recruiting of the child nodes, differing addresses will be given out to each child. Thus, the node receives a first and a second address from its parents, and may give up to C_m additional addresses to its children, with each address given to its children having a second portion equal to the nodes first or second address.

FIG. 6 is a flow chart showing operation of single node 400 during route discovery. In the following scenario node 400 may be the source, destination, or any intervening node that has been requested to find the best route between a source and a destination node. Additionally, the logic flow in FIG. 6 assumes that the network is established with each node having up to C_m child nodes and C_p parent nodes. The logic flow begins at step 601 where a node 400 receives a request (via receiver 402) to find a best route from a source node to a destination node. This request comprises the source address and the destination address. At step 603 node 400 determines if it is the destination node. If node 400 determines that it is the destination node, then the logic flow continues to step 611, otherwise the logic flow continues to step 605. At step 605 microprocessor 403 determines distances (i.e., number of hops) to the destination node through each parent and child node associated with node 400. In particular, the number of hops to the destination node from each parent and child node associated with node 400 is calculated by utilizing $D = (L_d - L) + (L_{p(c)} - L)$. Where $L_{p(c)}$ is the length of the parent or child name, L is the length of the shared last name by the destination node and the parent/child, and L_d is the length of the destination's name. Since each parent and child has different addresses, and some of them may have more than 1 name, their names will have different length, also the shared last name by the destination and the parent/child may be different.

Continuing, at step 607 a parent or child node having the shortest distance to the destination node is determined and at step 609 the request for route discovery is then passed to the node identified in step 607, and the above procedures take place in the identified node. At step 611 route information is provided back to the source node via transmitter 401 so that it can be updated with route information. In particular, routing information for routing data through the node having the shortest number of hops is provided. At step 613 database 405 is updated with routing information. In particular, database 405 is updated so that node 400 knows where to route data received from the source node, destined to the destination node.

The above procedure allows for a node to discover the shortest number of hops to the destination node by determining a number of hops through each parent and each child. Because multiple parents may exist for each node, a bridge or shortcut may exist for the route, greatly reducing the number of hops needed for relaying the message between the source and destination nodes.

While the invention has been particularly shown and described with reference to a particular embodiment, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention. For example, by carefully choose the transmitting power, one can control the average number of neighbors of the GL, e.g., 10 neighbors. Additionally, each neighbor node may randomly generate an identification ranged from 1 to 255 to distinguish it from other nodes, and each node sends back an ACK at a different time slot. If more than two nodes happened to pick the same number, they are required to repeat the process. It is intended that such changes come within the scope of the following claims.

Claims

1. A method for operating a node within an ad-hoc communication system, the method comprising the steps of:

associating with a first parent node;
receiving a first address from the first parent node;
associating with a second parent node; and
receiving a second address from the second parent node.

2. The method of claim 1 wherein the step of receiving the first address from the first parent node comprises the step of receiving a first and a second address portion, wherein the second address portion is equal to a first and a second address portion assigned to the first parent node.

3. The method of claim 2 wherein the step of receiving the second address from the second parent node comprises the step of receiving a third and a fourth address portion, wherein the fourth address portion is equal to a first and a second address portions assigned to the second parent node.

4. The method of claim 1 wherein the step of associating with the first parent node comprises the step of associating with the first parent node if no more than a predetermined number of parents exist for the node and the step of associating with the second parent node comprises the step of associating with the second parent node if no more than the predetermined number of parents exist for the node.

5. The method of claim 1 further comprising the steps of:

serving as a parent node to a first child node;
assigning the first child node a third address;
serving as a parent node to a second child node; and
assigning the second child node a fourth address.

******

6. A node existing within an ad-hoc communication system, the node comprising:

logic circuitry associating the node with a first and a second parent node; and

a receiver, receiving a first address from the first parent node and receiving a second address from the second parent node.

7. The node of claim 6 wherein the first address comprises a first and a second address portion, wherein the second address portion is equal to the first address.

8. The node of claim 6 wherein the second address comprises a third and a fourth address portion, wherein the fourth address portion is equal to second address.

9. The node of claim 6 wherein the logic circuitry associates the node with the first parent node if no more than a predetermined number of parents exist for the node and wherein the logic circuitry associates the node with the second parent node if no more than the predetermined number of parents exist for the node.

10. The node of claim 6 wherein the logic circuitry additionally serves as a parent node to a first child node, and serves as a parent node to a second child node.

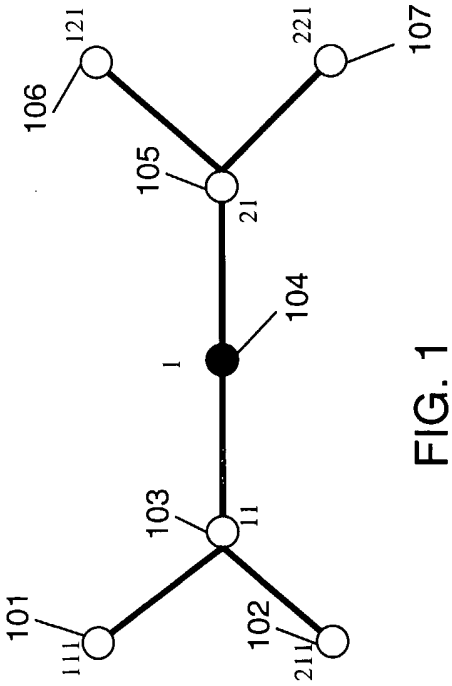


FIG. 1
100

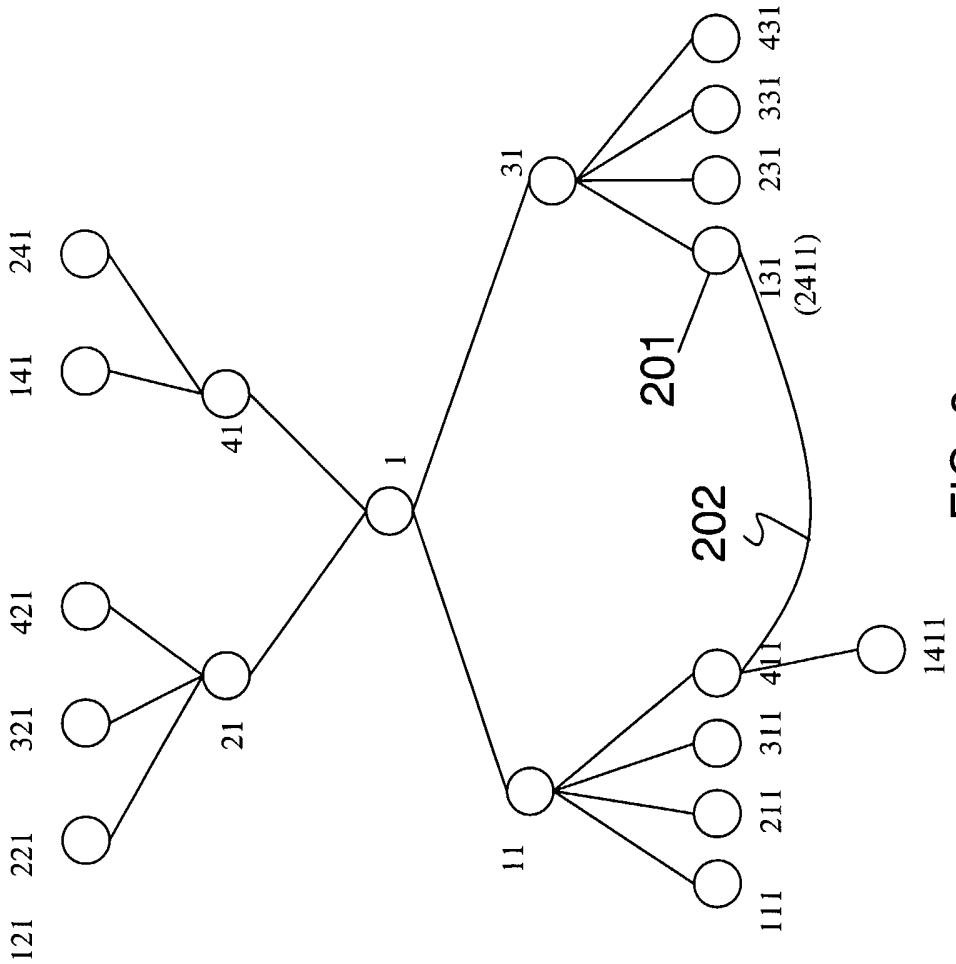
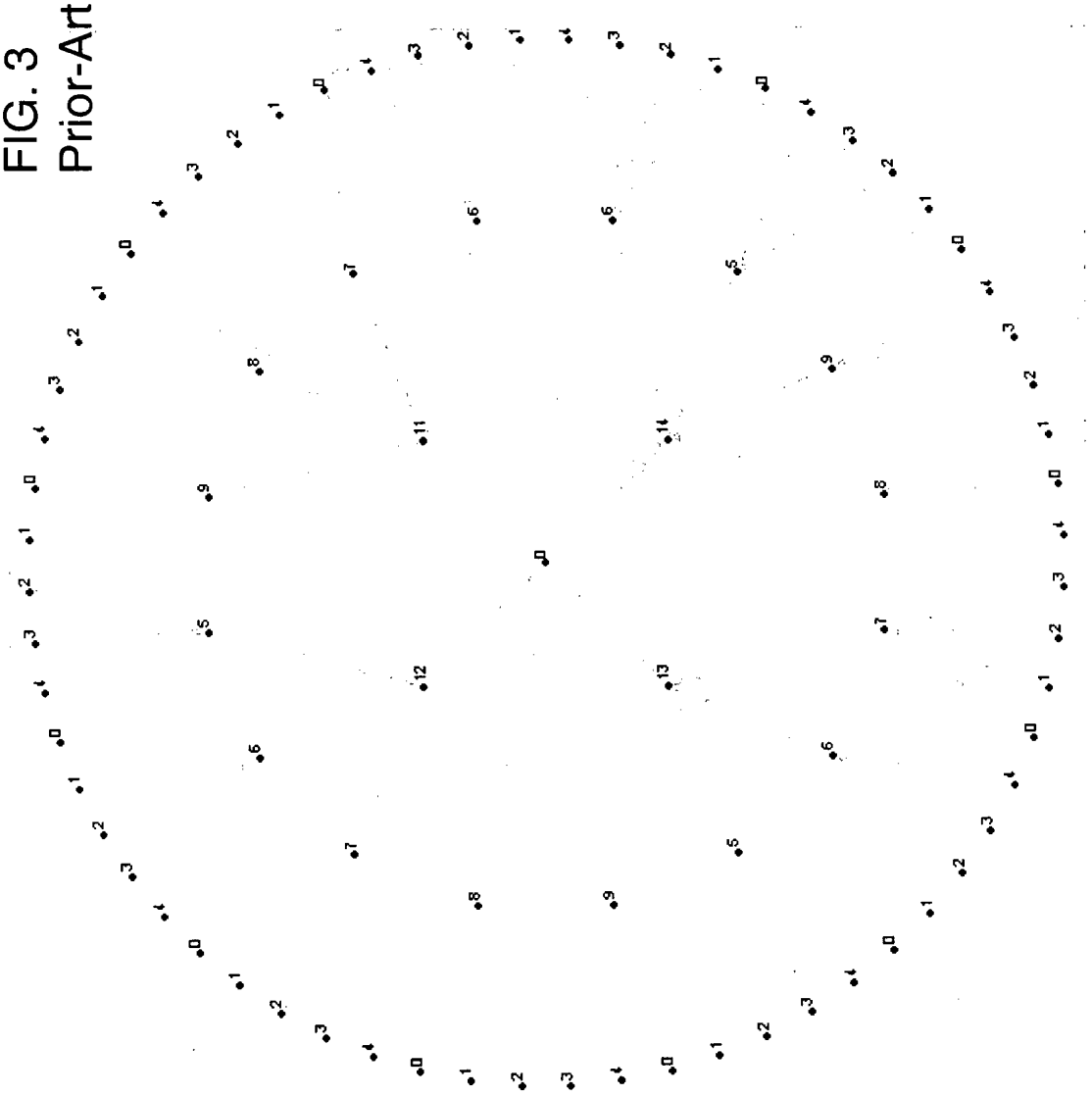

121
221
321
421
141
241
21
41
1
11
31
111
211
311
411
1411
131
(2411)
231
331
431
202
201

FIG. 2

FIG. 3
Prior-Art



Parent/Child tree structure for Cm=4

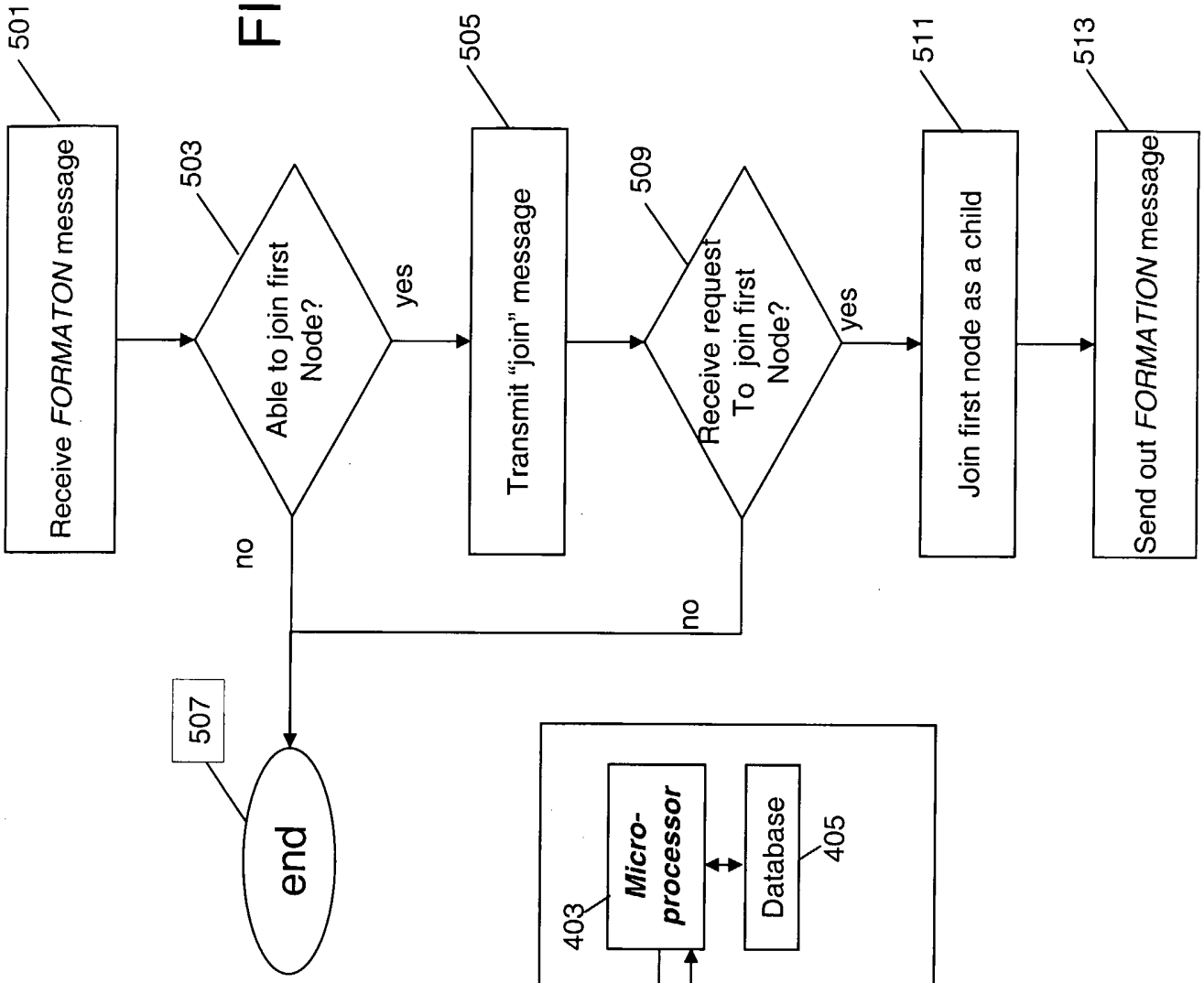

501
Receive FORMATON message
503
Able to join first Node?
yes
no
505
Transmit "join" message
509
Receive request To join first Node?
yes
no
511
Join first node as a child
513
Send out FORMATION message
507
end

FIG. 5

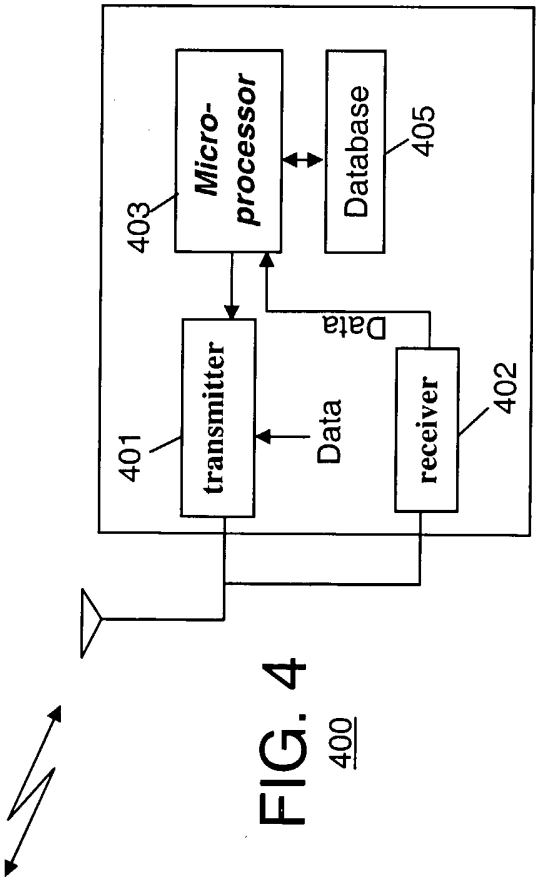

403
Micro-processor
Database
405
Data
401
transmitter
Data
receiver
402
400

FIG. 4

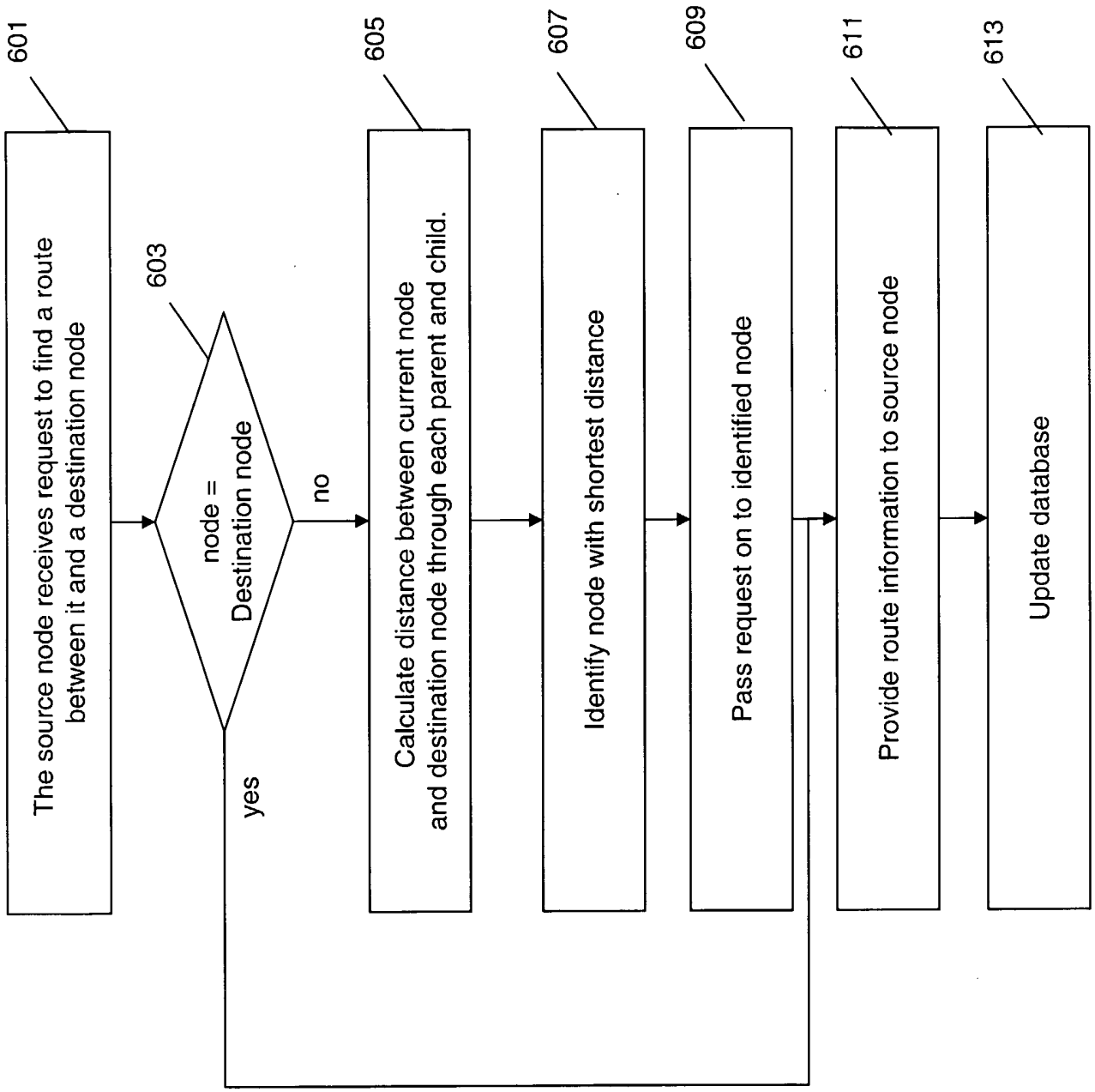

601
The source node receives request to find a route between it and a destination node
603
node = Destination node
yes
no
605
Calculate distance between current node and destination node through each parent and child.
607
Identify node with shortest distance
609
Pass request on to identified node
611
Provide route information to source node
613
Update database

FIG. 6