Application Report
SPRABT7 – August 2013

Cost Down Metro Network

Fan Zhang, Shikun Yu MCU

ABSTRACT

Wireless solutions for utility meter reading are becoming increasingly popular. With wireless communication, there is no need to lay wires in buildings to collect remote meter reading, and it easy to maintain. A mesh network, which can be applied easily with very low resource use, quick response, and flexible frequency options, has great potential. This application report describes a cost down metro network that is relatively simple to implement and uses less resources than well-known alternatives such as Zigbee®. The ability to actively join and report data periodically makes this network suitable for many sensing network application fields.

It is expected that the master software can be absorbed into the ultra-low power microcontroller unit (MCU) such as MSP430F5438A from TI, and slave software can be absorbed into main MCU of an electricity meter device such as the MSP430FE425 from TI.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Key requirements for metering-based mesh network are as follows:

- There are niche industrial markets for expensive AMR systems, but the cost must be extremely low to achieve a mass market.
- Obviously, AMR is always seen as a way to read consumption without visiting every meter. In markets where tampering is a major issue, AMR is also seen as a means to more closely monitor the status of a meter.
- Manually tuning a radio network is not a realistic option; a successful solution must be wireless. A meter fitted in nearly any practical location in a building must work and be accessible. A meter could be high in a tower or underground in a cellar (although cellars are not common in Asia).
- The 2.4 GHz ISM band has lots of competing users, poor range, and is easily shielded by metal objects. A <1 GHz (for example, 433 MHz) solution suits AMR much better. Being able to easily change the band is important at <1 GHz, as the available bands and channels vary widely.
- A mesh network structure is important. A simple star topology, with low-power radios, would not offer an attractive reach from any concentrator or reader node.
- Processing must be sufficiently lightweight; that is, it can be handled on an existing MCU that implements the main functions of the product – for example, metrology.
- Memory requirements (both RAM and ROM) must be compatible with putting the processing on an existing MCU that implements metrology. This implies that only minimal state information can be held resident on the nodes, and that message buffers must be of moderate size.
- Authentication is very important. In wire-line AMR systems, people do not always treat this fact seriously. In wireless AMR, everyone is highly conscious of the need for effective authentication. The data is essentially financial information, so the need for security should be considered critical. Encryption of the data is desirable, but solid authentication is a necessary feature.
- Jamming resistance is a valuable feature. Jamming may be accidental, in which case it probably comes and goes. When jamming is not present, the concentrator can read meters, thus jamming is merely a nuisance. Jamming may be intentional, to prevent monitoring of the meters. Where tampering has gained the status of a sport, such jamming could be very troublesome. Mesh schemes which can route around jammers are attractive. A frequency agile system seems attractive. However, a sufficient number of radio channels may not be allocated for AMR at <1 GHz to make this useful.
- Some people look for the ability to approach an area, and quickly extract the data from all meters in that area with a portable reader. This functionality can be accomplished more effectively by gathering all data at a concentrator, and extracting data for all subscribers quickly from that one node. Electricity substation always provides a suitably located site, under the control of the utility, where such a concentrator can be sited. Such a concentrator could economically connect to a wide area radio network, such as GPRS or UMTS, for continuous use and to monitor tampering. The concentrator can methodically map the mesh of nodes, and can frequently check for tampering. If all mesh state information is stored at this concentrator, very little state needs to be held in each meter.
- Speed is not of high importance. If the radio network is always running, a concentrator should always have reasonably up-to-date information about each node, even if the data rate is quite low.
- The ability to run at extremely low average power, from a small battery, is attractive. This feature allows the system to monitor gas or water meters. This feature is also valuable in electricity meters, to continue monitoring them during blackouts and tamper conditions.

There are two kinds of devices in the network, a master node and a slave node.

A master node is used as data collector, and there is only on master node in a network. A master node establishes, manages, and maintains the network. A master node stores connection link information of the whole network for route wireless packets.

Typically, a slave node is connected to one or more sensors. A slave node gets desired information from associated sensors and sends the sensing data through the network to the master node. The number of slave nodes depends on the size of the network. Nothing for routing is stored permanently in slave nodes. The only information stored in a slave node is the address of its father-node, which is used when actively reporting time pressing data, such as alarm.

This cost down metro network has the following features:

- Self organization, self maintenance
- All operations are done by embedded devices; a PC is not required.
- Mesh topology
- Multihop communication to enlarge coverage
- Multiroute to avoid the effect of the failure of a single node
- Slave node can actively request to join into network without waiting for scan from master node.
- Slave node can actively report emergency data without waiting for data request from master node.
- Combination of on-demand query and periodical report
- Few resource requirement for slave node

2. Principle of Operation

Because this network supports both on-demand query and periodical report, to make them coexist, communications in the network are divided into two parts, the downlink part and uplink part.

The downlink part operates similarly to Simple Wireless Mesh Network [1], in which all communications are initiated by the master node. The master node maps all slave nodes by probing outward step by step, and finally forms the mesh topology. All connectivity information and routing data are stored in the master node. Slave nodes do not store any connectivity information for routing in downlink communications. Master node, which completely controls the network, requests actions of the slave nodes and expects responses from them.

The uplink part is primarily used to support active communication initiated by slave nodes. A slave node can actively request to join the network without waiting for the next period when its neighbor nodes are called to discover new nodes. This decreases network buildup time. Also, slave nodes can actively initiate data report to the master. This is especially useful for the collection of emergency information, such as alarm and other time pressing data. A slave node needs to store only the address of its father-node, which is more near to master node than it in logical links and is used to forward active data report to master.

3. Frame Format

3.1 Physical Layer Packet Format

CC1101 low-power sub-1 GHz RF transceiver from TI is used to implement wireless transmission. The format of the data packet can be configured and consists of the following items (see Figure 1).

- Preamble
- Synchronization word
- Optional length byte
- Optional address byte
- Payload
- Optional 2-byte CRC

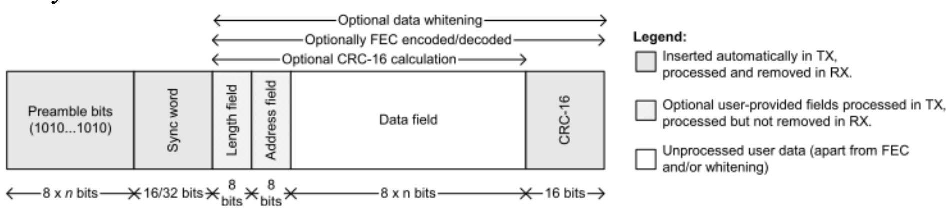

Figure 1. Physical Frame Format

With proper configuration, most operations are done automatically by CC1101.

3.2 Application Layer Frame Format

Figure 2 shows frame format of data payload, which is filled into Data field of Figure 1.

Length	To_ID	From_ID	Type	Reserved	HopNum	Route	Data

Figure 2. Frame Format

Table 1 describes the meaning of each field in Figure 2.

Table 1. Frame Field Description

Field	Size	Description
Length	1 byte	Length of the frame.
To_ID	1 byte[*]	Address of the desired receiver of the frame. In multihops communication, the To_ID field indicates the next-hop, not the final destination.
From_ID	1 byte[*]	Address of the sender. In multihops communication, the From_ID field indicates the sender of current transfer, not the original sender.
Type	1 byte	Type of the frame, such as Discovery Request, Data Response, and so forth.
Reserved[**]	1 byte	Reserved for future extension.
HopNum	1 byte	Number of hops in the route of the frame.
Route	n byte	Address list of the route, from the original sender to the final destination. The size of this field is (HopNum + 1)[*].
Data	m byte	Payload of the frame. The size of this field depends on type of the frame.

[*] Currently, device address is 1 byte. It may be extend in future version.
[**] This field may be used as Priority control in future version.

4. Frame Type

Devices in network use various frames to achieve different purposes. Table 2 describes all supported frames.

Table 2. Frame Type Description

Link	Type	Description
DownLink	Discovery Request	Initiate by master node. Request a specified slave node to discover for new communicating link. It contains route list from master node to that specified slave node.
	Discovery Broadcast	Initiate by master or slave node. Discovery Broadcast frame is similar to Discovery Request frame. The only difference is that To_ID field in Discovery Broadcast frame is set to 0 to indicate it is a broadcast frame. When a slave node receives a Discovery Request, it sends out a Discovery Broadcast.
	Discovery Response	Initiate by new slave node (include node has not joined into network and node has already joined; to node which has already joined, new logical communicating link is established in discovery process). Inform master node that new logically communicating link has been found. It uses the reversed route from Discovery Request frame.
	Discovery NonResponse	Initiate by slave node. Inform master node that no new communicating link is found. It uses the reversed route from Discovery Request frame.
	Data Request	Initiate by master node. Request data from specified slave node. It contains route list from master node to that specified slave node.
	Data Response	Initiate by slave node. Respond to master node with desired data. It uses the reverse route from Data Request frame.
	Call Request	Initiate by master node. Ask for response from specified slave node to check whether it works normally. It contains route list from master node to that specified slave node.
	Call Response	Initiate by slave node. Respond to master node to inform it works normally. It uses the reverse route from Call Request frame.
UpLink	Join Request	Initiate by new slave node. It sets To_ID to 0 to broadcast this request. This frame is used to find connection links with its neighbor nodes. This frame is single hop, and should not be forwarded.
	Join Response	Initiate by master or slave node. Respond to the requesting node to confirm the connection link.
	Join Report	Initiate by slave node. Inform master node about the new established connection link. There is no hop-list for routing in Join Report frame because slave node does not maintain routing information for the whole network. In Join Report process, each node just sends or relays this frame to its father-node, until the frame reaches master node.
	Join Confirmation	Initiate by master node. Inform the slave node, who is the original sender of the previous Join Report frame, that Join Report frame has been received successfully. It contains route list from master node to that specified slave node.
	Data Report	Initiate by slave node. It is used by slave node to actively send data to master node. There is no hop-list for routing in Data Report frame because slave node does not maintain routing information for the whole network. In Data Report process, each node just sends or relays this frame to its father-node, until the frame reaches master node.
	Data Confirmation	Initiate by master node. Inform the slave node, who is the original sender of the previous Data Report frame, that Data Report frame has been received successfully. It contains route list from master node to that specified slave node.

5. Operation Description

Five main kinds of processes implement the network protocol:

- Discovery
- Data request
- Maintenance
- Active Join
- Active Data Report

The following sections describe each process in detail.

5.1 Discovery

The Discovery process is used to form the mesh topology in network buildup time, and is also periodically used to update connection links as network topology may change during run time.

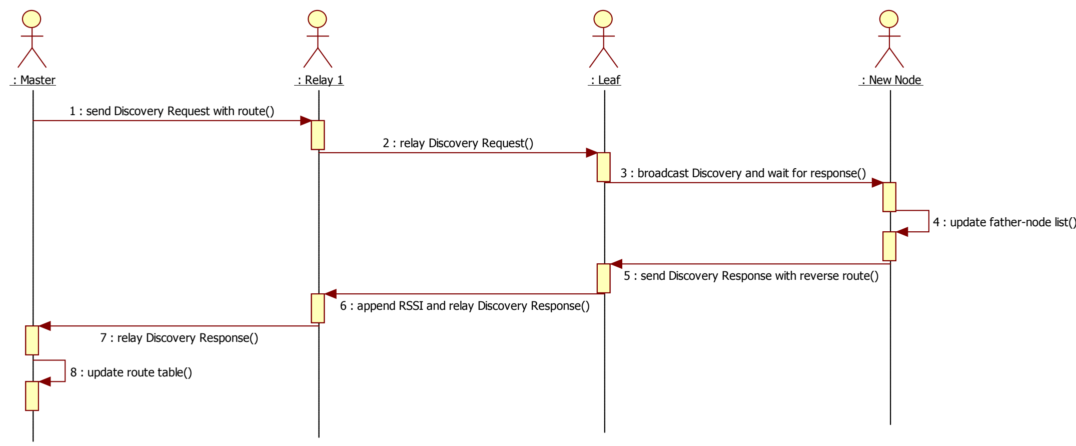

Figure 3. Discovery With Response, Sequence Chart

Figure 3 shows a typical sequence chart of a successful Discovery process. In this sequence chart, master node requests leaf node to scan for new node. Relay 1 node is an intermediate node in the forwarding route.

Master node sends out a Discovery Request frame, which contains a hop-list as source route from the master node to a specified slave node (leaf node in Figure 3). All neighbor nodes in range can receive this request frame, but only the specified next-hop node in the next-list (relay 1 node in Figure 3) forwards this request frame. This avoids broadcast storm involved by simple flooding forward. All the intermediate nodes operate in the similar way, and finally the Discovery Request frame reached the desired final destination node (leaf node in Figure 3).

Then, the final destination node sends out a Discovery Broadcast frame to scan for new nodes (or new connection links). If any, the new node (or node with a new connection link) responds by sending a Discovery Response frame back to master node. It uses the reverse route from the received Discovery Broadcast frame. Finally, master node receives this Discovery Response and stores new connection link information. The new established connection link may be used in source route in subsequent request process. Meanwhile, the new node marks the node, which sends out that Discovery Broadcast frame, as its father-node.

After sending out the Discovery Broadcast frame, that slave node (leaf node in Figure 4) waits for the response. If timeout and it fails to get a Discovery Response frame, it sends a Discovery NonResponse back to the master. This nonresponse frame uses the reverse route from the received Discovery Request frame, as shown in Figure 4.

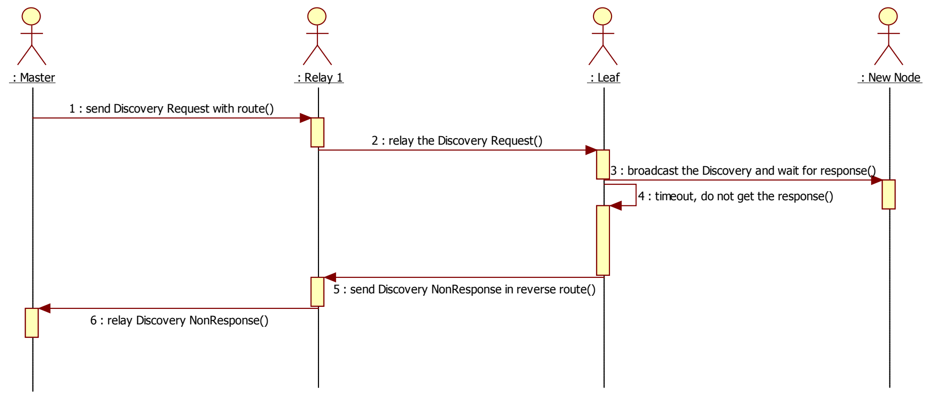

Figure 4. Discovery With NonResponse, Sequence Chart

After sending out the Discovery Request frame, master node waits for the response. If timeout and it fails to get any response (Discovery Response or Discovery NonResponse), master node recognizes that currently used source route is invalid, which means at least on connection link in the hop-list is broken.

Then, it should individually call each node in this invalid source route, and determine the broken connection link, update connection links maintained by master, and search for a new source route based on the updated connection links. Figure 5 shows this process.

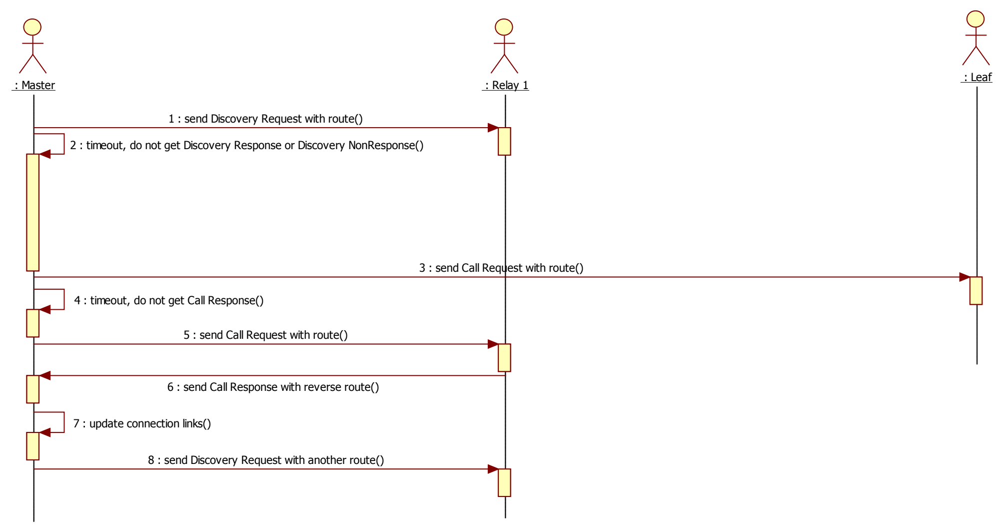

Figure 5. Discovery Failure, Maintenance, Sequence Chart

5.2 Data Request

The Data Request process is used to query desired data from specified slave node.

Figure 6 shows a typical sequence chart of a successful Data Request process. In this sequence chart, master node requests data from leaf node. Relay 1 node is the intermediate node in the forwarding route.

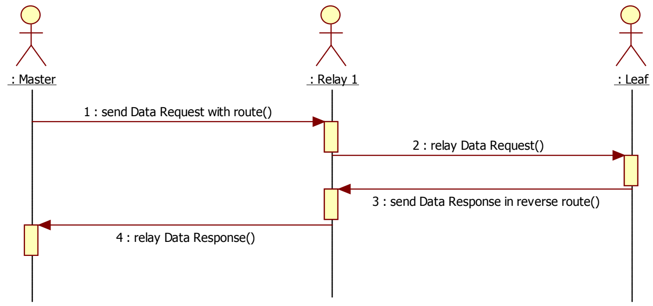

Figure 6. Data Request With Response, Sequence Chart

After sending out the Data Request frame, master node waits for the response. If timeout and it fails to get the response, master node recognizes that the currently used source route is invalid.

Then, it should individually call each node in this invalid source route, and determine the broken connection link, update connection links maintained by master, and search for a new source route based on the updated connection links. Figure 7 shows this process.

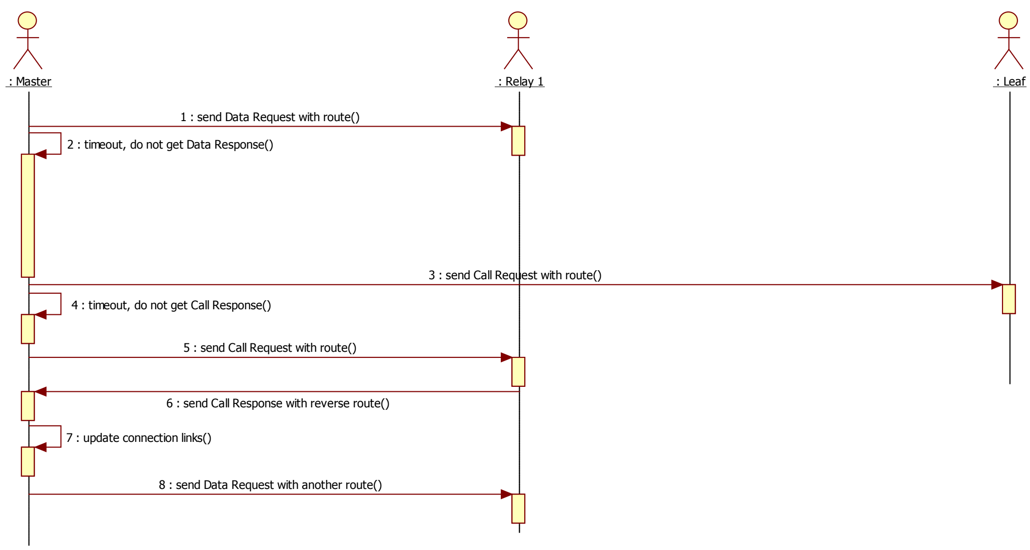

Figure 7. Data Request Failure, Maintenance, Sequence Chart

5.3 Maintenance

When timeout and master fails to get the expected response, the currently used source route is recognized as invalid, which means at least one connection link in the hop-list is broken. Then a Maintenance process starts, using the Call Request and Call Response frame to determine the broken connection link and update link information it maintained. The updated link information is used to search for usable source route. See Figure 5 and Figure 7.

5.4 Active Join

The Active Join process is a beneficial supplement to the Discovery process; Active Join decreases buildup time that has been established in the network and also makes master more sensitive to the topology changes.

Figure 8 shows a typical sequence chart of a successful Active Join process. In this sequence chart, the new node actively requests to join the network through leaf node. Relay 1 node is father-node of leaf node and master node is father-node of relay 1 node.

The Active Join process is initiated by a new node (which has no direct or indirect connection link with master node; see new node in Figure 8) by broadcasting a Join Request frame. If master node or any already-joined slave node (which has direct or indirect connection link to master node, see leaf node in Figure 8) receives this broadcasted Join Request frame, it responds by sending back a Join Response frame. After receiving a Join Response frame, the new node recognizes itself as an already-joined node and adds the address of the sender of the Join Response frame (leaf node in Figure 8) as its father-node.

Meanwhile, after sending back Join Response frame, the slave node (leaf node in Figure 8) sends out a Join Report frame to report this newly established connection link. There is no hop-list in the Join Report frame, as slave nodes do not maintain routing information. The To_ID field of the Join Report frame is father-node address of the current slave node. Each intermediate slave node in the forwarding list relays the Join Report frame to its father-node. And finally, the Join Report frame reaches master node. Then the master node sends a Join Confirmation frame to the original sender of that Join Report frame (leaf node in Figure 8) with source route, to inform that Join Report frame is successfully received by master node.

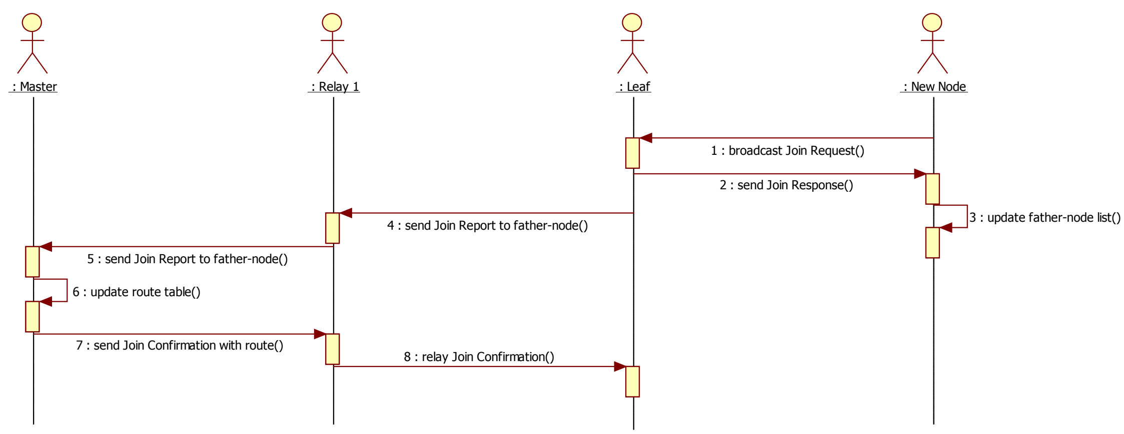

Figure 8. Join With Response, Sequence Chart

5.5 Active Data Report

The Active Data Report process is a beneficial supplement to the Data Request process, to trigger data collection based on the changing state of a sensor. This is especially useful for emergency data collection, such as alarm and time pressing data. It is also very useful in many sensing network application fields

Figure 9 shows a typical sequence chart of a successful Data Report process. In this sequence chart, leaf node actively reports data to a master node. Relay 1 is the father-node of leaf node, and master node is the father-node of relay 1 node.

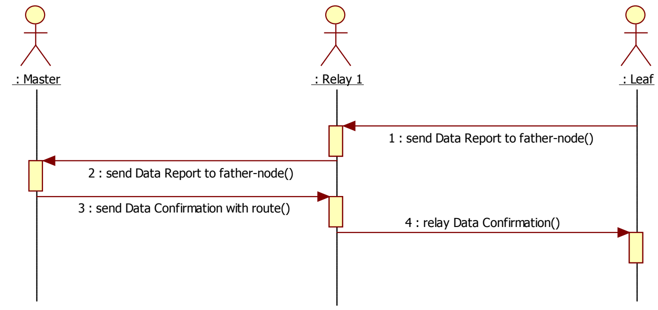

Figure 9. Data Report With Confirmation, Sequence Chart

When a slave node (leaf node in Figure 9) has data to report, it sends a Data Report frame to its father-node (relay 1 node in Figure 9). After receiving a Data Report frame, the father-node (relay 1 node in Figure 9) forwards the frame to its own father-node (master node in Figure 9). And finally, the Data Report frame is forwarded to master node. Then master node returns a Data Confirmation frame to the original sender of the Data Report frame (leaf node in Figure 9) with source route, to indicate that the Data Report frame was successfully received.

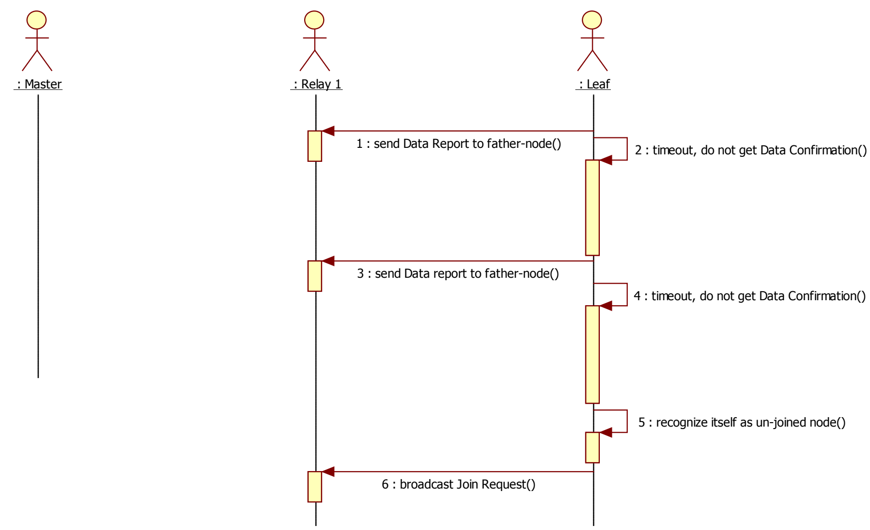

Figure 10. Data Report Without Confirmation, Sequence Chart

After sending out the Data Report frame, the original sender of the Data Report frame (leaf node in Figure 10) waits for the response. If timeout and it fails to get the response, it resends Data Report. If the slave node tries several times and still fails to get any response, it recognizes itself, leaves the network, and should rejoin the network before it can report data. Figure 10 shows this process.

6. Flow Charts

Figure 11 shows general flow chart of master node.

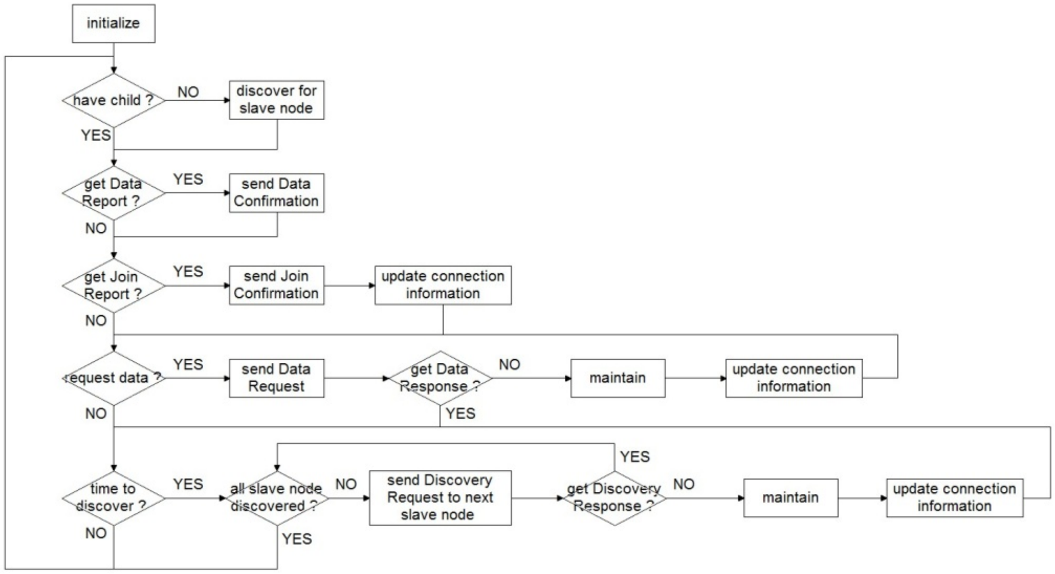

Figure 11. Flow Chart of Master Node

Figure 12 shows general flow chart of slave node.

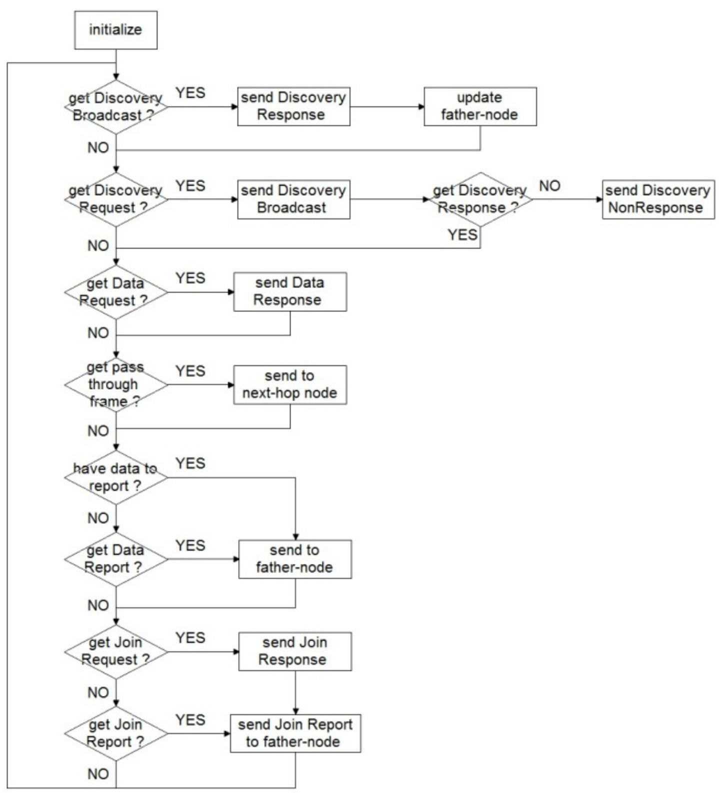

Figure 12. Flow Chart of Slave Node

7. Workflow Example

7.1 Establish Connection Link

7.1.1 Initial State

Figure 13 shows the initial state of all nodes. There are six new nodes, and they are unjoined nodes now.

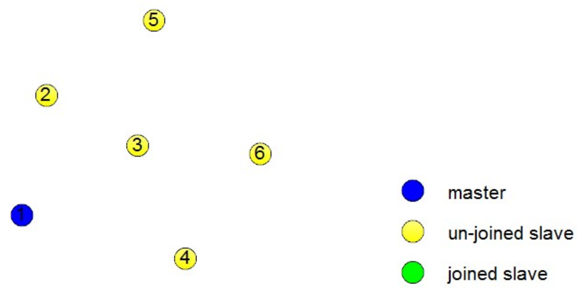

Figure 13. Initial State

7.1.2 Level-1 Nodes Joined

Nodes 2, 3, and 4 are in radio range of master. They join into the network through Active Join process or Discovery process from the master. See Figure 14.

The master receives the following application layer frames:

```
05 00 02 80 00 00                    // 02 broadcast Join Request
05 00 03 80 00 00                    // 03 broadcast Join Request
05 00 04 80 00 00                    // 04 broadcast Join Request
```

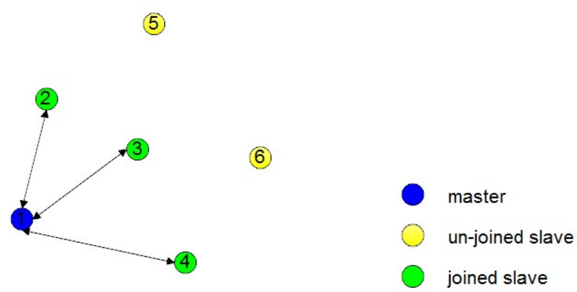

Figure 14. Level-1 Nodes Joined

7.1.3 All Nodes Joined

Nodes 5 and 6 actively join into the network through other already joined slave nodes. See Figure 15.

The master receives the following application layer frames:

```
08 01 02 82 00 00 02 05 EF                    // 02 send Join Report to 01
08 01 03 82 00 00 03 06 C8                    // 03 send Join Report to 01
```

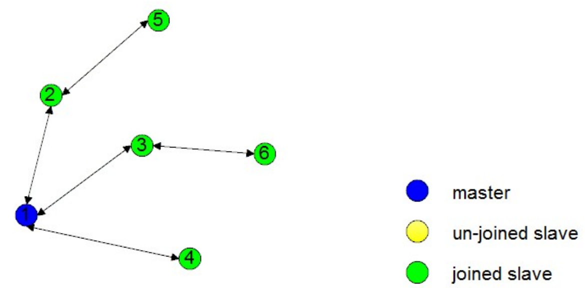

Figure 15. All Nodes Joined

7.2 Form Mesh Topology

7.2.1 Level-1 Nodes Discovered

The master discovers its direct child-node, and then requests each joined slave node to discover for new nodes or new connection links. Figure 16 shows logic topology after all level-1 slave nodes have discovered.

The master receives the following application layer frames:

```
0A 01 02 02 00 02 05 02 01 DC E1              // find connection between 02 and 05
0A 01 02 02 00 02 03 02 01 CA CE              // find connection between 02 and 03
08 01 02 03 00 01 02 01 F7                    // no more connection
0A 01 03 02 00 02 02 03 01 F2 FD              // find connection between 03 and 02
0A 01 03 02 00 02 05 03 01 C2 C9              // find connection between 03 and 05
0A 01 03 02 00 02 04 03 01 EA F0              // find connection between 03 and 04
0A 01 03 02 00 02 06 03 01 CE CE              // find connection between 03 and 06
08 01 03 03 00 01 03 01 F8                    // no more connection
0A 01 04 02 00 02 03 04 01 F5 EB              // find connection between 04 and 03
0A 01 04 02 00 02 06 04 01 02 F9              // find connection between 04 and 06
08 01 04 03 00 01 04 01 ED                    // no more connection
```

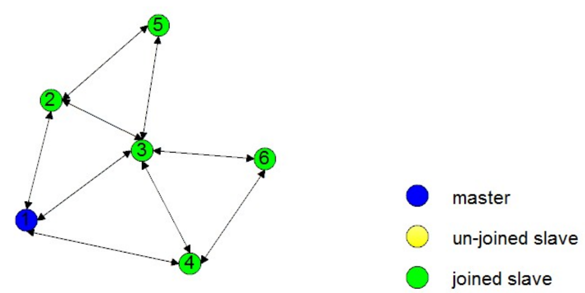

Figure 16. Level-1 Nodes Discovered for New Connections

7.2.2 All Nodes Discovered

Figure 17 shows logic topology after all level-2 slave nodes have been discovered.

The master receives the following application layer frames:

```
0B 01 02 02 00 03 03 05 02 01 C2 C4           // find connection between 05 and 03
09 01 02 03 00 02 05 02 01 C2                 // no more connection
0B 01 03 02 00 03 04 06 03 01 C4 C8           // find connection between 06 and 04
09 01 03 03 00 02 06 03 01 D2                 // no more connection
```

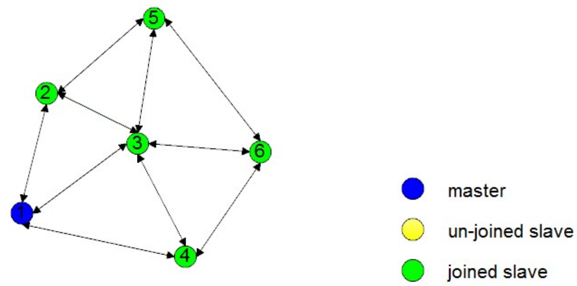

Figure 17. All Nodes Discovered for New Connections

7.3 Active Data Report

As slave nodes are set to actively report data to the master periodically. There is no route list in the Data Report frame; each node just sends or relays the Data Report frame to its own father-node. Application layer frames received by the master follow:

```
0A 01 02 84 00 00 02 22 22 22 22        // 02 send Data Report to 01
0A 01 03 84 00 00 03 33 33 33 33        // 03 send Data Report to 01
0A 01 04 84 00 00 04 44 44 44 44        // 04 send Data Report to 01
0A 01 02 84 00 00 05 55 55 55 55        // 02 relay Data Report to 01
0A 01 03 84 00 00 06 66 66 66 66        // 03 relay Data Report to 01
```

7.4 Periodical Maintenance

The master periodically requests each joined slave node to discover a new node or new connection links. If a slave node has responded Discovery Broadcast from another node, it never responds to Discovery Broadcast from that node again. So, if the real connection relationships do not change, no new connection link is found and all slave nodes respond with Discovery NonResponse.

```
08 01 02 03 00 01 02 01 EF                          // 02 send Discovery NonResponse to 01
08 01 03 03 00 01 03 01 F1                          // 03 send Discovery NonResponse to 01
08 01 04 03 00 01 04 01 F3                          // 04 send Discovery NonResponse to 01
0A 01 02 03 00 02 05 02 01 F7 F6     // 02 relay Discovery NonResponse to 01
0A 01 03 03 00 02 06 03 01 EB F3     // 03 relay Discovery NonResponse to 01
```

7.5 Data Query

Data query is initialized by the master. Source route is contained in a Data Request frame. The final destination node responds this request to the master using reverse route in a Data Request frame.

```
0C 01 02 05 00 01 02 01 22 22 22 22 FE              // 02 send Data Reponse to 01
0C 01 03 05 00 01 03 01 33 33 33 33 0D              // 03 send Data Reponse to 01
0C 01 04 05 00 01 04 01 44 44 44 44 E1              // 03 send Data Reponse to 01
0E 01 02 05 00 02 05 02 01 55 55 55 55 CC CE        // 02 realy Data Reponse to 01
0E 01 03 05 00 02 06 03 01 66 66 66 66 D9 DC        // 03 relay Data Reponse to 01
```

References

[1] *CC430-Based Simple Wireless Mesh Network Mains Switch* SLAA487
CC1101 - Low-Power Sub-1 GHz RF Transceiver SWRS061H
MSP430F543xA, MSP430F541xA Mixed Signal Microcontroller SLAS655B
MSP430F541xA, MSP430F543xA Code Examples SLAC375C

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265