IQMESH, Technology for Wireless Mesh Networks: Implementation Case Studies

Vladimir Sulc
MICRORISC s.r.o.
Jicin, Czech Republic
sulc@microrisc.com

Radek Kuchta | Radimir Vrba
Faculty of Electrical Engineering and Communication
Brno University of Technology
Brno, Czech Republic
kuchtar | vrbar @feec.vutbr.cz

***Abstract* – in this paper, IQMESH, a new networking technology for wireless mesh networks, its basic principles and related routing algorithms are presented. The presented technology was developed especially for applications in the field of buildings automation and telemetry. However, other applications such as smart grids or street lighting can also benefit from straightforward implementations and low system resources requirements. An implementation for IQRF communication platform, described at the end, shows actual system resources requirements in a specific scenario limited by 240 hops and 65 thousand devices.**

Keywords-wireless; mesh; networking; routing; algorithm; IQRF; IQMESH;

I. INTRODUCTION

IQMESH is a networking technology developed for Wireless Mesh Networks (WMN) utilizing packet transmissions. In such networks, messages are sent in smaller parts called packets. A packet has information about a recipient and, in general, the mesh network. This is transmitted from a sender to its recipient through nodes connected in the mesh network. A strategy for sending packets from one node to another is commonly known as routing, and its goal is to deliver packets efficiently and reliably.

A mesh network in which every node has a direct link to another node is a fully connected mesh network. In real WMN only partially connected mesh networks are used, which means that there is no universal direct link between devices. As an illustrative example of such mesh network topology and related routing is to compare it to vehicles traveling between cities. The whole path from the origin to the destination consists of numerous individual roads connecting cities. Searching for the best connection between two cities is similar to mesh networks finding the shortest and most efficient path, from the origin to the destination, between two selected nodes.

A link can be established between any two nodes in a mesh network. In a network consisting of n nodes and one coordinator, the number of possible links will always be lower or equal to N_{MAX} calculated as (1).

$$N_{MAX} = \frac{n\,(n-1)}{2} \qquad (1)$$

Devices in WMN communicate with each other wirelessly, so communication between two devices is usually limited by the communication distance, or so called range limitation of these two devices. Positions of devices in general WMN are not known and the range limitation can depend on many conditions, therefore the routing is usually a great algorithmic challenge to find the best path the packet should travel along. More nodes result in more possible links and consequently to more combinations of links.

Many different routing algorithms are used practically. A flooding, routing based on tables and random routing are just a few basic examples of such algorithms. Unfortunately, due to the many specifics of WMN and limitations of the target, application is not possible to easily utilize such algorithms.

Flooding in a general mesh network is based on propagation of the packet over the whole network and is to be considered as the most reliable for WMN. Real implementations of WMN in industrial, scientific and medical ISM radio bands are limited physically by the connection speed, the so called bit rate, resulting in big delays and low network responsiveness.

Routing algorithms based on the sharing and distribution of routing tables or vectors are usually considered to be the most efficient. High memory demands and big overload in the case of distribution routing tables usually limit usage of such algorithms for larger WMNs where resources of nodes controllers are not limited by the economy of the project.

Possible packet collisions in connection with lower bit rates limits real implementations of random routing in WMN and practically disqualifies telemetry and control applications where the highest reliability should be achieved.

WMNs are nowadays considered and already used as a communication platform for many different applications in the field of telemetry and automation. Automatic meter reading AMR, street lighting control or smart grids are just few examples of such applications utilizing networks with hundreds or thousands of devices. Therefore both the cost of communication devices and high reliability of the routing should be priorities. Technology described in this paper provides both reliable and effective packet delivery solutions with minimal demands on system resources, it is an extension of the paper discovering technology at ICN2012.

II. RELATED WORK

Efficient and reliable packet delivery in large wireless networks consisting of hundreds or even thousands of devices and supporting up to several hundred hops is a big algorithmic challenge. Considering actual speed, output power and spectrum limitations, as well as economic factors, a flooding mechanism seems to be the most viable for most target applications.

 ISBN: 978-1-61208-186-1

Therefore, flooding is commonly used in wireless ad-hoc networks. There are many techniques of flooding differing in control algorithms, efficiency, reliability and overhead.

The simplest flooding technique is based on re-transmitting only new, not yet registered packets. In this scheme every packet should be identified and is re-transmitted only once. This mechanism guaranties that a packet is delivered to all nodes at minimal costs. In the real environment of a WMN, collisions would affect functionality and result in high traffic [1]. Reducing flooding traffic is the goal of many approaches to make the flooding mechanism more reliable and efficient [1]. Proposing a probable flooding scheme, e.g., distance-based, location-based or cluster-based flooding.

Schemes in category 1-hop neighborhood are based on knowledge of the closest neighbors reachable directly in one hop. Different approaches [2-5] are based on 1-hop neighborhood knowledge. Cai et al. [2] propose adding the list of its 1-hop neighborhood to the packet, and recommends to the receiver not to forward the message if its complete 1-hop neighborhood is already included in the received list.

Schemes in the category 2+ hops neighborhood are based on storage of the neighborhood, which is limited by the number of hops from each node. In this category every node knows the network topology up to n-hops. In [6] Qayyum et al. propose a heuristic algorithm to compute multiple relays. Ko et al. in [7] propose improving broadcast operations for ad-hoc networks using 2-hop connected dominating sets.

Spohn et al. in [8] argue by simulations that protocols focused on making an optimal broadcast tree with the implicit assumption that all nodes should be reachable from the source may no longer be true because flooding protocols in wireless sensor networks are used to deliver data packets towards a single, or only subset, of destination nodes and proposes a new flooding protocol for utilizing directional information to achieve efficiency in data delivery.

Time Synchronized Mesh Protocol, described in [9] is based on TDMA and requires sharing time information and precise time synchronization of all nodes. Overall power consumption would benefit from the time synchronization, but on the other hand interference, collisions and environment influences would impact delivery reliability.

III. IQMESH

IQMESH is the networking technology developed for WMN with a coordinator and utilization of packet transmissions. Reliability is achieved through a flooding mechanism, collisions are avoided by TDMA and its routing efficiency is based on the virtual routing structure VRS, created by the coordinator during discovery. The following paragraphs provide a step by step analysis of particular parts of the IQMESH technology.

A. Basic principles

WMN is a general network of devices connected wirelessly. Every device in the network has some unique information (address) enabling addressing inside of the network - MAC, node ID, index, address, etc. Packets sent in such network include the address of the recipient. The principle of IQMESH technology is to extend this addressing space and define a new virtual routing structure in the network. A coordinator will dedicate to every device, found during discovery process, a unique Virtual Routing Number VRN, which will be used for future routing. Figure 1a shows an illustrative example of a standard network, where its nodes can be addressed by their address N1 – N5, after discovery, additional routing information is added as shown in figure 1b. Only VRN are used for the routing, while devices' addresses are used solely for addressing. Flooding and other routing algorithms can benefit from systematic indexing of nodes by VRN, e.g., if the VRN reflects distance by the number of routing hops from coordinator to the node.

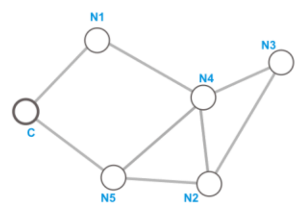

Fig. 1a Network example

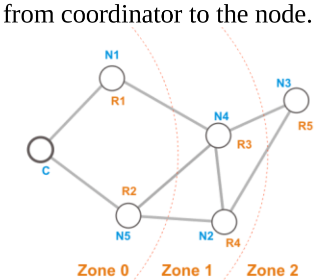

Fig. 1b Network after discovery

B. Discovery

During discovery, the coordinator seeks out nodes connected to the network and dedicates them to a unique VRN, reflecting their distance from the coordinator. For example, an incremental indexing can be used. The coordinator starts to search its neighbors. All devices responding to its "Answer Me" type message will receive their VRN. Based on received response it is assumed that the link between coordinator and responding nodes should be symmetrical, enabling future routing both to and from the coordinator. All nodes directly responding to the coordinator have a direct link to the coordinator, so they should belong to Zone 0, being directly accessible without routing. Then, the coordinator incrementally asks all nodes from Zone 0 to discover their 1-hop neighborhood and then dedicates a VRN to all newly found nodes which have not been found in the previous step, and thus not belonging to the Zone 0. Each node can also store some additional information in this step, e.g., respective zone number, parent's VRN, parent's network address or VRN of the first node in respective zone. This information would later be used for routing optimization. Processing all answers from nodes belonging to Zone 0, the coordinator will know all nodes belonging to Zone 1, which are nodes accessible to the coordinator by one routing hop. The same procedure will be then invoked recursively for all nodes belonging to zones Zone 1 and higher until all nodes are found or until there are some further zones available. At the end of discovery every found, and thus discovered, node has a unique VRN reflecting its distance from the coordinator. In typical applications such as smart buildings, telemetry systems and street lighting the discovery is made just once during the installation phase.

C. Routing

The goal of packet routing in target applications is to reliably and efficiently deliver data over the network. In IQMESH based networks, the flooding mechanism is primarily used. VRS created during discovery process is directionally flooded. The network would be flooded from the coordinator to the node for all control purposes or from the node to the coordinator for data collection. A special order of VRS together with TDMA enables a directional, efficient and collision free flooding mechanism based on VRN. Every node routing packet in its dedicated time slot will also add to the packet its own VRN_X enabling other nodes to know and consequently synchronize to their respective time slot. The coordinator uses VRNC equal to 0. The network routing mechanism is illustrated in Fig. 2.

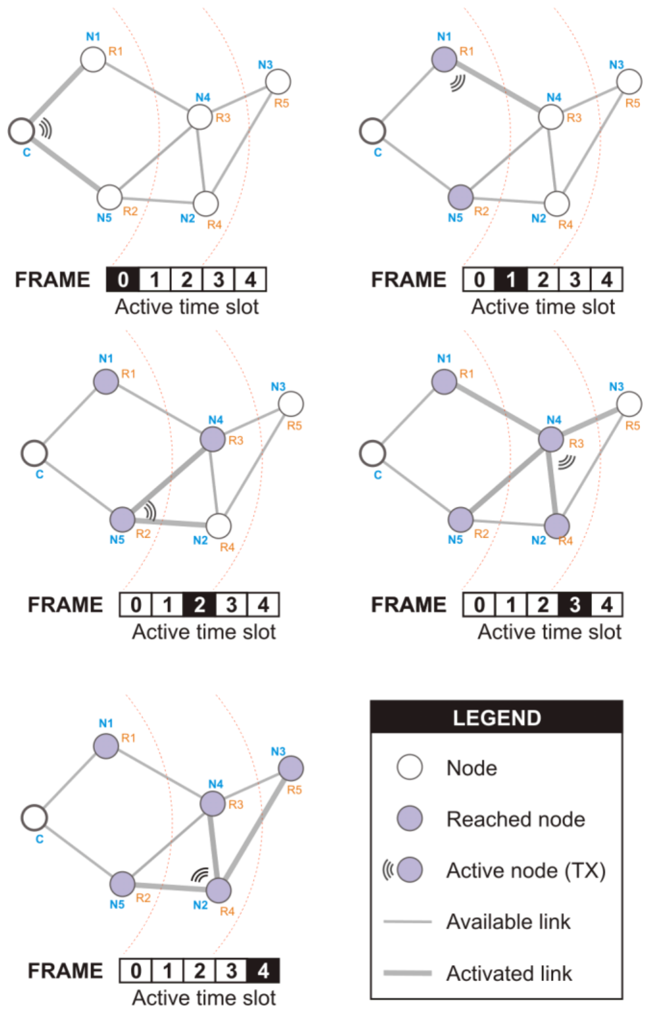

Fig. 2 Routing mechanism

D. Time synchronization

In contrast to many other techniques, e.g., [9], there is no need to share precise time information over the network. Routed packets keep track of number of hops made. This number corresponds with the respective time slot based on VRN for every device. This mechanism, together with adding information of length of time slot, enables efficient re-synchronization of all devices based on packet reception. In addition, dynamic timing based on packet length can be supported.

E. Resources

Generally, the coordinator needs to hold the information of the discovered nodes, whereas each node should keep VRN information only. This means no special HW resources are required to implement IQMESH technology. Storing some additional information recognized during discovery, as described in future paragraphs, can dramatically increase the efficiency of the routing. Specific IQRF implementation showing real system requirements will be mentioned further on.

F. Optimizations

Additional system resources dedicated to the coordinator or to each node can increase future routing efficiency. Storing the address or VRN of a parent node by each node could be mentioned as a good example. In such case, every node can reach the coordinator quite efficiently by tree topology via parent nodes. Also, the storing of minimal VRN in the respective zone, or equivalent saving of the maximal VRN of the previous zone by each routed node can be used to increase routing efficiency. Software techniques used during discovery or following discovery process can also increase the final efficiency of the routing, e.g., the coordinator can exclude all leaf nodes without child links from VRS during discovery.

G. Reliability

The described IQMESH technology ensures reliable and efficient directional flooding. Due to the expected redundant links in many real WMNs, it dramatically increases reliability. A temporarily lost link will obviously not cause failure of packet delivery. Routing mechanisms making use of underlying TDMA avoid conflicts as every routing node has just one dedicated time slot corresponding to its VRN. Many tests with environment noise simulations have confirmed reliability increase. Usually only noise generated during time slots dedicated for devices without redundant links can cause failure in packet delivery. Noise generated during the first time slot usually affects delivery, as no redundant links have yet to be created, which is a first time slot issue FTSI. Overall performance in standard office building environments was measured and a fail rate based on several weeks of experimentation resulted in 1 not delivered packet from 17 250 transmissions. Two additional slots for the coordinator were used to fix the FTSI.

H. Efficiency - routing from coordinator

As redundant paths resulting from the principle of VRS flooding are expected and packet collisions are eliminated by TDMA, it might not be obvious to use reception acknowledgment during flooding. However, this assures fair routing efficiency without any impact to reliability. Based on the TDMA, flooding routing realized via VRS for nodes with $VRN_X < VRN_A$, where VRN_A is VRN of addressed node, and assumption that every node is addressed in the same frequency, the average frame will consist of time slots, where n is the number of nodes in a WMN:

$$T_{AVG} = \frac{\sum_{k=1}^{n} \mathrm{VRN_k}}{n} = \frac{n}{2} \quad (2)$$

Generally, blind flooding efficiency in similar cases would be calculated as a number of links to ensure 100% reliability of packet delivery. Comparing (1) and (2), we can see dramatic efficiency increase.

For any addressed node within the zone Z_x, only nodes belonging to previous zone Z_{x-1} should make the routing without any strong impact to the reliability. The resulting efficiency based on this presumption will be higher, but always dependent on the topology of the specific WMN. The following formula reflects expected system efficiency for such a scenario:

$$T_{AVG} \leq \frac{n}{2} \quad (3)$$

Efficiency of this routing scenario, skipping redundant routes by nodes in the same zone Z_x for specific node X, can be expected as (4), where VRN_{Zx-1} is VRN of the last node related to the zone Z_{x-1}. Based on this principle, all nodes related to the zone Z_0 can be addressed directly without routing.

$$T_{AVG} = VRN_{Z_{x-1}} \quad (4)$$

I. Efficiency - routing to coordinator

As the matter of fact, there is information about parent nodes recognized and stored during discovery of every node. This information means that there is a tree topology available for routing packets from nodes to the network coordinator. In such a scenario, every routing node in its time slot sends a packet exclusively to its parent. Therefore in using TDMA, the number of time slots is equal to the number of hops and, for each frame, corresponds to the zone number Z_X for the node originating communication. Assuming Z_{MAX} as a maximum zone number in the network while indexed from 0, it would be generally proven:

$$T_{AVG} \leq Z_{MAX} \quad (5)$$

Reliability increase is mostly preferred in typical WMN applications. Oriented flooding with redundant backup paths can be used for such applications. In this case, each node originating communication to the coordinator will use its own VRN_X number as a limit of hops. For such routing, similar efficiency like (2) is expected.

Avoiding redundancy of routing by nodes from the same zone, routing efficiency for a specific node X can be expressed similarly like in formula (4), where is a VRN of the last node related to the zone Z_{X-1}.

IV. IQMESH IMPLEMENTATION IN IQRF PLATFORM

IQRF is the communication platform and related technologies [10]. The name IQRF stands for an Intelligent Radio Frequency. Basic specifications and early designs were made in 2004, when, in Malaga, Spain, the first integrated modules were introduced. IQRF is the platform integrating a variety of components for building LR-WPAN in an easy way, simplifying and shortening the design phase of a wireless communication system. Specific implementation of IQMESH routing technology will be described in following paragraphs.

A. Network abstraction and limitations

IQRF platform addresses mainly LR-WPAN applications, such as building automation systems and telemetry systems. In such applications many devices can be connected in a fixed infrastructure, usually created during the installation process, and provides permanent links to other devices in this infrastructure. There are commonly other devices connected in the network without permanent links to the other devices in the network, e.g., because of the mobility of such devices or because of power supply limitations. Based on these criteria, IQRF abstracted network topology is described in Figure 3.

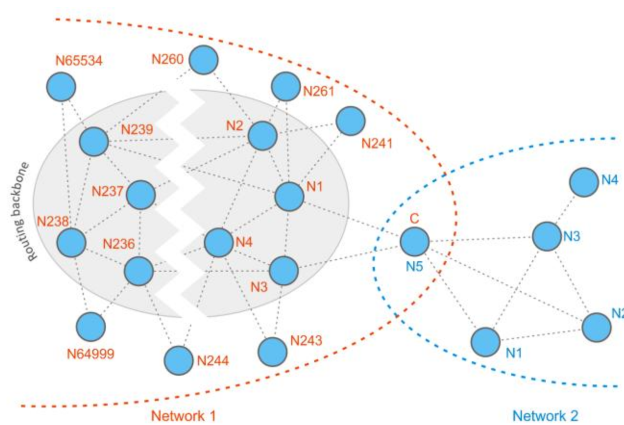

Fig. 3 IQRF abstracted network topology

The routing backbone usually consists of devices providing permanent links and bonded to the network during the installation process. Typical examples for such devices would be equipment like gateways, actuators or sensors mounted in specific locations. Such devices can be used as a routing backbone. IQMESH defines Virtual Routing Structure (VRS) to be used for directionally oriented flooding. VRS would be created from such infrastructure during the installation process (or later on). IQRF OS available on every transceiver module provides functionality for automation of such processes through function *discovery(x)* instigated by the coordinator and via servicing of system packets by node devices.

As visible from Figure 3 and description available in [10], several limitations were applied to increase efficiency of the network. Routing backbone consists of up to 239 devices, allowing very efficient one byte addressing, broadcast and also group messages. VRS is created from routing backbone devices very simply by calling *discovery(x)* function, where parameter *x* limits the number of zones, which is related to a depth of the network. Other devices will receive logical network address 0xFE from the coordinator during bonding. To extend the addressing scheme for addressing 65k devices, additional two byte user addresses are used. Every IQRF device can work simultaneously in two wireless networks, further extending the network or chaining of several networks allowing the possibility to build up larger systems.

IQRF specific implementation of IQRF OS 3.0x on TR-5xx transceiver modules supports several non-routing and routing schemes. A specific routing scheme is chosen in application based on requested efficiency and purpose by setting system variable *RTDEF*. Routing based on network logical address, tree routing to the coordinator and routing based on VRS are three basic routing schemes supported by current implementation of IQRF OS. Addressing in the network is realized via the logical network address obtained during bonding which is one byte long or via a two byte user address dedicated by the user by calling function *setUserAddress(x)*.

B. Dynamic time slots

To avoid conflicts within the network during routing, TDMA is used. One frame can include up to 240 time slots, allowing for up to 240 hops in the network. In IQRF, time slots are measured and set up in ticks, every tick is 10 ms long. As data load in the packet can consist from 1 byte up to 128 bytes, and 19.2 kb/s is the typical bit rate used for transmissions, 1.2 kb/s up to 115 kb/s are supported, the length of the frame would be too long if a fixed time slot is used.

Support of dynamic time slots based on the data load in the packet and requested purpose dramatically increased routing efficiency. Time slot is defined by setting the variable *RTSLOT* to the number of ticks convenient for a specific purpose. Polling request of the coordinator, e.g., can include just a specification of one or more nodes which should send data to the coordinator. In such a case, a minimum time slot 1 tick long can be used to propagate this request over the network, assuring delivery in 2.4 s in the worst case to any device. Time slot 5 ticks long together with simultaneous choosing of tree routing schemes can be used for a 128 byte long answer to assure maximum time efficiency.

Routing description, such as examples demonstrating routing and right parameters setup for specific purposes are available at User's and Reference guides, download-able from [10].

C. System resources

IQRF OS, including complete support both for the coordinator and nodes, is ported to TR-52Bx modules based on a PIC16F886 microcontroller. System resources used for routing and related services:

Program memory:	< 2k instructions
Data memory:	< 40 bytes (node mode)
	< 300 bytes (during discovery)
EEPROM:	< 40 bytes (node mode)
	< 2k bytes
Packet overload:	+ 6 bytes

D. Development, testing, results

The standard testing environment during development was based on a set of 200 node devices, each device including transceiver module TR-52BA inserted into a DK-EVAL-03 evaluation board and a GW-USB-04 In Circuit Wireless Programmer enabling bulk programming of all devices by one click and from one coordinator device consisting of transceiver module TR-52BA and CK-USB-02 universal programmer / debugger.

In-building applications mainly for lighting and dimming control realized in networks consisting of hundreds and thousands of devices confirmed reliability and the ability to work in real time. On the other hand, due to such a local environment, just a few hops were needed to cover the whole building.

The real challenge was street lighting control, covering large parts of towns, with networks composed of up to 200 devices with different networks using different channels to avoid spectrum concurrency. Fig. 4 shows one of such implementations realized in the suburb of Nitra, Slovakia, EU. Several kilometers range were covered by devices based on transceivers supporting only 3.2 mW of output power with small PCB antenna.

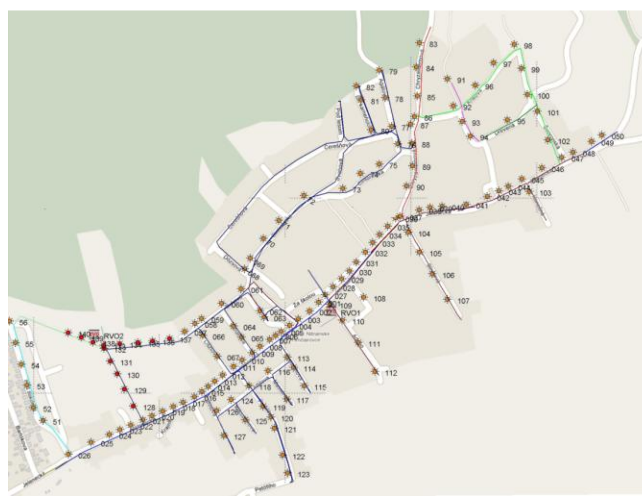

Fig. 4 Real implementation of street lighting application

V. IQMESH IMPLEMENTATION

Protocol implementation on specific hardware of TR-53BA transceiver modules will be briefly described and examples of the code will be presented to demonstrate easiness of use.

 ISBN: 978-1-61208-186-1

A. TR-53BA module with built-in operating system

Transceiver module TR-53BA is a tiny electronic board with complete circuitry needed for realization of wireless RF connectivity. It is a basic communication component of the IQRF platform, used also in all IQRF gateways, routers and devices. Transceiver modules operate in the 868 MHz and 916 MHz license free ISM (Industry, Scientific and Medical) frequency bands. Modules can be used as a communication peripheral for any electronic device, or, due to the high integration, also as a controlling board for stand-alone applications.

High integration and functionality implemented in operating system dramatically reduce the time of application development, while ultra low power consumption predetermines modules for use in a battery powered applications. Mechanical concept of the module allows optional montage to any board equipped by inexpensive SIM card connector and also soldering to the mother boards when high mechanical stability is requested.

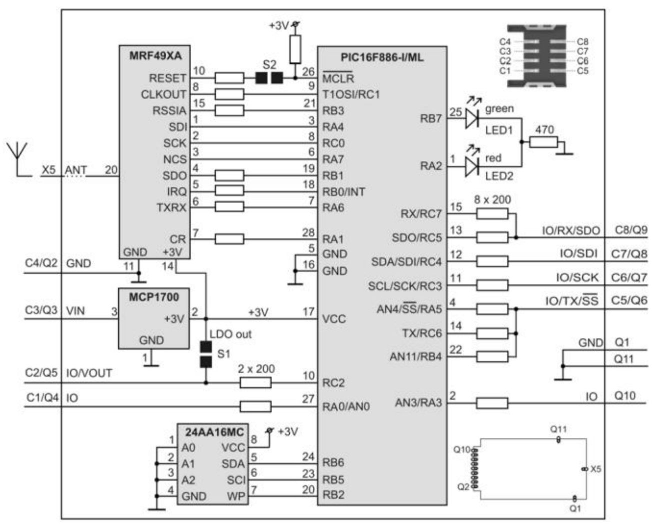

Fig. 5a TR-53BA simplified schematics

Fig. 5 shows simplified schematics and Fig 6 physical layout and main building blocks of the module. Microcontroller provides both input/output pins and basic interfaces for serial connection. Both RF and networking functionality is implemented in built-in operating system, so programmer can focus on his application only.

Detailed description of IQRF OS functionality, system architecture and both RF and networking related functionality are described in User's Guide [11] discovering packets description, bonding process, addressing techniques, description of implemented routing algorithms and examples. Reference Guide [12] provides detailed description of all available functions and shows their usage using basic examples.

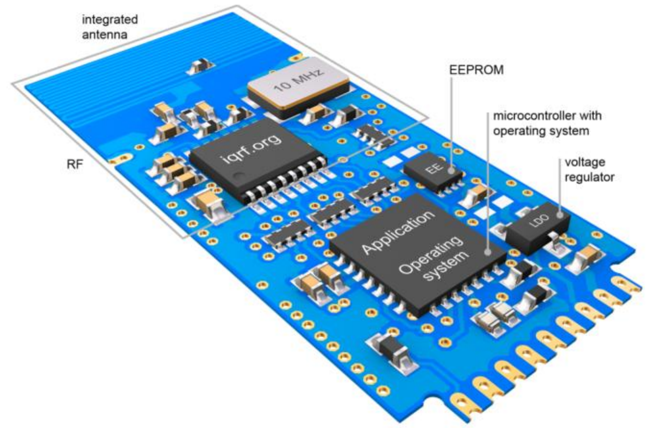

Fig. 6 TR-53BA physical layout

Operating system functionality related to wireless communication and networking will be described in future paragraphs and examples of the code will be provided for easier understanding.

B. Communication - packet transmission

As described in [11], IQRF OS has buffer oriented architecture. Sending a packet is realized by call of function *RFTXpacket(),* which takes *DLEN* bytes from *bufferRF* dedicated to RF communication and executes packet transmission. In the same way, packets receiving is realized via calling function *RFRXpacket()*. If some packet comes, data will be available in *bufferRF* and variable *DLEN* will hold the value indicating data length in the received packet. Code sequence Code 1 shows basic construction of packet transmission, code sequence Code 2 shows realization of simple receiver.

```
// Code 1 - packet transmission

PIN = 0;              // Peer-to-peer mode
DLEN = 10;            // 10 bytes TX definition
RFTXpacket();         // Transmits 10 bytes
```

```
// Code 2 - simple receiver

...

while (1)             // main loop
{
  if (RFRXpacket())
  {
     // ... code executed on packet reception
  }
  else
  {
     // ... code executed if timeout occurred
  }
```

C. Networking modes - Coordinator / Node

Besides peer-to-peer mode, networking mode is supported. Every device can work simultaneously in two different networks, in one network works like a Node, while in second network take roles of network Coordinator. This feature can be helpful for building wider networks by chaining networks or their fragments together. Switching between networks is done automatically upon successful packet reception. In this case, the following transmission will be implicitly addressed to the network from which packet came. Code sequence Code 3 describes functionality related to the switching between networks and packets relaying.

```
// Code 3 - networking modes


if (RFRXpacket())
{
  if (!_NTWF)          // ignore non-networking
    continue;          //    packets

  x = getNetworkParams();

  if (x.7)             // Coordinator is polling data
  {
    bufferRF[0] = MYSTATUS;
    DLEN = 1;
    RX = 0;            // RX defines addressee (C now)
    RFTXpacket();      // data sent back
  }
  else                 // Nodes in Net1 sending data
  {                    // data relayed to Coordinator
    setNodeMode();
    RX = 0;
    RFTXpacket();
  }
}
```

D. Bonding to other networks

In *Coordinator mode*, transceiver is managing its network of bonded Nodes, while in *Node mode* it is controlled by different Coordinator. Nodes should be bonded to the network before being able to send packets addressed to this network. Bonding is based on Node's request confirmed by the Coordinator via exchanging RF system packets.

An example of bonding initiated by Node_01 is shown in Code 4a and 4b. In this example, Node with address 1, e.g. a specialized device as a remote controller, requests the Coordinator to bond a new device to the network. Bonding on Node's side is initiated by pressing a button in this example.

```
// Code 4a - Bonding on Coordinator's side

if (RFRXpacket())
{
  if (!_NTWF)          // ignore non-networking
    continue;          //    packets

  if (TX == 1)         // packet sent from Node_01
  {
    if (bondNewNode())
    {
      pulseLEDG();
     // ... code called if new bond created
    }
    else
    {
     pulseLEDR();
     // ... code called when bonding failed

    }
  }
}
```

```
// Code 4b - Node bonds to the network

if (buttonPressed)
{
  if (bondRequest())
  {
    pulseLEDG();
    // ... code called if successfully bonded
  }
  else
  {
    pulseLEDR();
    // ... code called when bonding failed
  }
}
```

E. Routing

IQMESH implementation in IQRF OS 3.00 supports up to 239 routing devices for a packet. Several routing algorithms are supported, allowing programmers to change them based on specific topology and needs, e.g. reliability, speed or response time. OS ensures that the packet is ignored by all devices except of the addressee and routing devices.

Routing algorithm should be specified with respect to reliability and speed requirements. Routing algorithm is specified by setting up variable *RTDEF*, defining requested routing algorithm.

Implicitly, every Node device can route packets. As it is not convenient at every case, e.g. devices with only one connection (leafs) or moving devices, routing can be selectively disabled or re-enabled for individual nodes. This functionality is realized via functions *setRoutingOn()* and *setRoutingOff()*.

Discovery is the process invoked by network Coordinator to search all devices belonging to its network and automatically dedicate Virtual Routing Numbers to Node devices. It is executed by calling function *discovery(x)*, where parameter *x* specifies maximum zone number of the network to be discovered. Discovery function is usually called during the installation process or for healing purposes, in the case of major topology changes. Standard mesh routing mechanisms used for routing uses redundant routes, so minor changes usually do not requests explicit healing process or rediscovery.

Code sequence shown in Code 5a demonstrates polling data from specific Node, while sequences in Code 5b and Code 5c show returning data from the Node back to the Coordinator. This example also shows loop mechanism used for addressed Node synchronization to avoid transmission conflicts.

```
// Code 5a - Coordinator polls data from the Node

setCoordinatorMode();
RX = 1;                 // Node to be addressed
DLEN = 1;               // command sent to the Node
PIN = 0;
_ROUTEF = 1;            // routing requested
RTDEF = 2;              // DFM algorithm chosen
RTMAX = _HOPS;          // Number of hops specified
RTSLOT = 1;             // ticks dedicated for time slot
RFTXpacket();           // packet sent
```

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-186-1

```
// Code 5b - Node response by original routing scheme

if (RFRXpacket())
{
  if (!_NTWF)          // ignore non-networking
    continue;          //    packets

  if (TX != 0)         // allow only packets sent from
    continue;          //    Coordinator (address 0x00)

  if (bufferRF[0] != 1)// allow only polling command 1
    continue;          //    specified in the packet

  while (RTSLOT)       // synchronization to avoid
  {                    //    network conflicts
    waitDelay(RTDT1);
    RTSLOT--;
  }

  RX = 0;              // return back to Coordinator
  copyBufferCOM2RF();  // prepare data to the bufferRF
  DLEN = 64;           // 64 byte packet definition
  RTMAX = 0xFF;        // 0xFF translated to VRN
  RTSLOT = 5;          // each timeslot 5 ticks long
  RFTXpacket();        // sends packet
}
```

```
// Code 5c - Node response by TREE routing scheme

if (RFRXpacket())
{
  if (!_NTWF)          // ignore non-networking
    continue;          //    packets

  if (TX != 0)         // allow only packets sent from
    continue;          //    Coordinator (address 0x00)

  if (bufferRF[0] != 1)// allow only polling command 1
    continue;          //    specified in the packet

  while (RTSLOT)       // synchronization to avoid
  {                    //    network conflicts
    waitDelay(RTDT1);
    RTSLOT--;
  }

  RX = 0;              // return back to Coordinator
  RTDEF = _TREE;       // tree routing scheme
  RFTXpacket();        // sends packet
}
```

VI. CONCLUSION

IQMESH networking technology for wireless mesh networks, its basic principles and related routing algorithms were presented. Easiness of use and code efficiency was demonstrated on several examples. Specific implementation in IQRF communication platform was described. Like with any other technology or algorithm, IQMESH applications can benefit from technological advantages and would be affected by its limitations. The flooding scheme would be an excellent option for telemetry systems, e.g., AMR applications for water meters providing data just a few times a day or for street lighting applications. On the other hand, many redundant links and consequent time delays can generate difficulties in real time applications and missing support for node to node communication would create more programming work on the application layer.

VII. FUTURE WORK

Spreading over frequency spectrum instead of TDMA, a combination of both methods, achieving higher efficiency of the routing, increasing reliability in noisy environments and the usage of described technology are just few topics for future research. Future advanced data aggregation algorithms can benefit from VRS and routing schemes.

VIII. ACKNOWLEDGEMENT

This research has been supported and co-financed by the Ministry of Industry and Trade of the Czech Republic under contracts FR-TI1/058 "Project Smart House - Open Platform" and FR-TI3/27, project "Open Platform for Modern Cities".

[1] S. Y. Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. ACM MOBICOM, pp. 51-162, Aug' 1999.

[2] Ying Cai, Kien A. Hua, and Aaron Phillips, "Leveraging 1-hop Neighborhood Knowledge for Efficient Flooding in Wireless Ad Hoc Networks", 24th IEEE International Performance Computing and Communications Conference (IPCCC), April 7-9, 2005, Phoenix, Arizona.

[3] Xinxin Liu, Xiaohua Jia, Hai Liu, and Li Feng, "A Location Aided Flooding Protocol for Wireless Ad Hoc Networks", MSN 2007, LNCS 4864, pp.302-313, 2007.

[4] H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," In Proc. of the ACM Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile System (MSWIM), pp 61-68, Aug. 2000.

[5] Hai Liu, Pengjun Wan, Xiaohua Jia, Xinxin Liu and Frances Yao, "Efficient Flooding Scheme Based on 1-hop Information in Ad Hoc Networks", in proceedings IEEE infocom, Communications Society subject, 2006.

[6] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks," In Proceeding of the 35th Hawaii International Conference on System Sciences, 2002.

[7] M. A. Spohn, J.J. Garcia-Luna-Aceves, "Improving Broadcast Operations in Ad Hoc Networks Using Two-Hop Connected Dominating Sets," In Proceedings of IEEE Global Telecommunications Conference Workshops, 2004.

[8] A New Directional Flooding Protocol for Wireless Sensor Networks Young-Bae Ko, Jong-Mu Choi and Jai-Hoon Kim, Information Networking. Networking Technologies for Broadband and Mobile Networks Lecture Notes in Computer Science, 2004, Volume 3090/2004, 93-102, DOI: 10.1007/978-3-540-25978-7_10

[9] Technical Overview of TSMP

[10] Datasheets and examples from http://www.iqrf.org [October 15, 2011]

[11] IQRF OS v3.00 User's guide at http://www.iqrf.org/ug [January 15, 2011]

[12] IQRF OS v3.00 Reference guide at http://www.iqrf.org/rg [January 15, 2011]