

IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS

A PUBLICATION OF THE IEEE COMMUNICATIONS SOCIETY



SEPTEMBER 2010

VOLUME 28

NUMBER 7

ISACEM

(ISSN 0733-8716)

SIMPLE WIRELESS SENSOR NETWORKING SOLUTIONS
M. Dohler, K. S. J. Pister, W. Heinzelman, M. B. Srivastava, I. Stojmenovic, K. Römer

Guest Editorial M. Dohler, K. S. J. Pister, W. Heinzelman, M. B. Srivastava, I. Stojmenovic, K. Römer 969

Standardization

IEEE 802.15.5 WPAN Mesh Standard-Low Rate Part: Meshing the Wireless Sensor Networks
. M. J. Lee, R. Zhang, J. Zheng, G.-S. Ahn, C. Zhu, T. R. Park, S. R. Cho, C. S. Shin, and J. S. Ryu 973

Coverage, Localization, and Security

Constructing Low-Connectivity and Full-Coverage Three Dimensional Sensor Networks
. C. Zhang, X. Bai, J. Teng, D. Xuan, and W. Jia 984

Simple Movement Control Algorithm for Bi-Connectivity in Robotic Sensor Networks
. H. Liu, X. Chu, Y.-W. Leung, and R. Du 994

Distributed Target Tracking Using Signal Strength Measurements by a Wireless Sensor Network.
. A. Oka and L. Lampe 1006

Joint Synchronization and Localization Using TOAs: A Linearization Based WLS Solution. S. Zhu and Z. Ding 1016

NLOS Identification and Mitigation for Localization Based on UWB Experimental Data
. S. Marandò, W. M. Gifford, H. Wymeersch, and M. Z. Win 1026

A Randomized Countermeasure Against Parasitic Adversaries in Wireless Sensor Networks.
. P. Papadimitratos, J. Luo, and J.-P. Hubaux 1036

Distributed Sensing Architecture

Simple, Zero-Feedback, Distributed Beamforming with Unsynchronized Carriers
. A. Bletsas, A. Lippman, and J. N. Sahalos 1046

A Co-operative Beamforming Solution for Eliminating Multi-Hop Communications in Wireless Sensor Networks
. A. Kalis, A. G. Kanatas, and G. P. Efthymoglou 1055

(Contents Continued on Back Cover)



IEEE 802.15.5 WPAN Mesh Standard-Low Rate Part: Meshing the Wireless Sensor Networks

Myung J. Lee, Rui Zhang, Jianliang Zheng, Gahng-Seop Ahn, Chunhui Zhu, Tae Rim Park, Sung Rae Cho, Chang Sub Shin, and Jun Sun Ryu

Abstract—This paper introduces a new IEEE standard, IEEE 802.15.5, which provides mesh capability for wireless personal area network (WPAN) devices. The standard provides an architectural framework enabling WPAN devices to promote interoperable, stable, and scalable wireless mesh topologies. It is composed of two parts: low-rate WPAN mesh and high-rate WPAN mesh. In this paper, we present only low-rate WPAN mesh because it is designed to support wireless sensor networks. IEEE 802.15.5 low-rate part is a light-weight scalable mesh routing protocol that caters well to the requirements of resource-constrained wireless sensor networks. By binding logical addresses to the network topology, IEEE 802.15.5 obviates the need for route discovery. This eliminates the initial route discovery latency, saves storage space and reduces the communication overhead and energy consumption. A distributed link state scheme is further built atop the block addressing scheme to improve the quality of routes, robustness, and load balancing. The routing scheme scales well with regard to various performance metrics. The standard also provides enhanced functions such as multicast, reliable broadcast, power saving, time synchronization, route tracing and portability. We also present the performance evaluation of major functions performed with a 50-nodes tested deployed over a whole floor (100 x 140 ft²) at CUNY Engineering building. The results testify that the IEEE 802.15.5 will serve well for wireless personal area networks and wireless sensor networks.

Index Terms—WPAN, Mesh, IEEE 802.15.5.

I. INTRODUCTION

A MILESTONE of the development of wireless sensor networks is the release of IEEE 802.15.4 standard[1] in 2003. IEEE 802.15.4 is the first global standard for wireless personal area networks and wireless sensor networks (simply referred to as wireless sensor networks hereafter). As a Physical layer and MAC sublayer standard, IEEE 802.15.4 does not specify how to provision multi-hop routing (a.k.a. meshing) capabilities for wireless sensor networks, assuming it is the task of upper layers. IEEE has recently released IEEE 802.15.5 standard[2] to provide multi-hop mesh functions. This standard is tightly coupled with IEEE 802.15.4 to take full advantage of IEEE 802.15.4 while maintaining simplicity. In this paper, we introduce the features and functions of this new standard. The standard is composed of two parts: low-rate WPAN mesh and high-rate WPAN mesh. Our discussion

in this paper is however confined to the low-rate part because it is designed to support wireless sensor networks. We also present the performance evaluation of the low-rate part using a 50-node tested deployed over a whole floor (100 x 140 ft²) at CUNY Engineering building.

Several other standards organizations also come into play to support mesh functions on top of IEEE 802.15.4. ZigBee[3], an industrial alliance, is among the first who have been working on upper layers to provide multi-hop routing functionalities for wireless sensor networks. 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) and ROLL (Routing Over Low power and Lossy networks), two IETF working groups of the Internet Engineering Task Force (IETF), have been trying to extend IPv6 to wireless personal area networks. Another two important players are the HART (Highway Addressable Remote Transducer) Communication Foundation (HCF) and the International Society of Automation (ISA), both having been retrofitting their control and automation protocols to accommodate automation industries with wireless solutions.

To see where IEEE 802.15.5 stands, let us first highlight the features of the aforementioned solutions in the context of mesh networking. ZigBee ratified its first standard in 2004. Before another distinct standard called ZigBee Pro was released in 2007, ZigBee routing comprised AODVjr[4] and Cluster Tree [5]. In ZigBee Pro, Cluster Tree has been replaced by a stochastic addressing scheme, which picks up a logical short address randomly from the 16-bit address pool for assignment. This approach solves the "running out of addresses" problem, but loses the ability to route packets merely based on logical short addresses and requires additional mechanism for resolving possible short address conflicts. Up to date, however, ZigBee and ZigBee Pro are still not scalable enough to support very large scale wireless sensor networks, and they lack the support for battery-powered routers.

One of the key operations of 6LoWPAN is header compression. Out of the 127 bytes of the IEEE 802.15.4 data frame, only 81 to 102 bytes are available to upper layers. This poses a big challenge to 6LoWPAN as how to fit the minimum 1280-byte IPv6 maximum transmission unit (MTU) into the small payload of IEEE 802.15.4 data frame. Simple fragmentation without compression is unlikely viable. 6LoWPAN uses both stateless compression[6] and stateful compression[7][8] schemes. By exploiting cross-layer redundancy and leveraging shared state within contexts, 6LoWPAN can compress the adaptation, network, and transport layer header fields all into a few bytes. Notwithstanding that 6LoWPAN only adds 7 to 12 bytes of overhead to a raw IEEE 802.15.4 data frame, which

Manuscript received 1 May 2009; revised 16 February 2010.

M. J. Lee, R. Zhang, G.-S. Ahn, and J. S. Ryu are with the Department of Electrical Engineering, City University of New York, New York, NY, 10031, USA.

J. Zheng is with EMC, USA.

C. Zhu and T. R. Park are with Samsung in USA and Korea respectively.

S. R. Cho is with Chung-Ang University, Korea.

C. S. Shin is with ETRI, Korea.

Digital Object Identifier 10.1109/JSAC.2010.100902.

is even less than ZigBee's 7 to 15 bytes of overhead, each node needs to implement the basic IPv6 protocol stack plus the newly introduced adaptation layer so as to be IP-capable. This consumes the precious memory and makes the operations more complicated. Therefore, the decision to make an IP-capable low power wireless sensor network should be carefully made considering the involved cost and benefit.

Recently an IPv6 Routing Protocol for Low power and Lossy Networks (LLNs), referred to as RPL[9], has been submitted to IETF ROLL working group. RPL is a close comparison to IEEE 802.15.5 mesh routing, since both are based on logical tree structure and yet try to overcome the shortcomings of early logic tree routing protocols such as Cluster Tree[5]. For example, RPL uses Directed Acyclic Graph (DAG), which is equivalent to logical tree used in IEEE 802.15.5. RPL relies on rank and IEEE 802.15.5 relies on tree level to detect and solve loop problems.

The HART Communication Foundation (HCF)[10] rolled out a wireless mesh networking standard, WirelessHART, in 2007. WirelessHART uses TDMA to provide two important features that are needed for industrial sensing and control but currently missing from ZigBee Pro: deterministic latency and deterministic reliability. A strong competitor of HCF is the International Society of Automation (ISA), who has been developing its own industrial wireless mesh networking standard, officially noted as ISA 100.11a (often referred to as ISA SP 100), which also supports the two features. WirelessHart and ISA 100.11a standards do not serve as upper layer protocols of IEEE 802.15.4 because these standards use only the physical layer of IEEE 802.15.4.

The new standard IEEE 802.15.5, can be delineated by two key words, *mesh* and *simplicity*.

In terms of *mesh*, IEEE 802.15.5 is a light-weight scalable mesh routing protocol that well caters to the requirements of resource-constrained wireless sensor networks. By binding logic addresses to the network topology, IEEE 802.15.5 obviates the need for route discovery. This eliminates the initial route discovery latency, saves storage space compared to routing table, and reduces the communication overhead and energy consumption. IEEE 802.15.5 employs a flexible block addressing scheme for logic address assignment as well as network auto-configuration. The scheme takes into account the actual network topology and thus is fully topology-adaptive. A distributed link state scheme is further built atop the block addressing scheme to improve the quality of routes, in terms of hop count or other routing cost metrics used, robustness, and load balancing. All this is done during network initialization using the native IEEE 802.15.4 association primitives and a couple of IEEE 802.15.5 mesh sublayer commands. The network topology reflected in logic addresses is used as a guideline to tell towards which direction (rather than next hop) a packet should be relayed. The next hop is derived from each relaying node's local link state table. The routing scheme scales well with regard to various performance metrics. The ability to provide multiple paths also precludes the need for explicit route repair, which is the most complicated part in many wireless routing protocols.

For *simplicity*, IEEE 802.15.5 is tightly coupled with IEEE 802.15.4. As shown in Figure 1, it adds a thin mesh sub-

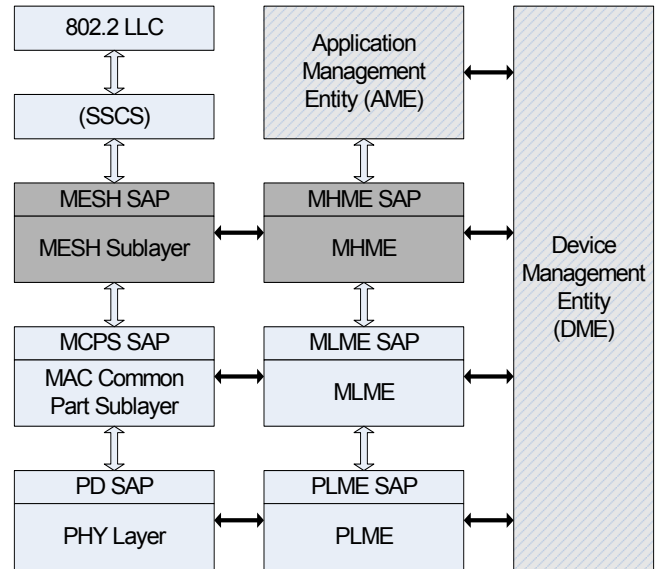


Fig. 1. Reference Model for IEEE 802.15.5

layer between the MAC sublayer and the service specific convergence sublayer (SSCS). It takes full advantage of IEEE 802.15.4 and maintains almost an identical set of service access points as the MAC sublayer. The basic mesh function is accomplished by using 802.15.4 primitives and a couple of mesh sublayer commands. For simplicity, the standard supports only a single PAN. Other enhanced functions like multicast, reliable broadcast, synchronized and asynchronous power saving, route tracing, and portability are done via a few other mesh sublayer primitives and commands. Compared with other mesh networking approaches, this cross-layer optimization approach not only simplifies the overall protocol stack but also makes it painless for upper layers to migrate from IEEE 802.15.4 one-hop star networks to IEEE 802.15.5 mesh networks. Users can expect most off-the-shelf IEEE 802.15.4 devices will be mesh-ready, at trivial or no additional complexity and cost.

The rest of this paper is organized as follows. Section II describes the basic mesh functions of IEEE 802.15.5 low rate WPAN mesh, including tree formation, address block, link state, and unicasting. Section III describes the enhanced functions such as multicasting, power saving, reliable broadcasting, and portability support. Section IV presents performance evaluations of the major functions. Section V concludes the paper.

II. BASIC MESH FUNCTIONS OF IEEE 802.15.5 LOW RATE WPAN MESH

The design principle of the mesh algorithms for IEEE 802.15.5 focuses on the scalability, which encompasses hierarchy, localization, and minimal overhead. Traditional reactive and proactive algorithms, for example AODV[11] and DSDV[12], fall short of being an efficient and scalable solution for large wireless sensor networks as it relies on network-wide broadcast for route discovery or link state exchange. Also, each node needs to store either routing table or link state information, which grows rapidly along with the network size. Algorithms embedding network hierarchy for a better

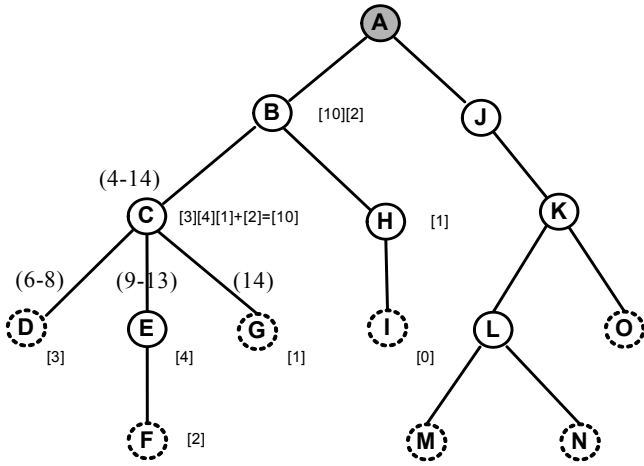


Fig. 2. Tree Formation and address assignment. [3] means the number of required addresses is 3, and (6-8) means that the address block is 6 to 8.

scalability have been introduced, making use of dominating sets and clusters as seen in OLSR[13]. However, the control overhead and energy consumption for the maintenance of link state table or routing table at each node still falters in resource-constrained wireless sensor networks. In addition, the special role of certain set of nodes (e.g., cluster head) can drain battery unevenly thus shortening the network life time. IEEE 802.15.5 builds in a hierarchy, that is, a tree-based local link state for mesh network. Tree structure provides a simple forwarding while the local link state provides alternative paths and optimized data forwarding. Importantly, the link state information exchanges are done locally, eliminating the overhead for network-wide broadcast and conserving the memory space of sensor nodes.

IEEE 802.15.5 comprises three mandatory basic functions: tree formation and addressing, local link state, and data forwarding/routing.

A. Tree formation and address block assignment

The tree formation starts with the first node in the network designating itself as the root and beginning to accept association requests from other nodes. After a node is successfully associated, it determines whether to become a parent node which allows other nodes to join the network through it. When a joining node receives multiple association responses, it should choose a proper node to join considering the tree depth or link quality (further details in the next section). After the tree reaches its bottom, that is, no more nodes are waiting to join the network, then, the address reporting process begins from leaf nodes. Leaf nodes will report to their individual parents two numbers: the number of child nodes and additional address space for all future use for example accepting new nodes. Upon receiving these two numbers from all its child nodes, a parent node calculates the same two numbers by summing up all requests from child nodes and its own, and reports to its parent. This process repeats until the root of the tree receives the information from all the branches, after which the root begins to assign addresses. Note that two-byte *short address* is used here. An address management policy is needed to control additional address requests. A simplistic

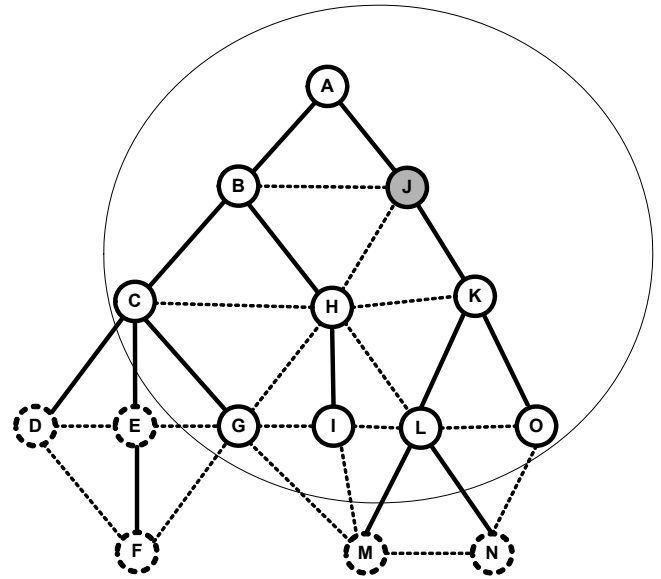


Fig. 3. The 2-hop neighbors of node J. The solid lines constitute a tree and dashed lines indicate mesh links. Both links form a meshed tree.

approach may be to allocate all the remnant addresses to all network nodes equally. Figure 2

shows an example of the address report and assignment. The address assignment takes a top-down procedure. The root will assign a block of consecutive addresses to each branch below it, taking into account the requested number of addresses. This procedure continues until the bottom of the tree is reached. After address assignment, a tree is formed and each node has an address table for tracking its branches. For example, in Figure 2, the node C has an address block (4-14) because the address blocks for three branches are (6-8), (9-13), (14), respectively. It adds the address block (4-5) to use the address 4 for itself and the address 5 reserved for potential expansion of the tree.

B. Local Link State(LLS) management

After tree formation, every node builds up its own local link state (LLS) information. The additional complexity of combining the block addressing with the LLS scheme is minimal. A node broadcasts several Hello messages to exchange link state information with its immediate neighbors when it receives an address block from its parent. The Hello message contains its own address block (begin and end address), tree level, and the address blocks of one hop neighbors. Upon receiving the Hello message, a node constructs 2-hop neighbor information including peer mesh nodes, optionally with respective link quality. Figure 3 shows the 2-hop neighbors of node J. Whenever a node receives the Hello message, it will update its link state information. One can extend this to *k-hop* ($k \geq 2$) neighbors but at extra costs. A node's LLS information contains two components: neighbor list and connectivity matrix. An example the neighbor list is shown in Table1: *begAddri* is the beginning address of the address block owned by neighbor *i*; *endAddri* is the ending address of the address block owned by neighbor *i*; *treeLeveli* means the tree level of neighbor *i*; *hopsi* is the number of hops from "this node" to neighbor *i*; *linkQualityi* describes the value of link

TABLE I
NEIGHBOR-LIST, EVERY ROW REPRESENT ONE NEIGHBOR NODE'S
INFORMATION.

<i>begAddr1</i>	<i>endAddr1</i>	<i>treeLevel1</i>	<i>hops1</i>	<i>linkQuality1</i>
<i>begAddr2</i>	<i>endAddr2</i>	<i>treeLevel2</i>	<i>hops2</i>	<i>linkQuality2</i>
...
<i>begAddrN</i>	<i>endAddrN</i>	<i>treeLevelN</i>	<i>hopsN</i>	<i>linkQualityN</i>

TABLE II
CONNECTIVITY MATRIX

	J	A	B	C	G	H	I	K	L	O
J		+	+	-	-	+	-	+	-	-
A			+	-	-	-	-	-	-	-
B				+	-	+	-	-	-	-
C					+	+	-	-	-	-
G						+	+	-	-	-
H							+	+	+	+
I								-	+	-
K									+	+
L										+
O										

quality such as *RSSI* or *LQI* from the underlying transceiver. Note that the link quality field is only for one hop neighbor for simplicity.

From the 2-hop neighbor information, a node can construct a connectivity matrix. We implement the connectivity matrix using a bitmap to conserve RAM space. For example, Table II illustrates the connectivity matrix for node J. The plus ('+') and minus ('-') at the cross cell of two nodes indicate they are or are not directly connected. The connectivity matrix shown here is symmetric because only bi-directional links are recorded. The matrix will become non-symmetry if one accounts asymmetric links.

C. Data Forwarding for Unicast

The data forwarding algorithm uses a divide-and-conquer approach. It first gets a *direction* towards the destination and performs a local search to choose the best next hop along the *direction*. The pseudo code in Figure 4 illustrates the procedure of data forwarding. Each node will calculate the next hop from its own address block and local link state information. It checks whether it has "any" information about the destination (category 1) or not (category 2). The category 1 includes the following three cases: the destination is (i) one of its descendants, (ii) one of its neighbors, (iii) one of its neighbors' descendants. For case (i) it just forwards the packet down to the destination along the tree path. For case (ii), if the destination is a 1-hop neighbor, it just forwards to the destination. If the destination is a 2-hop neighbor, it searches the connectivity matrix for the next hop node connecting the destination. For case (iii), an "anchor" node that can see the final destination as one of its descendants is to be found. Once the anchor node is found, forwarding a packet from the anchor node toward the final destination is the same as the case (i). The anchor node for the case (iii) is the node with the smallest address block (calculated by $endaddr - beginaddr$). The rationale is that the node with the smallest address block is the closest node to the final destination. If a relaying node does not have any clue about the destination due to the limited local view (e.g., category 2), the relaying node has to guess a

```

1: func_nexthop(dst)
2:   if(dst falls in address block of one of my neighbors or descendant)
3:     anchor = node with smallest address block
4:   else
5:     anchor = one of neighbors with smallest "tree level+hops"
6:   end if
7:   nexthop = Getonehopneighbor(anchor);
8: end func
9: func_Getonehopneighbor(anchor)
10:  current_hops = hop number of the anchor;
11:  while current_hops > 1
12:    for each neighbor nbi with a hop number of current_hops
13:      for each neighbor nbj directly connected to nbi
14:        hop_number of nbj = current_hops - 1;
15:      end for
16:    end for
17:    current_hops = current_hops - 1;
18:  end while
19:  return one of the neighbors with hop_number of 1;
20: end func

```

Fig. 4. Pseudo-code for unicast forwarding

next hop towards the destination. A simple scheme may be to forward the packet to the node's parent node toward the root node. We adopted a heuristics still making use of the neighbor table. The relaying node will examine its 2-hop neighbor table for each node's tree level and the number of hops from this relaying node. The node showing the minimum sum of the tree level and the number of hops from this node will be the "anchor" toward the final destination. The rationale is that if a node is closer to the root but far away from this node, it may not be a good anchor. After "anchor" has been chosen, a "breadth first" local search (e.g., examining all two hop nodes first) is used to find a next hop node to the chosen anchor node. Note there may be multiple nodes qualifying for the next hop choice, which, in fact, provides resilience for our data forwarding algorithm as shown by line 19 of Figure 4.

III. ENHANCED MESH FUNCTIONS

Basic mesh functions presented in Section II are essential to establish and operate a mesh network. For a variety of envisioned applications for wireless sensor networks, the basic mesh functions alone are not sufficient. In this section we present briefly the major enhanced functions that IEEE 802.15.5 is equipped with. Readers are encouraged to refer to the standard document for more details.

A. Multicast

The multicast protocol is designed tightly coupled with the unicast protocol and relies on both unicast logical tree and local link state information. Thanks to the existence of the unicast logical tree, the multicast protocol can build a shared, a multicast logical tree can be established without flooding network with join requests. In addition, the availability of the local link state information can improve the optimality of the multicast tree. Therefore, with very limited extra effort over unicast routing, this standard provides a very efficient and robust multicast protocol. By using the "Type" field, Multicast protocol reuses the entire two byte short addresses. Note here that the mapping from IP to IEEE 802.15.5 unicast/multicast addresses can be made with little effort.

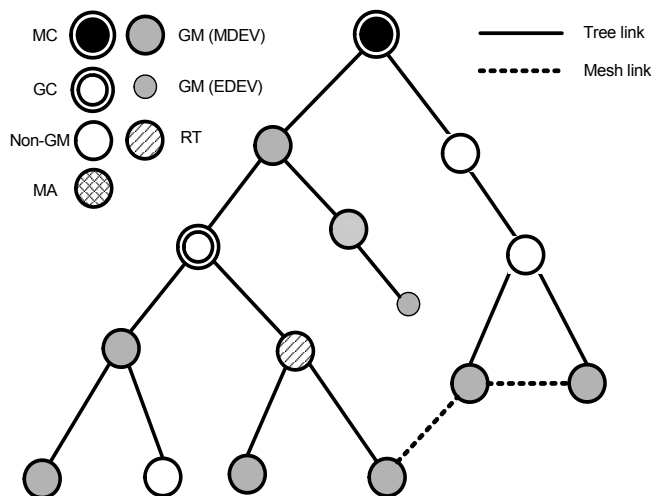


Fig. 5. entities of multicast protocol

Several logical entities necessary for the operation of multicast protocol have been defined as illustrated in Figure 5. The Mesh Coordinator (MC) is at the root of the unicast tree and keeps management information of all multicast groups such as group address and the address of Group Coordinator (GC). A Group Member (GM) is a device that actively participates (sending and receiving multicast frames) in a multicast group, while a Router (RT) is a device that relays multicast frames on a multicast tree but not a GM. A Group Coordinator (GC) is the controller of a multicast group. Each group should have its own GC and a device can be GC for multiple multicast groups. A Multicast Agent (MA) represents one or more of its child devices in a multicast group and handles all the routing functions for its child devices. A device can join a multicast group by sending a group join request to a closest GM in its neighbor list. If no GM is found in its neighbor list, it sends request to the GC. If it does not know the address of the GC, it sends the request to MC which will forward the request to GC. The device becomes GM after receiving a reply from a neighboring GM or the GC. A connection between the tree and the device is set by the reply. A routing table is not needed for multicast data forwarding. The decision of forwarding or not depends on the status of a device in a multicast group. When a device is a GM or RT of a multicast group, it will forward the packets for this group by broadcasting the packet at the MAC layer (one hop); otherwise, it will discard the packet. End devices (EDEV) being further resource constrained need a special consideration. They can only join and leave multicast groups through their parent devices. In this case, the parent devices act as their MAs.

B. Energy Saving

IEEE 802.15.4 provides a energy saving mechanism named a beacon mode. But it has shortcoming for mesh networks. Although the mode can be extended to multi-hop networks by creating a tree topology, each device has to receive a beacon from its parent device, and a data frame is exchanged only among devices within a parent-child relationship, which critically limits mesh functionalities.

IEEE 802.15.5 provides two energy saving algorithms to support mesh communication. Both methods are designed

based on the non-beacon mode. Since the mode does not provide any time structure, IEEE 802.15.5 defines a time structure to manage active and inactive times. Based on the method managing the active time, they are named Asynchronous Energy Saving (AES) and Synchronous Energy Saving (SES). AES is an algorithm utilizing an asynchronous time schedule for data transmission. With AES, each device defines its own active duration and inactive duration, and the time durations among devices are not synchronized. On the other hand, SES synchronizes active durations of all devices in the network. SES can support fast frame forwarding over multi-hops during the synchronized active time duration, but it requires the network-wide synchronization. Although two algorithms provide energy saving solutions for various networks, we focus on AES due to the limited space. AES algorithm is illustrated in Figure 6 and Figure 7. In AES, similar to a beacon frame in the beacon mode, a control frame named Wakeup Notification (WN) is transmitted, creating the asynchronous active duration. In order to transmit a frame to a peer device that may be in the inactive time duration, AES uses two different methods. For unicasting, a device uses a receiver oriented method. As presented in Figure 6, a transmitter (Dev B) waits for the active duration of the receiver (Dev A). After receiving a WN of the receiver, it transmits a frame within the active duration. In AES, the duration of active time can be flexibly configured by each device to receive multiple frames or to minimize energy consumption. If the remaining active duration is not long enough, a transmitter request additional time by transmitting an Extension Request (EREQ) frame. When it comes to broadcasting, a device uses a transmitter oriented mechanism. It wakes all neighbors up by transmitting an EREQ frames for longer time duration than a wakeup interval, then it broadcasts the data frame.

C. Reliable Broadcast

Reliable broadcast is necessary for WSNs and has many applications such as service discovery, tasking, and even data transfer. Recently, there have been several proposals to provide reliability in WSNs ranging from transport layer to link layer, which can be categorized into two approaches: negative acknowledgment (NAK) based[14][15] and positive acknowledgement ACK based[16]. In general, ACK schemes suffer from ACK implosion problem while NAK schemes cannot handle the case of all frames being lost at a particular node. Considering these, IEEE 802.15.5 adopts a scheme based on the randomization of ACK and the timer, named Timer-based Reliable Broadcast (TRB) scheme. The TRB schedules ACK transmission times to reduce ACK collisions and thus energy consumption. Moreover, receivers that need to send ACKs to the transmitter simply broadcast the received data frame without sending ACKs. The received data frame anyhow needs to be forwarded and it also can act as an ACK back to the transmitter (overhearing).

A node that broadcasts or relays broadcast frames maintains a Broadcast Transaction Table where a bitmap is used to indicate whether the frame is acknowledged for each neighbor. If any acknowledgments is missed from any neighbors after the RBCastTxTimer expires, the transmitter rebroadcasts.

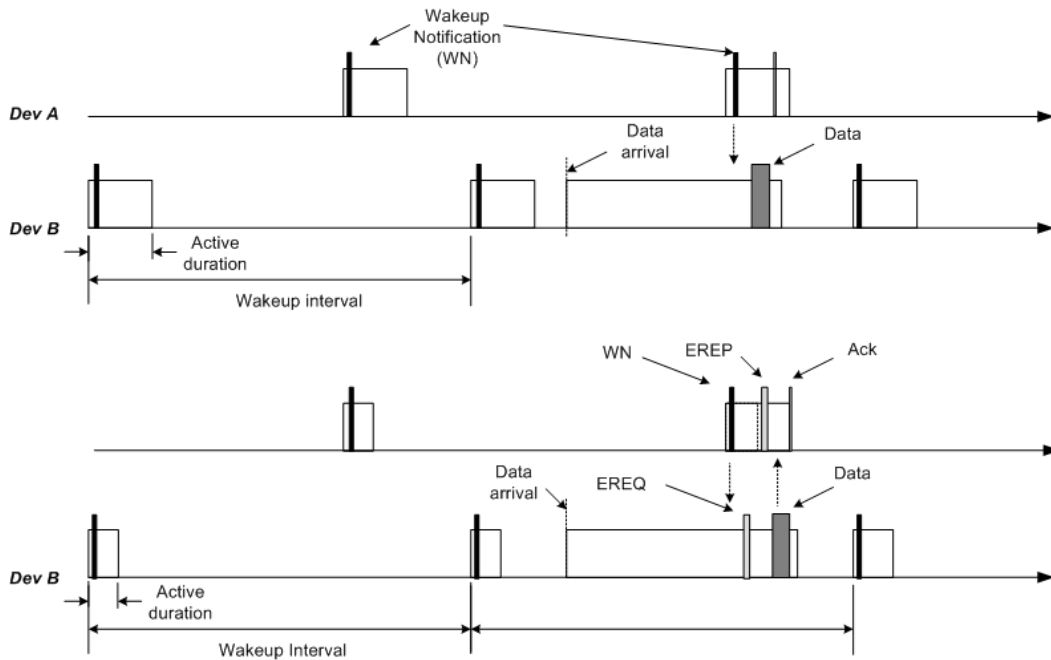


Fig. 6. Example Timeline of AES Unicast: device B Transmits a frame to device A

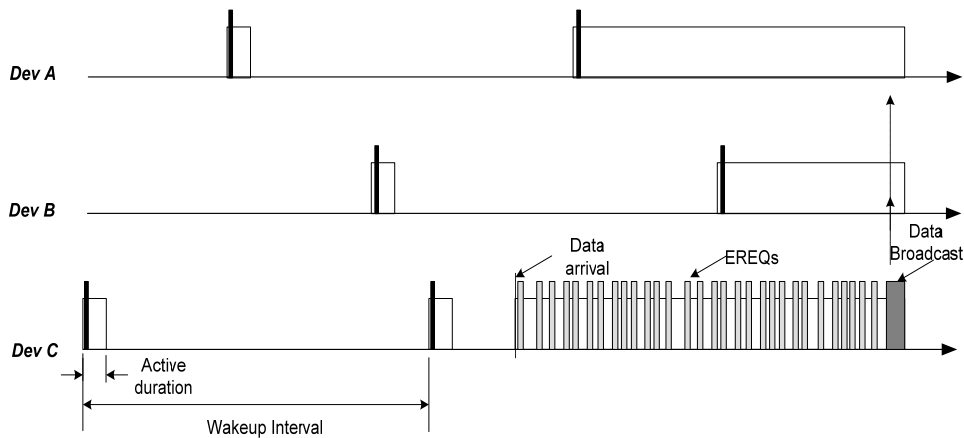


Fig. 7. Example Timeline of AES Unicast : device C transmits a broadcast frame

D. Portability support

While most of the applications for wireless sensor networks expect static network infrastructure, some applications require a portion of nodes mobile. For example, the sink or data collector in WSN may be carried around by a person or a car. IEEE 802.15.5 restricts the support of mobility to leaf nodes, therefore, the term "portability". Note here that the movement of nodes other than leaf nodes are not supported by the standard and will be regarded as node failure, which is taken care of by unicast routing.

The first step of portability support procedure is the detection of the movement of a portable node. Both the portable node and its current parent node can detect the movement from the transmission failures and the loss of periodic Hello message. Upon detecting a movement, the portable node rejoins the network. It is the same as the case of new node joining except it maintains the address of its previous parent. The portable node gets a new address from its new parent.

Upon successful rejoining of the network, the portable node informs its new address to its previous parent and relevant k-hop neighbors of previous parent. Previous parent and relevant k-hop neighbors update their neighbor tables and associated connectivity matrices so that they can re-route packets to the new address of the portable node and also inform the sender of the portable node's new address. Finally, the portable node will broadcast its hello message to let current neighbors update their own neighbor tables and connectivity matrices.

IV. PERFORMANCE EVALUATION OF MAJOR FUNCTIONS

This section discusses the performance issue of the major functions in IEEE 802.15.5 low rate part. Considering the rather broad scope of the paper and page budget, we will weigh more on the basic functions than enhanced ones. Basic functions are evaluated using a 50-node testbed deployed over a whole floor of CUNY engineering school. Since the testbed is not available to all the co-authors, some enhanced functions have been evaluated using a smaller setting.

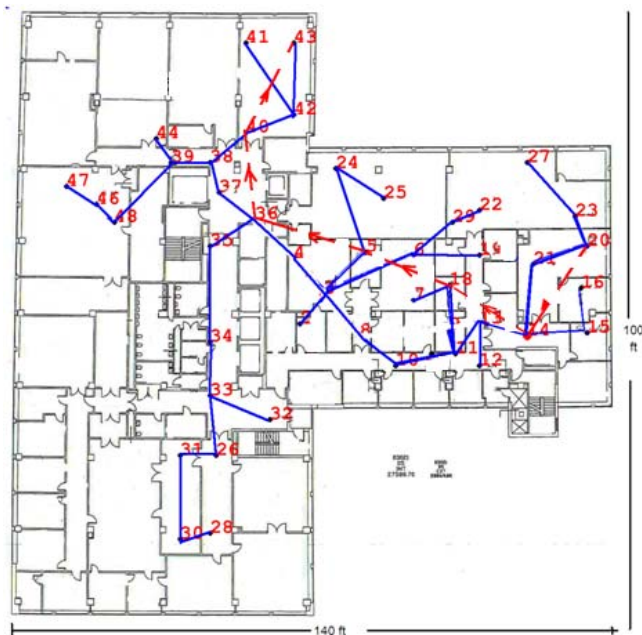


Fig. 8. Testbed: 50 nodes in the 5th floor of engineering building. Solid lines represent tree structure and dotted lines represent the routing path in one instance of experiments.

A. Basic Functions

A simulation study is reported in [17] to assess the performance of basic mesh functions. To further demonstrate the feasibility of the standard protocols in real applications, we implemented major functions (unicast and multicast) in a real testbed, consisting of 50 Micaz nodes each with an MIB600 gateway [18] deployed in the 5th floor of Engineering building as shown in Figure 8.

A common mesh layer software platform has been developed based on IEEE 802.15.4 PHY and MAC stack provided by TI CC2420[19]. The implementation of the algorithms is written in C and compiled with AVR-GCC. Control Network running TCP/IP protocols is built up to upload binary executables to exchange control messages and testing results between the sensor nodes and the server. The testbed starts as we upload our program binary code beginning from the root. Each node will wait for a time $T = N - i * \text{time_interval}$ and then reports the number of children to its parent. The process repeats until the root node collected the number of child nodes. N is a predetermined value chosen as 12 considering the tree depth of 50 nodes and i represents a tree level for each node. It takes about 30s to let all nodes join the tree and exchange mesh information. Finally, the experiment results can be displayed by the Java and PHP program running in server with Mysql database.

1) *Unicast routing*: The most fundamental part of the mesh lies in its ability to provide the end-to-end data delivery. Considering this, we have tested the unicast routing by using various scenarios and multiple performance measures. The first group of experiments is performed to show the end-to-end packet delivery ratio, defined as the number of successfully delivered packets divided by the number of packets generated at sources. In order to suppress the bias from a particular topology, we setup 5 different tree topologies starting with

different root nodes located in different rooms. For each tree structure, source and destination nodes pairs are selected randomly and final results are averaged. Every node keeps 2-hop local link state information. The data packet size including the IEEE 802.15.4 MAC and PHY is 59 bytes and the maximum MAC layer retransmission is three times. Figure 9(a) shows the packet delivery ratio versus the number of source/destination pairs with different offered loads. The packet delivery ratio maintains over 95% decreasing slightly as the number of nodes and the offered load increases. Analyzing data trace file obtained by the packet sniffer, we find that most packet losses are due to the MAC layer contention and unstable wireless lossy link between sensor devices especially the nodes around the elevator, for example, wireless links between node 4 and node 36.

In Figure 9(b), we examined the packet delivery ratio against different number of hops. Interestingly, increased number of hops between sources and destinations shows almost no negative effect on packet delivery ratio for the scenario tested. The higher the traffic loads, however, the lower the packet delivery ratio gets mainly due to the MAC layer contention. End-to-end delay performance is presented in Figure 9(c), showing increased delay along with the increased number of hops. The data transmission and the MAC layer retransmissions seem to be the main causes.

The second group of experiments considers a typical WSN scenario: multiple source nodes report data periodically to a sink node. Compared with the first experiment, the number of sources and offered traffic impact critically as shown in Figure 10(a). We observe a visible drop (around 10% compared with peer to peer case) in the packet delivery ratio when the number of sources exceeds 20 and data rate is 1 packet/second. This drop is due to IEEE 802.15.4 MAC contention, not the routing algorithm itself. Since the 20 chosen sources are 1-7 hops away from the sink, multiple MAC retransmissions are inevitable and amplified at links for example between node 4 and 36. The funneling effect around the sink node is observed in Figure 10(b). The packet delivery ratio degrades as the traffic increases, but shows a negligible influence by the hop distance. The reason is that regardless of the hop distance from the sink, all traffic generated by the sources will funnel to the sink. The end-to-end delay is almost the same as the first group of experiments, except for the case of 1 packet per second traffic in which the end-to-end packet delivery suffers more contention and thus more MAC layer retransmission.

2) *Local Link State Information*: As explained in section II, the data forwarding is optimized by the local link state information. We investigate the impact of the size (k -hop) of local link state information on the data forwarding decision. $k=0$ means that the data forwarding is purely done along the tree by parent and child relationship. $k=1,2$ means one-hop neighbor or two hop information. Every neighbor node information occupies 9 bytes in a neighbor table. The connectivity matrix varies according to the maximum number of nodes in the neighbor list and in our experiment it has two sizes: 16 bytes for neighbors up to 16 nodes or 32 bytes for neighbors up to 32 nodes. For $k=1$ we use 16 Bytes and 32 bytes for $k=2$. Table III shows the end-to-end delay, average number of neighbors, and average size of local link state

TABLE III
PERFORMANCE EVALUATION WITH VARYING LOCAL LINK STATE SIZES.

hop	End-to-end delay of unicast routing	Average number of neighbors	Average Local Link State Information size
K=0	33.74ms (7.8hops)	3.1	27.9 bytes
K=1	2.85ms (6.0 hops)	5.8	68.2bytes
K=2	7.98ms (2.75 hops)	12.5	144.5 bytes

information size regarding the nodes along the data path. With the help of local link state information, the data forwarding strategy has notable improvement. The path optimization is done in our case using the shortest path via higher LQI link. The overhead to accommodate more information is tens or hundreds of bytes, which is reasonable compared with 4K RAM size in Micaz node. Note here that we do not compare our algorithm with AODV because it is unfair to compare them when they belong to different categories (reactive versus proactive).

B. Multicasting

For the evaluation of the multicasting function, we use the same testbed as in the previous section. We vary the number of group members which are randomly chosen. Each point in Figure 11 is the average of five experiments with the corresponding number of group members. The overhead shown in Figure 11(a) is the number of bytes of 'group join request' and 'group join reply' packets that are transmitted. The overall join overhead peaks when the number of group members is 16. Each group member has to send a group join request and receive a group join reply back. Hence, more group join request and reply packets are generated along with increasing number of group members. On the other hand, as explained in Section III.A, a node can join directly to its neighboring group member. With 21 group members, a node has the higher probability of having a group member in its one-hop neighborhood than with 16 group member. Therefore, the overall join overhead with 21 members is lower than the one with 16 members. For the same reason, the join overhead per member shown in Figure 11(a) is decreasing as the number of the group member is increasing. Figure 11(b) illustrates the link cost of the multicast tree. The definition of the link cost[20] is calculated as $C = Lm/Lu$, where Lm is the average length of the multicast tree and Lu is the average length of unicast routing path. In the experiment, the length is measured as the number transmissions that are necessary to deliver a data from a group member to all other group members. Figure 11(b) show that the link cost is less than 0.5 when there are more than 8 group members. This indicates that multicast is more than twice efficient than unicast in such cases. If there are only two group members in the network, the multicast and the unicast is the same. Hence, the link cost is 1. Figure 11(c) shows the packet delivery ratio of multicast routing. Note that the multicast data is not acknowledged and there is no retransmission if any member misses the data. The packet delivery ratio is increasing as the number of group members is increasing. The reason is that a node can have many chances to listen to the same data if there are many group members in the neighborhood.

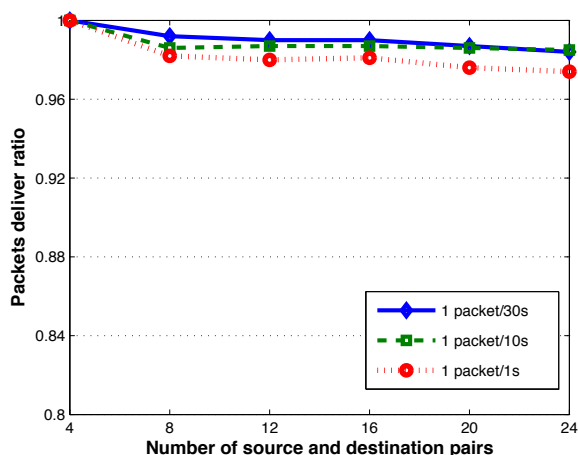
C. Energy Saving

In order to compare the performance of real systems, we have implemented AES with the same platform that CUNY testbed used. In practice, it is expected that the nodes around a sink node or aggregators in a sensor network will experience similar energy consumption pattern. We use the active ratio as the performance measure. The active ratio is defined as the measured active time for transmission, reception and idle listening divided by the total time spent.

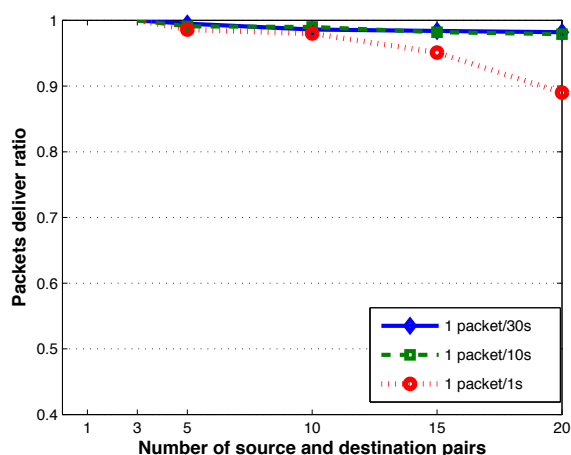
For comparison purpose, we also implemented two well-known asynchronous algorithms on our platform. The first, the long preamble emulation (LPE), is an emulation of BMAC[21]. The other, long preamble emulation with acknowledgement (LPEA), is an implementation of XMAC[22]. A star topology is created with one coordinator and there end devices. End devices transmit data frames to the coordinator with a uniform distribution. We consider different wakeup intervals ranging from 123 ms to 3.9s, which are from the standard when the beacon order is 3 to 8. We experimented for 800s at each condition. The results are compared in Fig 12 When the wakeup interval is under 0.5s, the energy consumption of LPE shows the best result. This is attributed to the minimum time duration to ensure the transmission activity. In LPEA and AES, a longer time is required to reserve the time for an ACK or an EREQ. However, the consumption of LPE rapidly increases when the wakeup interval increases. LPEA and AES show again very similar results. In both algorithms, a device waits for the receiver to wake up for the half of a wakeup interval on average. Note that devices with AES passively wait for the time duration. Thus, the time duration is not occupied for one transmission and can be used for data exchange between neighboring devices. It is interesting to observe that the active ratio versus wakeup interval shows a convex curve that attains the minimum at around 1 second.

V. CONCLUSION

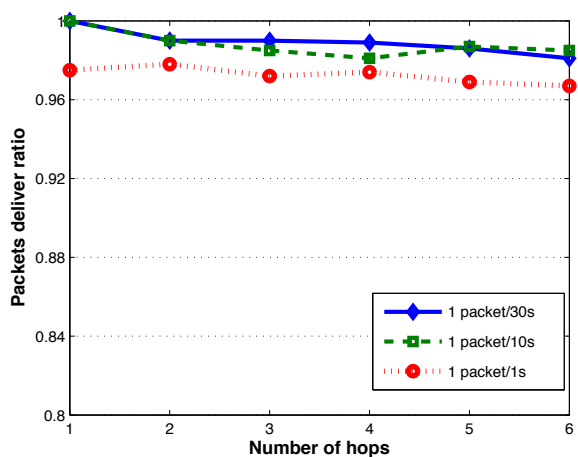
A new IEEE standard for wireless sensor networks, IEEE 802.15.5 WPAN Mesh (low-rate part), is introduced in this paper. The basic mesh functions comprise tree formation, allocation of address blocks, distributed local link state information, and mesh routing. The network formed from these basic building blocks exhibits many desirable properties such as scalability, simplicity, and reliability. In particular, the k-hop local link state information and tree structure with address block scheme facilitate the mesh routing by providing a guideline to extract the next hop toward a destination even when the local link state cannot give any information about the destination. Major enhanced functions are presented also: multicasting, energy saving, time synchronization for synchronous power saving, reliable broadcasting, and portability support. These enhanced functions will help the new standard find a broader market



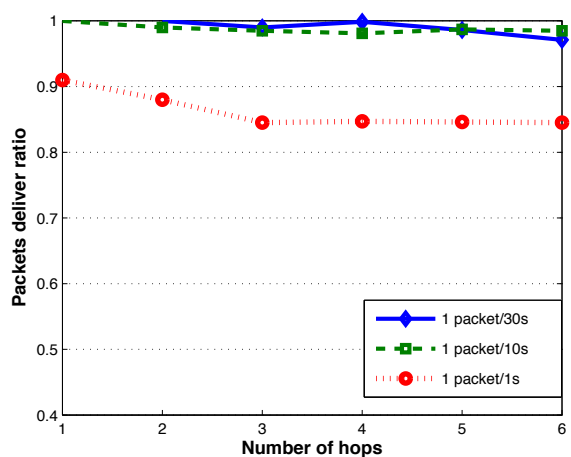
(a)



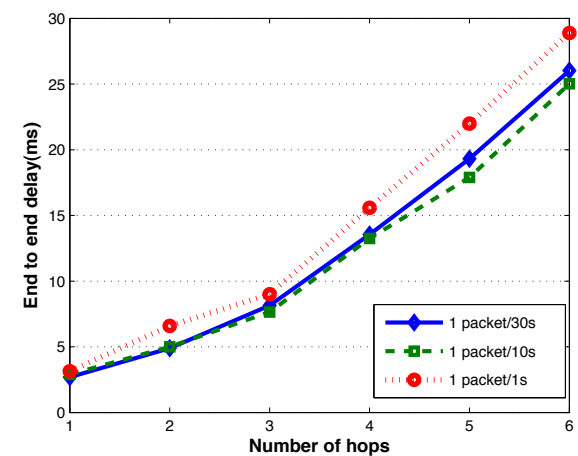
(a)



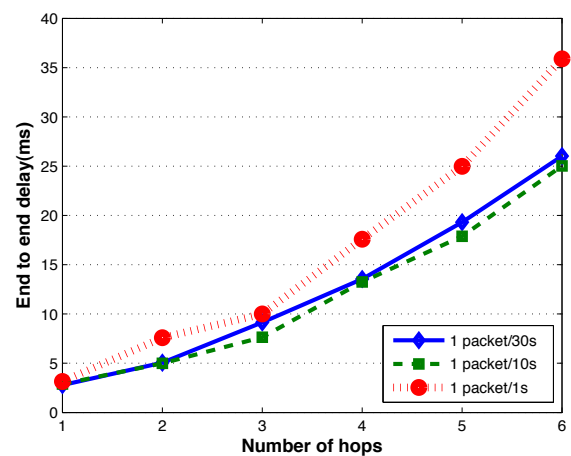
(b)



(b)



(c)



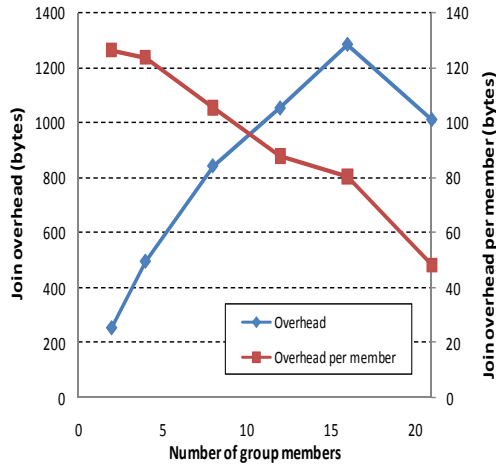
(c)

Fig. 9. (a): The packet deliver ratio vs number of sources/destination pairs. (b): Packet deliver ratio vs number of hops. (c): Average latency vs number of hops.

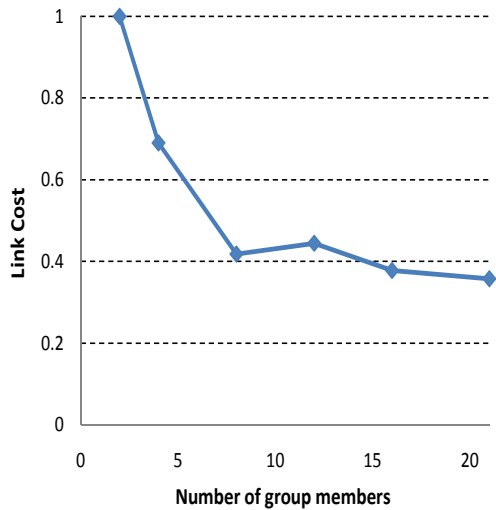
Fig. 10. (a): Packet deliver ratio in sources to sink scenario with respect to different number of sources nodes. (b): Packet deliver ratio vs the number of hops. (c): End-to-end latency vs different number of hops.

for wireless sensor networks. The testbed experimental results are very encouraging. The protocols demonstrate satisfactory performance against all the measures applied. For example, the packet delivery ratio maintains over 97% even when the offered traffic reaches 1 packet/sec in the case of multiple

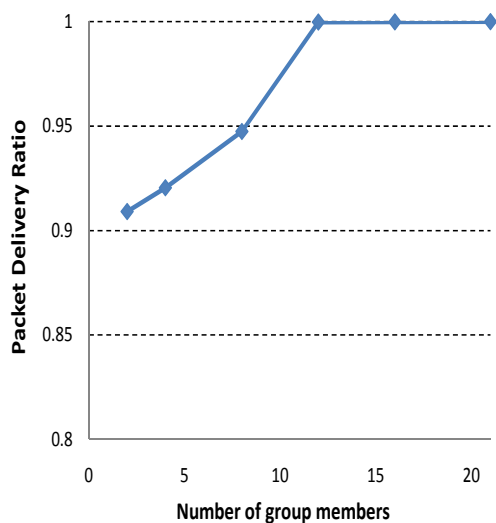
source-destination pairs and around 90% in multiple sources-to a sink style traffic. For multicast, the cost of a new node joining node decreases as the number of group members increases, as expected. Given the limited page budget, we strive to provide insights behind the protocol design to help



(a)



(b)



(c)

Fig. 11. (a): Signaling overhead for joining the multicast group. The number of group members (x-axis) includes the group coordinator and group members. (b): Link cost of the multicast tree. (c): Packet delivery ratio of multicast routing.

readers better understand the standard. With many desirable features demonstrated in this paper, we believe that IEEE

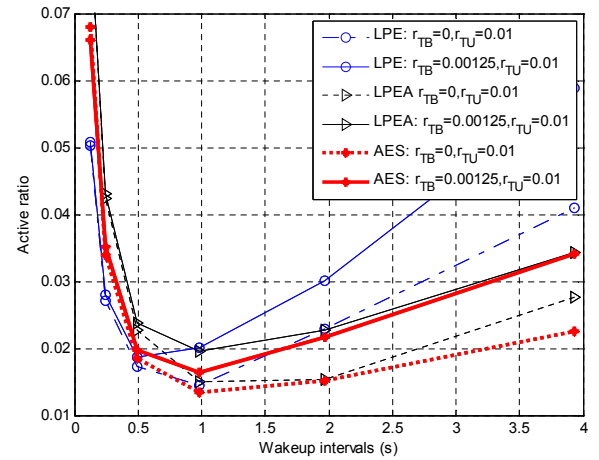


Fig. 12. Experimental comparison of active ratios. r_{TU} and r_{TB} stand for arrival rate for unicast and broadcast traffic respectively from a transmitter.

802.15.5 will help realize many envisioned applications for wireless sensor and control networks.

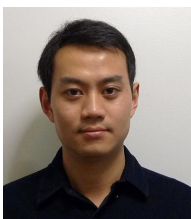
REFERENCES

- [1] *IEEE 802.15.4: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4, 2006.
- [2] *IEEE 802.15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.5, 2009.
- [3] Zigbee alliance. <http://www.zigbee.org/>.
- [4] I. D. Chakeres and L. Klien-Berndt, "AODVjr, AODV Simplified," *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, pp. 100–101, July 2002.
- [5] L. Hester, Y. Huang, O. Andric, A. Allen, and P. Chen, "neuron netform: A self-organizing wireless sensor network," in *Proc. 11th IEEE ICCCN Conference*, Miami, FL, USA, 2002, pp. 364–369.
- [6] J. H. G. Montenegro, N. Kushalnagar and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," *IETF RFC 4944*, September 2007.
- [7] J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams in 6LoWPAN Networks," *IETF Internet draft, draft-ietf-6lowpan-hc-06*, October 2009.
- [8] J. H. S. C. Z. Shelby, P. Thubert and E. Nordmark, "Neighbor Discovery for 6LoWPAN," *IETF Internet draft, draft-ietf-6lowpan-nd-03*, May 2009.
- [9] T. Winter and P. Thubert, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," *IETF Internet draft, draft-ietf-roll-rpl-04*, October 2009.
- [10] Hart communication foundation. <http://www.hartcomm.org/>.
- [11] C. E. Perkins, E. M. Belding-Royer, and I. D. Chakeres, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF RFC 3561*, July 2004.
- [12] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers," in *Proceedings of the ACM SIGCOMM*, London, UK, 1994, pp. 234–244.
- [13] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," *IETF RFC 3626*, October 2003.
- [14] R. S. S.-J. Park, R. Vedantham and I. F. Akyildiz, "Garuda: Achieving effective reliability for downstream communication in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 2, pp. 214–230, 2008.
- [15] A. T. C. C.-Y. Wan and L. Krishnamurthy, "Pump-slowly, fetch-quickly (psfq): A reliable transport protocol for sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 862–872, 2005.
- [16] K. Tang and M. Gerla, "Mac reliable broadcast in ad hoc networks," in *Proc. IEEE Military Communications Conference (Milcom'01)*, Mclean, VA, USA, 2001, pp. 1008–1013.
- [17] J. Zheng and M. J. Lee, "A resource-efficient and scalable wireless mesh routing protocol," *Special Issue of Elsevier Ad Hoc Networks Journal on Wireless Mesh Networks*, vol. 5, pp. 704–718, 2007.
- [18] Crossbow micaz motes. <http://www.xbow.com>.
- [19] Cc2420 datasheet. <http://www.chipcon.com>.
- [20] J. C.-I. Chuang and M. A. Sirbu, "Pricing multicast communication: A cost-based approach," in *Telecommunication Systems*, 1998, pp. 281–297.

- [21] J. H. J. Polastre and D. Culler, "Versatile low power media access for wireless sensor networks," in *2th ACM Conference on Embedded Networked Sensor Systems(SenSys 2004)*, Baltimore, MD, USA, 2004, pp. 95–107.
- [22] E. A. M. Buettner, G. V. Yee and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *4th ACM Conference on Embedded Networked Sensor Systems(SenSys 2006)*, Boulder, CO, USA, 2006, pp. 307–320.



Myung J. Lee (lee@ccny.cuny.edu) Dr. Myung Jong Lee received BS and MS from Seoul National University and Ph.D from Columbia University. He is currently a professor at the Dept of Electrical Engineering of CUNY. His recent research interests include wireless sensor/ad hoc networks, and CR networks. His publication is over 130 journals, book chapters, and conference papers and 25 U.S and International Patents (pending incl.). He is an associate editor for IEEE communications magazine and also actively participates in international standard organizations (the chair of IEEE 802.15.5 TG and a vice chair for ZigBee NWK WG). His group contributed NS-2 module for IEEE 802.15.4, a standard NS-2 distribution widely used for wireless sensor network researches.

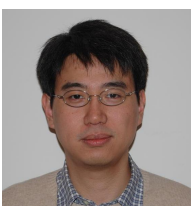


Rui Zhang (rzhang@ee.ccny.cuny.edu) Rui Zhang is a PhD candidate in Electrical Engineering in City University of New York (CUNY). He received B.S and M.S degrees from Xi'an Jiao Tong University and University of Electronic Science and Technology of China respectively in China, both in Electrical Engineering. His researches focus on wireless mesh network issues, including theoretical modeling and protocol design/implementation works. He is actively involved in IEEE standard activities and has several contributions to IEEE 802.15.5 and IEEE 802.15.4e standards. He is also responsible for IEEE 802.15.5 prototype implementation on testbed.



mesh routing protocol

Jianliang Zheng (jzheng99@yahoo.com) Jianliang Zheng received his Ph.D. degree in electrical engineering from CUNY in 2006. He has published 20 journal and conference papers with more than 450 citations. He also holds 6 U.S patents (some pending). His current research interests are wireless sensor networks, wireless mesh routing, and information security and management. He contributed the IEEE 802.15.4 LR-WPAN ns2 simulation module as well as the ZigBee network ns2 simulation module. His Topology-guided Distributed Link State (TDLS)



fiber networks. He is a member of the IEEE and the IEEE Communication Society.

Gahng-Seop Ahn (gahn@ccny.cuny.edu) Gahng-Seop Ahn received the Ph.D. degree in Electrical Engineering from Columbia University, New York, NY in 2009. He is a research assistant professor in the department of Electrical Engineering, the City College of New York and a member of the advanced wireless networking lab (AWNLI) and the Center for Information Networking and Telecommunications (CINT). His research interests includes Wireless sensor networks, body area networks, smart utility networks, cognitive radio networks, and radio-over-



the IEEE 802.15.5 LR-WPAN mesh standard. His research interests include Wireless Networking in general, and more specifically in the areas of Wireless Sensor Networks, Wireless Local Area Networks and Wireless Personal Area Networks.

Chunhui Zhu (c.zhu@samsung.com) Chunhui Zhu received his Ph.D. and M.S. degrees in Electrical Engineering from City University of New York, New York City, NY. He received the B.E. degree in Engineering of Telecommunications from Nanjing University of Posts and Telecommunications, Nanjing, China. He is currently a senior engineer at Samsung Electronics Corp., located in San Jose, CA. Prior to going to schools in the US, he was a senior/principal engineer at China Telecom, Xiamen, China. Chunhui Zhu was the technical editor of



networks. He is an active contributor to IEEE 802.11, 802.15 and ZigBee standard.

Tae Rim Park (taerim.park@samsung.com) Tae Rim Park received B.S. and M.S. from Inha University, Korea in 1998 and 2000, respectively, and Ph.D degree in electrical engineering and computer science from Seoul National University, Korea, in 2005. From 2005 to 2008, he was an assistant research professor with the Electrical Engineering Department of City University of New York (CUNY). Since 2008, he has been a senior research engineer with Samsung Electronics. He has been doing researches on embedded systems and wireless



Samsung Advanced Institute of Technology (SAIT) Korea. He also was a member of research staff of Electronic and Telecommunication Research Institute (ETRI), Korea. His research interests include Wireless Networking, Wireless Sensor Networks, and Wireless Personal Area Networks.

Sung Rae Cho (srcho@cau.ac.kr) Sungrae Cho received the Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology in 2002 and the B.S. and M.S. degrees in Electronics Engineering from Korea University, Korea. He is currently an associate professor in School of Computer Science and Engineering, Chung-Ang University (CAU), Korea. Prior to joining CAU, he was an assistant professor in Dept. of Computer Sciences, Georgia Southern University (GSU) until 2006 and a senior member of technical staff at



Chang Sub Shin received the B.S. in Computer Engineering from Kyungil University and, the M.S in Computer Engineering from Kyungpook National University, Korea in 2001. Since 2001, he has been a researcher in ETRI (Electronics and Telecommunications Research Institute), Korea. His research interests include MAC and Network protocols of WSN.



Jun Sun Ryu (junsun.ryu@gmail.com) Jun Sun Ryu received B.E. in Department of Electronics and Information Engineering from Seoul National University of Technology, Seoul, Korea, in 2006. He had developed ZigBee 2004, 2006, and pro Stack from 2005 to 2007. He is currently an M.S student at the City College of New York. His research interests include Wireless Sensor Network and Wireless Personal Area Network.