ISSN 2278 - 3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.5, Pages : 46-45 (2013)
Special Issue of ICETCSE 2013 - Held during October 21, 2013,Mysore, India

A Comparative Study of the Communication Architectures for Wireless Sensor Networks

Vaishnavi Y
Alcatel Lucent India Pvt Ltd, Bangalore, India
vaishnavi.y@alcatel-lucent.com

Narendran Rajagopalan
National Institute of Technology Puducherry, Karaikal, India
narenraj1@gmail.com

Abstract : The future is going to be a global network connected through wireless sensors which respond to queries and events registered. In this work, an attempt is made to compare two communication architectures for wireless sensor networks. RAP and BitCloud are two communication architectures that are compared and analyzed. This work highlights the advantages of the architectures and helps in making the right choice for specific applications.

Keywords: Communication Architecture, Wireless Sensor Network, BitCloud, Real-time communication.

INTRODUCTION

Wireless Networks [2] has evolved within a short span of time. A set of sensor nodes forming a cooperative network is known as Wireless Sensor Network (WSN). Some of the common resources forming a node are processing power with MCUs or CPUs, memory with flash and program memory, Radio Frequency transceiver, batteries as power sources and sensors and actuators for sensing. The nodes are self-organized in an ad-hoc fashion. Wireless Sensor Networks have varied applications which include; surveillance at difficult terrains, health monitoring with the aid of body network formed with many sensor networks which would monitor blood pressure, heartbeat rate, etc.

Due to the resource limitations of wireless sensors, the protocols that can be successfully used in other wireless networks cannot be applied for wireless sensor networks. Some of the important issues in Wireless Sensor Networks are as follows:

1. Designing an efficient Media Access Control protocol giving better throughput considering the limitations of the WSN. MAC protocol is responsible for deciding who will be given access to the medium at any point in time. There are contention based MAC protocols defined in the literature, such as Carrier Sense Multiple Access with Collision Avoidance(CSMA/CA). Since CSMA/CA uses binary exponential backoff, it might lead to latency and limited throughput. Hence specialized and customized MAC algorithms are needed for WSNs, performing better than B-MAC.
2. Routing is another challenge in WSNs. Dynamic Source Routing and Ad-hoc On-demand Distance Vector Routing algorithm are resource intensive routing algorithm for wireless networks which cannot be used in the case of WSN. Many factors like residual energy of a node, the link quality through a node and geometric position need to be considered to design an efficient routing algorithm.
3. Power saving in a wireless sensor can be achieved through efficient MAC algorithm with lesser collisions. The transceivers can be switched to lower power modes for power efficiency. Even with efficient routing protocols, energy efficiency can be achieved. But before forcing a sensor node to sleep, care should to be taken to ensure that there is always a path between the nodes for transmission.

ISSN 2278 - 3091
International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.5, Pages : 46-45 (2013)
Special Issue of ICETCSE 2013 - Held during October 21, 2013,Mysore, India

4. The process of determining the geographical location of a sensor node is called node localization. GPS cannot be used for this purpose since it would be expensive and also there would be power draining. Hence the tradeoff between the accuracy of the localization and the power utilization needs to be arrived at.
5. Clocks of the wireless sensor networks must be synchronized since many system and application tasks depend on it. The time duration between which synchronization must happen and the number of messages to be exchanged to achieve this has to be optimal for efficient performance.

Communication plays a significant role in a WSN. In this paper, an attempt is made to analyze and compare two different communication architectures, RAP – A real time communication architecture for Wireless Sensor Networks and Atmel's BitCloud architecture.

REAL TIME COMMUNICATION ARCHITECTURE (RAP) FOR WSN

RAP[1] provides high level query and event services for distributed micro sensing applications. The proposed network stack is light weight and uses location addressed communication model[4]. A new packet scheduling policy[3] called as velocity monotonic scheduling that can take into account distance and time constraints is used.

The motes in WSNs are real time embedded systems with data communication having timing constraints like end-to-end deadlines. Since motes are very small in size and with limited resources, the resources need to be utilized as efficiently as possible. The velocity monotonic scheduling assumes that each sensor mote knows its location with GPS or other location based services. RAP reduces the deadline miss ratio to 17.9%.

Fig.1 depicts the communication architecture of RAP protocol. RAP provides general purpose APIs which can be used for distributed micro-sensing and control in sensor networks. The application programs would interact with RAP through a set of Application Programming Interfaces provided by a Query/Event Service. The query/event service at the sensors send query results back to the base station (nodes with relatively more resources and energy). The wireless sensor network communication is supported by a network stack which includes transport layer Location Addressed Protocol (LAP)[10], Geographic Forwarding routing protocol[11,12], Velocity Monotonic Scheduling layer and a prioritized MAC. The network stack constituting a set of efficient algorithms reduces the end-to-end deadline miss ratio of the communication network[5,6]. Communication in a sensor network can be classified into two categories, local coordination and sensor base communication. Communication that takes place for the coordination in sensors located within a local geographical region is classified under local coordination. Reporting aggregated data to the base stations which can be multiple hops apart is classified under sensor base communication. Sensor-base communication uses location as the target address, so that any sensor in that region can respond to the query. Communication in sensor network may suffer from congestion also called as hot regions.

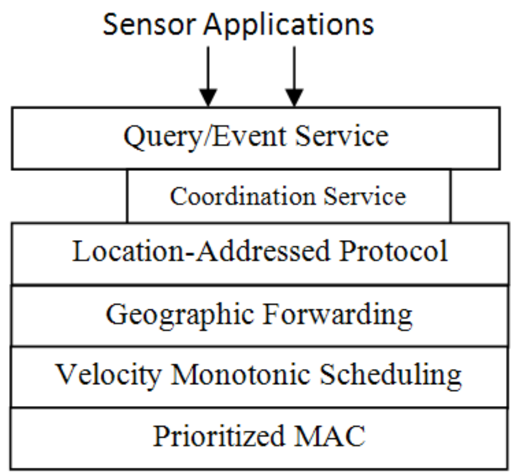

Fig 1: The Architecture of RAP communication (Courtesy [1])

The advantages of RAP include the following:
1. General purpose APIs are provided which are suited for distributed micro sensing and control in wireless sensor networks.
2. Number of packets meeting their end to end deadlines is maximized.
3. Scalability with large number of nodes and hops.
4. Reduce communication and process overhead.

Location Addressed Protocol (LAP)[1] is a connectionless transport layer in the network stack. LAP is similar to UDP except that all messages are addressed by location instead of IP address. Three types of communication are supported by LAP: unicast, area multicast, and area anycast.

1. Unicast delivers a message to a node that is closest to the destination location. Unicast can be used by sensors to send query results to base stations.
2. Area multicast delivers a message to every node in a specified area. Area multicast can be used to register for an event or send a query to an area, for coordination among nodes in a local group.
3. Area anycast delivers a message to at least one node in a specified area. Area anycast can also be used for sending a query to a node in an area. The node can initiate group formation and coordination in that area.

Geographic Forwarding (GF) makes a greedy decision to forward a packet to a neighbor if 1) it has the shortest geographic distance to the packet's destination among all immediate neighbors; and 2) it is closer to the destination than the forwarding node. When such nodes do not exist, the Greedy Perimeter Stateless Routing (GPSR)[9] protocol can be used to route packets around the perimeter of the void region. The only state on each node maintained by GF and GPSR is a table of the locations of immediate neighbors. Because GF uses immediate neighborhood information to make localized routing decisions, it is highly scalable with regard to the number of nodes, network diameter, and the rate of change in topology. GF works best in sensor networks that usually have high node densities and support location-addressed communication. Location addressed communication means that GF can be used without a location directory service, which introduces extra management and communication overhead. High node density causes two desirable properties of GF in sensor networks. First, the greedy forwarding algorithm described above has a high success probability in finding a good path from source to destination resulting in efficient communication. Second, the number of hops is approximately proportional to the distance that a packet has to travel. Hence, the distance between a node and a packet's destination can serve as an indication of the packet's hop count.

A key component of real-time communication architectures is the packet scheduling policy which determines the order in which incoming packets at a node are forwarded to an outgoing link. In the existing ad-Hoc networks, packets are typically forwarded in First Come First Serve (FCFS) order. FCFS scheduling does not work well in real-time networks where packets have different end-to-end deadlines and distance constraints. Instead, competing packets should be prioritized based on their local urgency. In the context of sensor networks, packet scheduling should be both deadline-aware and distance-aware. Deadline-aware means that a packet's priority should relate to its deadline. The shorter the deadline, the higher is the packet priority. Distance-aware means that a packet's priority should relate to its distance from the destination. The longer the distance the higher is the packet priority.

Since packet priority should be decided based on both distance and deadlines, Velocity Monotonic Scheduling (VMS) is proposed. VMS assigns the priority of a packet based on its requested velocity. A packet with a higher requested velocity is assigned a higher priority.

VMS improves the number of packets that meet their deadlines because it assigns the "right" priorities to packets based on their different urgencies on the current hop. VMS also solves the fairness problem described in sensor networks because packets that are far away from the base station will tend to have higher priorities when it competes against other packets that are closer to the destination. Here, two priority assignment policies are investigated: Static Velocity Monotonic (SVM) and Dynamic Velocity Monotonic (DVM), depending on whether the requested velocity of a packet is updated dynamically in intermediate nodes.

Each packet is assigned a priority based on its requested velocity and queued at the network layer when there are multiple outstanding packets. Several options are available for implementing priority queues. One approach is to insert all packets into a single queue ordered by priority. When the queue becomes full, higher priority incoming packets overwrite the lower priority ones. The benefit of this solution is that it accurately reflects the order of the requested velocities, and allows all the packets to share the same buffer regardless of their priority. The

ISSN 2278 - 3091
International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.5, Pages : 46-45 (2013)
Special Issue of ICETCSE 2013 - Held during October 21, 2013,Mysore, India

approach however, requires implementing a data structure whose insertion time, in the worst case, grows logarithmically in the number of packets. To bind the queue insertion overhead, another approach used, is to maintain multiple FIFO queues, with each queue corresponding to a fixed priority level. Each priority corresponds to a range of requested velocities. A packet is first mapped to a priority and then inserted into the FIFO queue that corresponds to its priority. This approach is more efficient because ordering need not be to be performed for every incoming packet. The per-packet overhead is logarithmic only with respect to the number of priority levels and not with respect to the number of packets. To further reduce the overhead, after each packet insertion in a priority queue, its requested velocity and priority is not updated until it reaches the next node.

Assuming that packets that miss their deadlines are useless, priority queues actively drop packets that have missed their deadlines to avoid the wastage of bandwidth.

Local prioritization at each individual node is not sufficient in wireless networks because packets from different senders can compete against each other for a shared radio communication channel. To enforce packet priorities, MAC protocols should provide distributed prioritization on packets from different nodes. Extensions of the IEEE 802.11 wireless LAN protocol have been investigated to provide distributed prioritization. Recently EDCF has been specified in the proposed 802.11e standard to provide different transmission priorities.

ATMEL BITCLOUD ARCHITECTURE FOR WSN

The Atmel BitCloud[7,8] internal architecture follows the suggested separation of the network stack into logical layers, as found in IEEE 802.15.4 and ZigBee. Besides the core stack containing protocol implementation, the BitCloud architecture contains additional layers implementing shared services (for example, task manager, security and power manager) and hardware abstractions (for example, hardware abstraction layer (HAL) and board support package (BSP)). The APIs contributed by these layers are outside the scope of core stack functionality. However, these essential additions to the set of APIs significantly help reduce application complexity and simplify integration. The BitCloud API Reference manual provides detailed information on all public APIs and their usage.

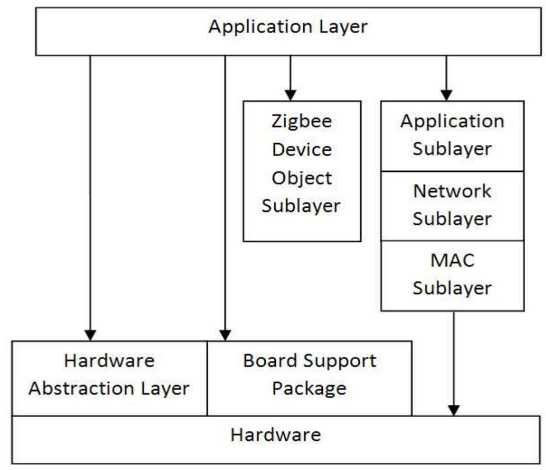

Fig 2: BitCloud software stack architecture.(Courtesy [7])

Fig. 2 depicts the BitCloud architecture. The topmost of the core stack layers, Application sub layer (APS), provides the highest level of networking-related API visible to the application. Zigbee Device Object (ZDO) provides a set of fully compliant ZigBee Device Object APIs, which enable main network management functionality (start, reset, formation, join). ZDO also implements ZigBee Device Profile commands, including Device Discovery and Service Discovery. There are three vertical service components responsible for configuration management, task management, and power down control. These services are available to the user application and may also be utilized by lower stack layers.

Configuration server (CS) is used to manage the configuration parameters provided with the Atmel BitCloud stack.

Task manager is the stack scheduler that mediates the use of the MCU among internal stack components and the user application. The task manager implements a priority based cooperative scheduler specifically tuned for the multi-layer stack environment and demands of time-critical network protocols.

Power management routines are responsible for gracefully shutting down all stack components and saving system state when preparing to sleep and restoring system state when waking up.

Hardware Abstraction Layer (HAL) includes a complete set of APIs for using on-module hardware resources (EEPROM, sleep, and watchdog timers) as

well as the reference drivers for rapid designing and smooth integration with a range of external peripherals (IRQ, TWI, SPI, USART, and 1-wire).
Board Support Package (BSP) includes a complete set of drivers for managing standard peripherals (sensors, UID chip, sliders, and buttons) placed on a development board.

CONCLUSION

This work is a study on different wireless sensor network architectures and their advantages. RAP architecture gives a theoretical perspective on the functional requirements on the communication architecture for supporting the functioning of wireless sensor networks. Atmel's bitCloud architecture is an implemented flexible architecture which is being used for application development successfully with good results.

REFERENCES

[1] Chenyang Lu , Brian M. Blum, Tarek F. Abdelzaher, John A. Stankovic, Tian He "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," *Eighth IEEE Conference on Real-Time and Embedded Technology and Applications Symposium, 2002.*

[2] I. Aad and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," IEEE INFOCOM 2001, Anchorage, Alaska, April 2001.

[3] M. Adamou, S. Khanna, I. Lee, I. Shin, S. Zhou, "Fair Real-time Traffic Scheduling over A Wireless LAN," Proceedings of the 22nd IEEE Real-Time Systems Symposium, RTSS 2001, London, UK, December 3-6, 2001

[4] S. Choi and K. G. Shin, "A Unified Wireless LAN Architecture for Real-Time and Non-Real-Time Communication Services," IEEE/ACM Transactions on Networking, 8(1), February 2000.

[5] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," Computer, 34(8), pp. 57-66, IEEE Computer Society Press, August 2001.

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Network Sensors," ASPLOS 2000.

[7] Atmel AVR2050: Atmel BitCloud, Developer Guide.

[8] Atmel AVR2052: Atmel BitCloud, Quick Start Guide.

[9] E. Altman, T. Basar, T. Jimenez, and N. Shimkin, "Competitive routing in networks with polynomial costs," IEEE Trans. Automat. Control, vol. 47, no. 1, pp. 92-96, 2002.

[10] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran, "Self-configuring localization systems: design and experimental evaluation," pp. 1-31, ACM TECS special Issue on Networked Embedded Computing, Aug. 2002.

[11] J. Cao and F. Zhang, "Optimal configuration in hierarchical network routing," Proc. Canadian Conf. Elect. and Comp. Eng., pp. 249-254, Canada 1999.

[12] T.-S. Chen, C.-Y. Chang, and J.-P. Sheu, "Efficient path-based multicast in wormhole-routed mesh networks," J. Sys. Architecture, vol. 46, pp. 919-930, 2000.