BIOINFO Sensor Networks
ISSN: 2249-944X & E-ISSN: 2249-9458 Volume 2, Issue 1, 2012, pp.-16-20.
Available online at http://www.bioinfo.in/contents.php?id=301

A CROSS-LAYERED INFORMATION SHARING PROTOCOL FOR WIRELESS SENSOR NETWORKS

WADILE K.A.[1], BHAMARE L.P.[1], PRERANA AUTADE[2] AND VAISHALI MANDHARE[3]

[1]Dept. of ELEX, Organization PREC Loni, Maharashtra M-tech RKDF, Bhopal, MP, India.
[2]Dept. of ETC, Organization PREC Loni, Maharashtra M.E.PREC, Loni (Pune), India.
[3]Dept. of IT, Organization PREC Loni, Maharashtra M. E. Comp. (BATU, Lonere), India.
*Corresponding Author: Email- autadeprerana@gmail.com

Received: February 21, 2012; Accepted: March 15, 2012

Abstract- Sensor networking has emerged as a promising tool for monitoring and actuating the devices of the physical world, employing self -organizing networks of battery-powered wireless sensors that can sense, process, and communicate. Cross-layer design and optimization is a new technique which can be used to design and improve the performance in both wireless and wire line networks. The central idea of cross-layer design is to optimize the control and exchange of information over two or more layers to achieve significant performance improvements by exploiting the interactions between various protocol layers. In this work, we undertake the task of developing and integrating a smart and power-aware protocol architecture. In this paper, we propose a new information sharing protocol architecture for sensor networks that can support existing protocols while simultaneously providing a platform for advanced cross-layer improvements. Our new architecture utilizes different services and data structures for providing information that can be shared among all layers of the protocol stack for increased network performance. This architecture has the advantage of maintaining the existing OSI layer structure while enhancing the performance of the network by providing a common framework for each protocol in the stack to access necessary information for protocol optimization .
Keywords- Architecture, Wireless Sensor Networks, Cross-layer.

Citation: Wadile K.A., et al. (2012) A Cross-Layered Information Sharing Protocol for Wireless Sensor Networks. BIOINFO Sensor Networks, ISSN: 2249-944X & E-ISSN: 2249-9458 Volume 2, Issue 1, pp.-16-20.

Introduction

Cross-layer designs have received much attention recently. While not as general as layered architectures, they prove to be more tunable and energy-efficient in many scenarios. This flexibility can be exploited by a middleware whose principal task is to adapt quality of service provided by the network to the application's needs using the pre-defined parameters of the cross-layer protocol. The ways in which a middleware (Milan) can control a cross-layer protocol for wireless sensor networks (DAPR), thereby ensuring that the network provides the application's required quality of service.

Smart environments exploit next generation technologies in the development of buildings, industries, homes, shipping, and transportation automation. These smart environments rely on sensory data from the real world. This sensory data is collected via sensor networks containing multiple wireless sensors in distributed locations.

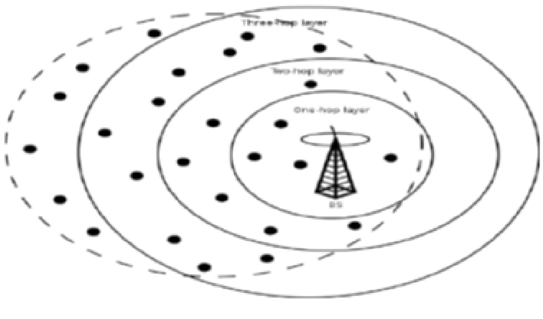

Fig .1-

Wireless sensors, commonly known as nodes, can be described as small self-powered electronic devices which are capable of communicating with each other or other wireless devices. The wireless sensor nodes use the Tiny Operating System (Tiny OS), which has been particularly designed for devices with hardware as well as software constrained requirements. Wireless sensor network differs from other wireless networks in several ways. First, it consists of physically small network nodes which perform sensing, processing, and then radio communications. Second, each node is configured with the same peer-to-peer networking protocol, thereby allowing a group of sensor nodes to form a self-configuring network. Third, the sensor nodes are energy constrained since they are designed to operate in specific areas for years with no maintenance . Depending on the specific application for which they are used, sensor networks can be further divided into *event-driven* sensor networks or *continuous monitoring* sensor networks. The main difference between an event-driven sensor network and a continuous monitoring sensor network is that in the former, the nodes remain in the sleep mode until some event occurs, as in the case of a sensor network devised for sensing forest fires. However, the main problem with this type of network is being able to switch nodes from a sleeping mode to a listening mode in a defined time. In contrast, in the latter case, sensor networks transmit data continuously from source nodes to sink nodes. One of the main and foremost problems faced by wireless sensor networks is that they are energy constrained, due to the fact that wireless sensor networks consist of sensor nodes which are battery operated, therefore it is impossible to recharge them, as they are intended to operate in specific areas for years with no maintenance. Hence, it is important to devise ways by which the energy efficiency of these sensor nodes can be increased so that the overall lifetime of the network is also improved. In this work, we explore a cross-layer technique that can achieve the goal of maximizing the energy efficiency in sensor networks. Cross-layer design is a new concept which has been devised for protocols of wireless networks such as ad hoc networks and sensor networks. More recently, a significant number of papers have proposed the use of cross-layer techniques in wireless sensor networks in order to achieve different objectives. Furthermore, it has been proven that cross-layer techniques help to improve energy conservation in wireless sensor networks. With cross-layer techniques, the different layers of the conventional Open System Interconnection (OSI) model interact with each other, irrespective of their positions in the model, to achieve a specific result. The traditional OSI layer architecture is modular in nature and has been implemented successfully in the case of wired networks. In the case of wireless sensor networks which have many constraints in terms of memory and energy, it becomes difficult to apply only the traditional protocol structure. Cross-layer designs have recently emerged as an effective approach and have been widely applied to wireless sensor networks. Constraints on energy, memory, storage resources, and low radio transmission capabilities of the wireless sensor nodes make cross-layer support more attractive.

Cross-layer schemes have advanced the idea that two or more layers can benefit from the same information; for instance, different decisions might be taken at the routing and transport levels based on the distance of the sensor to its next-hop neighbor. Many modern protocols share information across the protocol stack to ensure a coherent global design that prevents two different layers from causing opposite and unintended effects. To this end, the best way to provide information to all the layers is through a common framework. Yet this is not available today and thus protocol designers must resort to cross-layer designs that fuse layers in order to share information.

Clear advantages of cross-layer designs include the following: They provide a network and application specific response to the user's needs by closely adapting the protocol stack to the requirements and constraints of the deployment. They can greatly improve the network's performance and lifetime. They help rid the stack of unnecessary layers of the OSI model in some deployments. These advantages come at the cost of the following drawbacks: Cross-layer designs are hard to maintain and update, and they make it difficult to replace parts of the protocols. They require a coherent design for real gains and to avoid unforeseen and undesirable interactions. They cannot be ported to other applications since they lack generality. There are two main types of cross -layer improvements: Information Sharing, in which several layers share information, and *Layer* Fusion, in which operations from two or more layers are conducted jointly to optimize their output.

Methods and Materials

Cross-Layer Techniques Applied to Wireless Sensor Networks

A substantial amount of work has been done on cross-layer architecture in the last couple of years. A significant portion of such work has focused on cross-layer interaction between the MAC sub layer and the routing layer. Researchers focused on both power efficiency and scheduling, attempting to solve the problem of power efficiency and Quality of Service (QoS). Their proposed approach reduces the energy used to transmit and guarantee a certain level of bandwidth for the desired QoS. The main drawback of this approach is that it is a centralized approach where algorithms are executed by central agents having information of the network. Such an approach is more suited to wireless sensor networks where infrastructure support is available to the network. Researchers proposed a cross-layer protocol called the MAC-CROSS protocol that operates by exploiting the MAC and network layer information. The MAC-CROSS protocol is based on the S-MAC protocol. The main drawback of the S-MAC protocol is that the listening and the sleep periods are fixed. As a result, once their Network Allocation Vector (NAV) time expires, they wake up thereby wasting energy unnecessarily. The MAC-CROSS protocol overcomes this problem by allowing only nodes which actually take part in the communication to wake up and allowing the rest to be in the sleeping mode. Authors proposed a protocol called Latency and Energy aware MAC (LE-MAC) based on the cross-layer information obtained from the MAC and the network layer. The main aim of this protocol is to achieve energy efficiency and minimize latency. Authors propose a cross-layer design based on the MAC and the network layers. They proposed a Hierarchical Dynamic Source Routing (HDSR) protocol which is based on a modification of DSR. In the case of HDSR, the wireless sensor network is assumed to be divided into mobile nodes and forwarding nodes. The mobile nodes host the application, and the forwarding nodes actually route packets. A cross-layer technique has been proposed specifically for sensor networks with continuous monitoring capabilities. This approach attempts to improve power efficiency by removing collisions and idle listening (which wastes a lot of energy resources).Researchers propose a cross-layer design involving the MAC and the routing layers. Their design takes into account the energy losses which occur at the MAC layer during scheduling and during routing in the network layer, in order to improve power efficiency. In the same paper, the researchers propose algorithms known as Power Aware Random Scheduling (PARS) and Extended Power Aware Random Scheduling (EPARS). Finally, Safwat et al. proposed a cross-layer interaction involving the MAC sub layer, the routing layer, and the TCP layer to improve the throughput of ad hoc networks. They have proposed an extension to the DSR protocol at the routing layer where they have introduced two variables *β* and *β* is the number of times the DSR protocol will wait before invoking a route maintenance procedure if the MAC sub layer communicates that there was a communication failure, whereas is the variable with which the variable *β* compares its value before starting a route maintenance phase. Suppose that the value of *β* is set to 2 and the value of is always set to 1 before each communication when the source node starts communication with the destination node. If there is a communication failure, the message will be sent to the DSR protocol. The DSR protocol compares the value of with *β*. If is less than *β*, route maintenance is not initiated but if the value of is equal to or greater than *β*, the route maintenance phase is initiated. The objective of this approach is to minimize the routing overhead when it is not required. This is also the main goal of this work.

Media Access Control (MAC) Sub layer

Networks can be broadly classified into two types depending on the way they transmit data, namely, point-to-point networks or broadcast networks. In the case of point-to-point networks, a separate channel exists between two separate nodes. In contrast, in the case of a broadcast network, there is only one channel available which is shared by all nodes on the network. MAC protocols control access to this shared channel.

Network initialization is carried out on a common control channel. During the data transmission phase, the distributed TDMA receiver oriented channel (DTROC) assignment MAC protocol[3] is used. Each node is assigned a reception channel by BS, and the channel reuses is such that collisions are avoided. The node schedules transmission slots for all its neighbors and broadcasts the schedule. This enables collision-free transmission and saves energy, as nodes can turn off when they are not involved in a send/receive operation. The two steps of DTROC are channel allocation (the assignment of reception channels to the nodes) and channel scheduling (the sharing of the reception channel among the neighbors). DTROC avoids hidden terminal and exposed terminal problems by suitable channel allocation algorithms.

It is widely known that one of the main sources of energy wastage is due to the operation of the radio system. In turn, the operation of the radio is highly dependent on the medium access control (MAC) mechanism. Among the various paradigms, MAC based on Time Division Multiplexing Access (TDMA) principles have been recognized as one of the preferred alternatives. Various TDMA-based MAC proposals have already been reported in the literature.

Hidden Terminal Problem

Contention protocols (e.g., Carrier Sense Multiple Access (CSMA)) do not divide the channel into sub channels or pre assign them as in the case of scheduling protocols. However, with contention protocols, the common channel which is assigned on demand is shared by all nodes in the network (Mobile Ad Hoc Network (MANET) or ad hoc network in our case). The decision of which node should access the channel at a particular moment is handled by the contention mechanism. One of the main problems associated with contention protocols is the hidden terminal problem. This occurs when there are certain hidden terminals or a terminal within the signal range of the receiver but out of signal range of the transmitter. When node C transmits data to node B, node A which is out of transmission range of node C but within range of node B is unable to hear this transmission. As a result, node A, believing the channel to be free, starts its transmission to node B causing a collision. In this example, node A is the hidden terminal which results in collision and thus causes a loss of data or forces the nodes to resend the data. This, in turn, leads to energy inefficiency especially in ad hoc networking environments.

802.11 MAC Layer

The 802.11 standard is a family of IEEE standards for wireless Local Area Networks (LANs). The 802.11 protocols are also referred to as Wi-Fi. MAC sub layers perform the sensing operation on the channel by detecting the presence carrier signal on the channel with the help of the physical layer or by checking the value of the Network Allocation Vector. The process of using the Network Allocation Vector for channel sensing is known as virtual sensing as it does not carry out the actual physical signal detection on the channel. A node sets the value of the Network Allocation Vector on sensing a transmission between a source node and the destination node. If a source node wants to transmit data to the destination node, it sends a request to the destination node. When neighboring nodes of the source node overhear the request to send a packet, they set the value of the Network Allocation Vector. The destination node, on the other hand, upon receipt of the request to send from the source node sends a Clear to send to the source node. In this case, the neighbors of the destination node set the value of Network Allocation Vector.

Routing Layer

The routing layer carries out the task of delivering data, or more specifically, a packet from the source to the destination possibly across multiple networks. The routing layer ensures that the data reaches the destination. A good routing protocol can be described as one with the following properties: minimal routing overheads, routing should be done without creating loops, automatic load balancing, control of user over selection of routes, recovery from link failures, energy efficient, provides multicasting as well as QoS, and supports proper congestion avoidance mechanism.

DSR (Dynamic Source Routing Protocol)

DSR is an on-demand routing protocol that is initiated when a source node tries to send a packet to the destination node. The DSR protocol consists of five distinct phases: route discovery, route maintenance, data transfer, route caching, and route deletion. The operation of the DSR protocol can be summarized as follows.

The Route Cache

When a source node wants to transmit data to the destination node, it first checks DSR cache to see if there is a route available to the destination node. If a route from the source node to the destination already exists, then the source node starts sending data to the destination node. However, if there is no route information available for the destination node in the route cache, then the DSR protocol invokes the route discovery phase to establish a link between the source node and the destination node.

Route Discovery

The route discovery process is initiated by the DSR to find a suitable route between the source and the destination. This phase is accomplished by using a set of control packets called *Route Requests* and *Route Replies*. The Route Request consists of the destination address, the address of the source node, and a unique identifier. As soon as the source node sends control packets to the sending buffer in order to be broadcast, it initiates a timer called the *Send buffer Time Out* within which a response is expected. If the source node does not receive a reply within this timeframe, it will generate another route request packet using the exponential back off algorithm. When the neighboring node receives the Route Request from the source node, it checks whether it has already received a copy of this route request. If the receiving node has already received a copy of the route request, the current copy is discarded in order to avoid loops. However, if the receiving node has not received the same route request, it checks whether this route request was actually meant for it. If the destination address in the route request does not match the address of the receiving node, then the receiving node acts as a relay node and broadcasts the control packet to its neighboring nodes after making up an address entry of its own in the route request packet. There are two ways by which the receiving node can verify whether it has already received this route request or not. First, it can search its route table to check the entry of this particular route request packet. This is carried out by matching the unique identifier which is associated with each route reply to its entry in the route table. If an entry for this route request packet already exists in the case, the packet is thrown away. Second, the receiving node can check the entries in the route request packet to see whether it has already received the packet.

Route Maintenance

Due to the mobile nature of nodes in a sensor network, they may need to move from one place to another very often, making the current routes unusable. Therefore, the route maintenance process is an important phase in any routing protocol in a sensor network. When a source wants to communicate with the destination node, the MAC layer (802.11 in this case) sends a request to the destination node. If the destination node is still within the transmission range of the source node, it then sends clear to send to the source node, and the source node starts transmitting data to the destination node.

Proposed Cross-Layer Implementation

In this paper we proposed a cross-layer architecture using MAC and Routing layer. The cross layer architecture implemented in this paper is defined by the interaction of 802.11 MAC protocol and the Dynamic Source Routing protocol. The cross-layer architecture implemented in this work will be able to reduce routing overheads, by decrementing the route management process performed by the DSR protocol in most of the scenarios.

Extended MAC 802.11

In sensor networks, when a source sends a packet to its destination, the packet can get lost for a number of reasons: congestion due to high traffic, reduced signal strength due to mobility of network nodes or physical barriers, or power failure of nodes.

We have extended the 802.11 MAC layer for the sensor nodes to keep a record of the last received signal strength from each adjacent nodes. This record of the last signal strength helps to track whether the adjacent node remains within the transmission range or not. If the node is still within the transmission range and there is still a packet loss, then the loss may have occurred because of congestion. This information is passed to the upper layer where the DSR operates, and the route discovery process is not initiated. However, if there is a packet loss and the comparison with the last received signal strength shows that it is due to a link failure, then in this case DSR initiates a route discovery. We exploited Network Simulator-2 (NS-2) to implement the 802.11 MAC extension that stores the last signal strength received from all neighboring nodes. In addition, we also modified the 802.11 MAC layer to send a message to the upper layer in case there is a loss of communication but the destination node is still within the transmission range. In this case, the DSR algorithm does not initiate route discovery. When a packet is received by a node (from its neighboring node), the MAC layer extracts the ID of the sending node inside the received packet. This is done to determine whether a packet has been previously received from that node or not. If a packet from that node has not been received previously, a new entry is made for that particular node at the MAC layer, and the signal strength received from that node from the physical layer is stored. But, in case a packet has been received previously from that particular node, then it simply stores the current signal strength value received from that node. The received signal strength is used to find out whether the node is still within the transmission range. Once the signal strength has been used to determine whether the loss of communication is link related or not, an appropriate message needs to be sent to the upper layer. As explained earlier, the MAC layer transmits information with the use of a set of RTS and CTS messages. Once the RTS message has been sent, the sending node waits for the CTS message. If there is a problem in communication, the MAC layer waits for a certain period of time known as the back off period and retries later. After a certain number of unsuccessful attempts, the MAC layer informs the routing layer that the transmission was unsuccessful.

Extended DSR

The DSR protocol is initiated when a source node attempts to send a packet to the destination node whose IP address is or may be known to the source. As mentioned earlier, when a source node tries to communicate with the destination node (and after trying a certain number of times (at the MAC layer)), if it is unsuccessful, the DSR protocol assumes that the link is broken due to node mobility. It then triggers its route maintenance phase to search for alternative routes. However, the communication problem between the source node and the destination node could be because of other reasons (such as congestion) besides link failure. If the communication between the source and the destination breaks because of congestion, there is no need for DSR to initiate route maintenance because it creates unnecessary overheads on the node's energy resources which are what we are trying to minimize. Thus, as mentioned previously, we need to indicate to DSR whether the communication failure was due to a link failure or because of congestion. In our implementation, we extended the DSR protocol, and we pass such link failure/congestion information through a variable (called *xmit_reason* which holds the information on link failure or congestion between the source and the destination) from the MAC layer to the DSR routing layer. The variable *xmit_reason* is set to the value of *xmit_reason_high_strength* if the last received signal strength from the destination node shows that the signal strength received was equal to or greater than the threshold strength, indicating to the DSR protocol that the link is still intact. If this is not the case, then the DSR protocol initiates the route discovery/maintenance process to fix the route or to find an alternative route to the destination.

Future Work

Many improvements can be made to the implementation proposed in this research paper. One of these improvements is to extend our implementation to simulate the scenario when the last received signal strength from the destination node is greater than or equal to the threshold energy. When the source node wants to communicate with this node, but the node has already moved or has suffered power failure, a link failure will occur. Since our cross -layer architecture will only check the last received signal strength, the DSR will not invoke the route maintenance phase causing an infinite loop of maintenance messages (of what?). Another improvement we plan to investigate is to extend the cross -layer architecture proposed in this work to include TCP layer interactions.

The MAC layer used in the implementation is the 802.11 MAC layer which has its own limitations when it comes to deploying it in sensor networking environments because it is associated with a lot of overheads. Instead of the 802.11 MAC layer, we will investigate the use of Sensor MAC (SMAC) which is a MAC protocol designed for wireless sensor networks. SMAC has the potential to make the cross-layer architecture more energy efficient. Similarly, instead of using DSR in the architecture, more energy efficient routing protocols such as Hierarchical DSR (HDSR) can be used to improve energy efficiency.

Conclusion

Cross-layer design architecture aim at increasing the energy efficiency of sensor networks. Our proposed cross-layer architecture presented in this paper achieves the goal of improving energy efficiency by minimizing routing overheads. Our performance results obtained with the configurations used demonstrate that our cross-layer approach is effective in reducing unnecessary routing maintenance operations invoked by the DSR protocol when receiving nodes are within the transmission range of the transmitting node.

References

[1] Murthy S.R. and Manoj B.S. *Ad Hoc wireless networks Architecture and protocols by C*.

[2] Ortiz A.M., Olivares T., Royo F., Díaz P. and Orozco-Barbosa 1Albacete L. *Research Institute of Informatics University of Castilla-La Mancha* (UCLM) 02071-Albacete, Spain.

[3] Ian F. Akyildiz Mehmet C. Vuran O¨ zgu¨ r B. Akan, *A Cross-Layer Protocol for Wireless Sensor Networks*.

[4] Merlin C.J. and Heinzelman W.B. *Department of Electrical and Computer Engineering*, University of Rochester, Rochester NY fmerlin.