

Research Article

Cross-Layer Support for Energy Efficient Routing in Wireless Sensor Networks

N. Chilamkurti,¹ S. Zeadally,² A. Vasilakos,³ and V. Sharma¹

¹Department of Computer Science and Computer Engineering, La Trobe University, Melbourne 3086, Australia

²Department of Computer Science and Information Technology, University of the District of Columbia, 4200 Connecticut Avenue NW, Washington DC 20008, USA

³Department of Computer and Electrical Engineering, National Technical University of Athens (NTUA), Heroon Polytechniou 9, Athens 15780, Greece

Correspondence should be addressed to N. Chilamkurti, n.chilamkurti@latrobe.edu.au

Received 29 September 2008; Revised 10 January 2009; Accepted 18 February 2009

Recommended by Tommaso Melodia

The Dynamic Source Routing (DSR) algorithm computes a new route when packet loss occurs. DSR does not have an in-built mechanism to determine whether the packet loss was the result of congestion or node failure causing DSR to compute a new route. This leads to inefficient energy utilization when DSR is used in wireless sensor networks. In this work, we exploit cross-layer optimization techniques that extend DSR to improve its routing energy efficiency by minimizing the frequency of recomputed routes. Our proposed approach enables DSR to initiate a route discovery only when link failure occurs. We conducted extensive simulations to evaluate the performance of our proposed cross-layer DSR routing protocol. The simulation results obtained with our extended DSR routing protocol show that the frequency with which new routes are recomputed is 50% lower compared with the traditional DSR protocol. This improvement is attributed to the fact that, with our proposed cross-layer DSR, we distinguish between congestion and link failure conditions, and new routes are recalculated only for the latter.

Copyright © 2009 N. Chilamkurti et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

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. 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 [1, 2]. 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 [3] 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 [4]. 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 [5]. 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 [6]. 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 [7]. 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 [8].

The rest of the paper is organized as follows. We discuss related work in Section 2. In Sections 3 and 4, we present a brief overview of the Media Access Control (MAC) sublayer and the Dynamic Source Routing (DSR) protocol. In Section 5, we present our proposed cross-layer architecture that supports energy efficient routing in wireless sensor network environments. In Section 6, we present our performance evaluation results. We discuss future work in Section 7. Finally, we make some concluding remarks in Section 8.

2. Related Work

2.1. Routing Techniques. In the last decade, many energy efficient network routing protocols have been proposed for sensor networks. Previous routing schemes [9] involve direct communication protocols which facilitate direct communication between the source node and the base station. Therefore, for scenarios where the base station is quite a distance away from the source node, there is excessive usage of energy resources, ultimately resulting in a complete drainage of power. Such routing schemes are successful only where sensor nodes are near enough to the base station. In [10], the routing protocols proposed require each node to act as a router and route data from the source node to the base station through a set of selected intermediate nodes. These protocols differ from each other in terms of methodology and the algorithm used to select

the intermediate nodes which fall in the route between the source and the destination. The main disadvantage of these kinds of protocols is that sometimes nodes which lie near the station are included in the routing path, therefore causing their energy resources to drain off quickly, adversely affecting the overall power efficiency of the wireless sensor network.

In [11], a protocol is proposed for wireless sensor networks based on clustering technique, which involves dividing the nodes into a set of clusters each of which selects a cluster head. Rather than transmitting data directly from the node to the base station, each node transmits the data to the cluster head which, in turn, sends data to the base station. However, as pointed in [7], the cluster heads need to be selected based on a random rotation to make sure that the energy load is distributed evenly among all nodes. This selection leads to additional overheads.

2.2. 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 sublayer and the routing layer. In [12], 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.

In [13], the 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 [14]. 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. In [15], the 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.

In [16], the 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 [17]. 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. In [18], 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).

In [19], 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). The former exploits a pseudo random number generator to generate schedules to gain access to the wireless channel rather than using a set of request to send (RTS) and clear to send (CTS) packets which incur significant overhead. The latter (EPARS) saves energy by avoiding collisions (and thus retransmissions) and by exploiting energy-consuming control signaling. A major disadvantage of the proposed scheme presented in [19] is that it requires accurate time synchronization among neighboring nodes.

Finally, Safwat et al. [20] proposed a cross-layer interaction involving the MAC sublayer, 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 L . β is the number of times the DSR protocol will wait before invoking a route maintenance procedure if the MAC sublayer communicates that there was a communication failure, whereas L 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 L 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 L with β . If L is less than β , route maintenance is not initiated but if the value of L 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.

3. Media Access Control (MAC) Sublayer

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.

3.1. Hidden Terminal Problem. Contention protocols [21] (e.g., Carrier Sense Multiple Access (CSMA)) do not divide the channel into subchannels or preassign 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

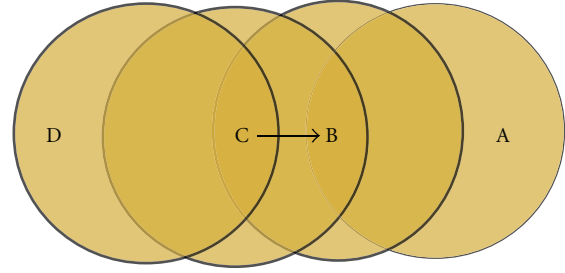


FIGURE 1: The hidden terminal problem.

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 [21]. 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 as illustrated in Figure 1. 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.

3.2. 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 [22]. MAC sublayers 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 [23]. 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 (Figure 2).

4. 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

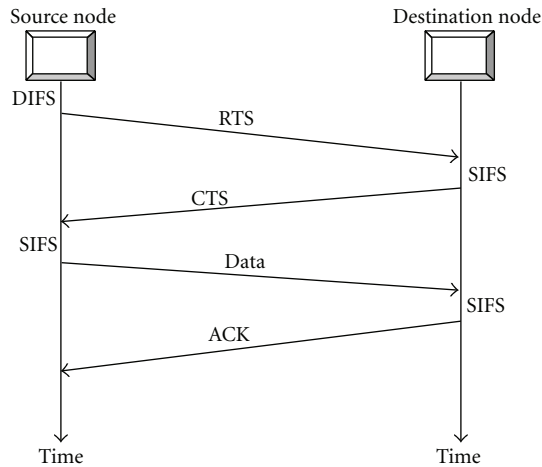


FIGURE 2: RTS/CTS Handshake.

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.

4.1. 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 *Sendbuffer 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 backoff 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.

Data Transfer. Once a route has been established between the source node and the destination node, the source node starts sending data.

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 [23].

5. 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.

5.1. 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. When a packet does not reach its destination, the DSR algorithm assumes that it is due to route failure and initiates a route search. This happens because the DSR does not have an in-built mechanism to determine whether this has been caused by congestion or node failure. However, if the packet loss has been caused by congestion, in this case, initiating a route discovery process actually ends up increasing network traffic, and worsens the congestion. The end result is that it increases the energy load on the nodes and decreases the

energy efficiency of the sensor nodes. Therefore, it would be beneficial if DSR could distinguish the packet loss caused by congestion versus the packet loss caused by node failure, and this is the main focus of this paper.

We have extended the 802.11 MAC layer (flow chart in Figure 3) 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) [24] 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 backoff period and retries later. After a certain number of unsuccessful attempts, the MAC layer informs the routing layer that the transmission was unsuccessful.

In our implementation, the information sent will inform the routing layer whether the destination node is still within the transmission range or not (by looking at the last received signal strength stored). If this last received signal strength reveals that the node is approachable, then the routing layer is informed not to initiate the route discovery process. Otherwise, a route maintenance step is initiated.

5.2. 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, as previously discussed, 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.

6. Performance Results

In this work we used Network Simulator-2 version 2.29 in all our performance evaluation tests. We used two configurations for our simulation experiments: the first configuration is a chain setup which is a static scenario where all nodes are aligned in the form of a line; the second configuration is an 11×11 grid setup. For both configurations, ftp traffic was generated for simulating TCP flows across the different nodes. The size of the TCP packet was set to 1024 bytes. Both simulations were set up to generate several routing faults. A routing fault is defined as the number of times the DSR protocol invokes the route maintenance process. The results were taken for both the extended DSR using our cross-layer approach exploiting MAC layer information and the normal DSR without using our cross-layer technique.

6.1. Chain Simulation Configuration. The chain configuration is a static scenario where 22 nodes have been setup in the form of a chain. Each node is situated at a distance of 150 meters from the adjacent node [25]. The transmission range for each node for this setup was set to 200 meters, whereas the carrier sense threshold was set to 400 meters. Two, four, and eight TCP flows were taken for the simulation where the numbers of hops were 4, 5, 6, 7, 8, 10, 12, 14, 16, and 20 respectively. The simulation time for this experiment was set to 2 minutes. Figures 4 and 5 show the simulation scenarios. The value of beta (where beta represents the growth rate of the TCP window W at every round-trip-time, we can say that beta represents the probing traffic of TCP) in the extended DSR was taken to be 2 [26].

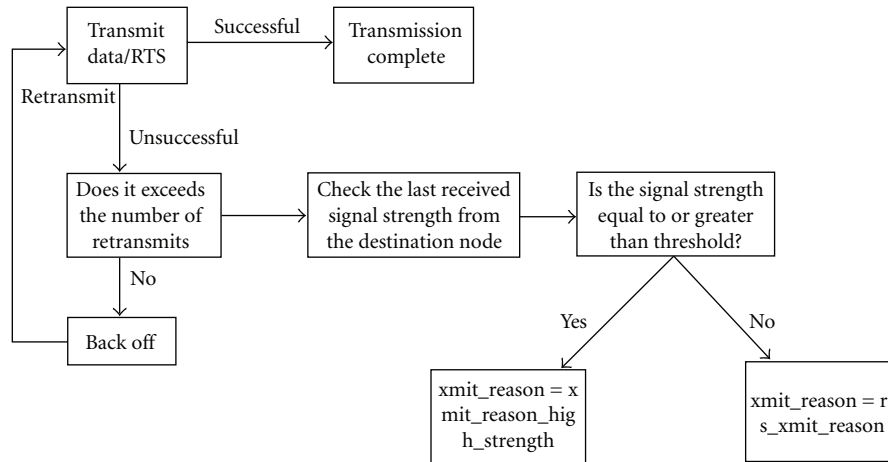


FIGURE 3: Flow chart for the extended MAC layer.

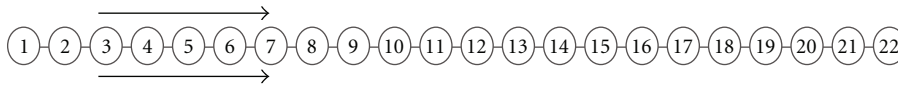


FIGURE 4: 2-parallel flows.

The simulation was executed for a period of 120 seconds, and the simulation results were obtained for the number of route errors that were generated using the normal MAC and DSR and for the extended MAC (Ex-MAC) and the extended DSR (Ex-DSR). The graphs which were generated for both simulations showed that the number of route errors that were generated in the normal DSR was much higher (almost double) than those which were generated in the extended DSR (Ex-DSR). The results clearly show that the normal DSR incorrectly assumed communication problems caused by congestion to be link-related errors and therefore performed the route maintenance operation. The chain scenario is a static scenario, therefore communication problems due to the mobility of nodes do not exist. But, the normal DSR has no way of finding out whether the communication problem between the source and the destination node is due to link failure or something else.

In the simulation experiment when the source node tries to communicate with the destination node and if communication was not possible (after a certain number of times) because of collisions, the MAC layer informs this to the DSR protocol. The normal DSR protocol assumes that it is a routing error caused by a link failure and initiates route maintenance with a set of route requests (RRQs) and route replies (RRPs). This, in fact, causes more traffic in the network and thus leads to more collisions further exacerbating the ongoing congestion. As a result, we have more route errors. On the other hand, with our proposed extended MAC layer and extended DSR approaches, when the source node tries to communicate with the destination node, after a certain number of attempts (backoffs), if communication is still not possible, the extended MAC checks the last received signal strength received from that particular node. If signal strength is equal to or greater than

the threshold strength, then the variable *xmit_reason* is set to *xmit_reason_high_strength* and is sent to the extended DSR. From this information, the Ex-DSR knows that the reason for communication errors is not because of link failure, and it checks the value of beta. If the value of beta is less than 2, in this case route maintenance is not initiated. Consequently, we experience no more additional congestion, and this ultimately leads to a lower number of routing faults compared to the normal DSR implementation (as shown in Figure 6).

6.2. 11 × 11 Grid Simulation Configuration. The 11 × 11 Grid Simulation configuration (as depicted in Figure 8) used a total of 121 nodes in the form of a matrix with eleven rows and an equal number of columns. Here again, each node is situated a distance of 150 meters from its adjacent node. The transmission range for each node for this setup was set to 200 meters, whereas the carrier sense threshold was set to 400 meters. The simulation time for this experiment was also set to 2 minutes. In this configuration, 11 nodes are situated in each row and column at a distance of 150 meters. The simulation was executed for a set of 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, and 22 flows. The set of 2 flows were introduced in such a way that their executions overlap with each other. Therefore, a TCP flow was induced from node 6 to node 116, and simultaneously another flow was induced perpendicularly from node 56 to node 66. The number of flows was gradually increased from 2, 4, and 6, and so on to a maximum of 22. Therefore, for 6 TCP flows, 3 flows were induced from node 6 to 116, 5 to 115, and 7 to 117, whereas the other three perpendicular flows were induced from node 45 to 55, 56 to 66, and 67 to 77. The flows were induced perpendicularly to each other to enhance collisions in the network. The tests conducted on the implemented

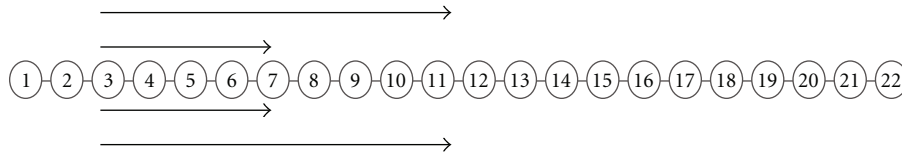


FIGURE 5: 4-parallel flows.

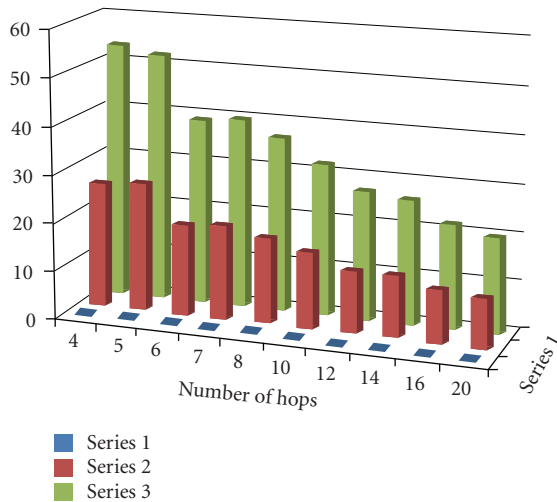


FIGURE 6: Routing faults in chain configuration. Series 1 = Routing Faults with Proposed Cross-Layer technique (extended MAC layer and extended DSR protocol). Series 2 = Routing Faults with $\beta = 2$. Series 3 = Routing Faults without our proposed Cross-Layer technique.

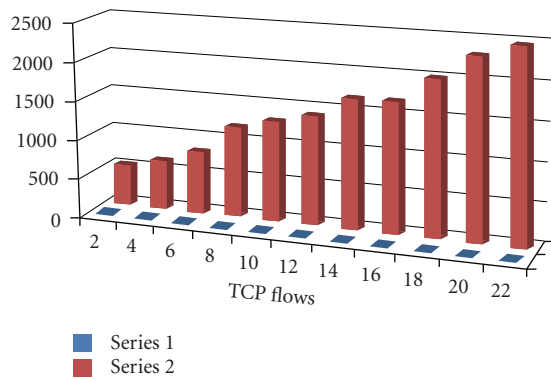


FIGURE 7: Routing faults in 11×11 grid configuration. Series 1 = Routing Faults Proposed Cross-Layer technique (extended MAC layer and extended DSR protocol). Series 2 = Routing Faults without our proposed Cross-Layer technique.

cross-layer architecture using the 11×11 grid configuration show that our proposed cross-layer approach does reduce the number of route faults (as illustrated in Figure 7). As a result, the DSR protocol does not invoke route maintenance when the routing error was caused by congestion.

The tests conducted on the static configuration verify the assumptions made during the design of our proposed cross-layer architecture. Since all nodes were static, the nodes

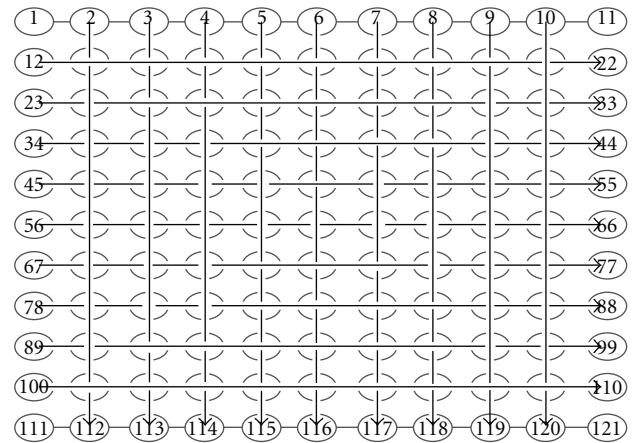


FIGURE 8: 11×11 grid configuration.

clearly showed that the routing faults were reduced to zero when the simulations were executed using our proposed cross layer architecture.

6.3. *Manhattan Configuration.* In this scenario, we used 100 nodes which were arranged in the form of a square of 500 meters by 500 meters. Six streets were defined in this square, three horizontal and three vertical, where each street consisted of two lanes. The nodes moved along these 12 lanes. Each node had a carrier sense threshold of 130 meters, and the transmission range was kept at 60 meters. The experiments were conducted for five different mobility scenarios. The speed of the nodes for the 5 scenarios was kept at 1 m/s, 2 m/s, 5 m/s, 10 m/s, and 20 m/s. The results obtained are shown in Figure 9.

Figure 9 shows the experimental results obtained for the Manhattan configuration where 20 TCP flows were induced for each scenario. The results showed that as the node mobility increases, so did the number of routing faults. However, in this configuration, the difference between the use of cross-layer and without the use of cross-layer is not as high as we have seen for the static configuration. The reason behind this is that for the mobility scenario, the nodes are actually moving causing the link failure. Therefore the cross-layer also incurs a number of routing faults. However, our cross-layer technique was still able to differentiate between a link failure caused by node mobility from the one caused by congestion. Therefore, it was still able to minimize the routing faults as there was no need to induce a route maintenance procedure where the communication error was not due to mobility but due to congestion.

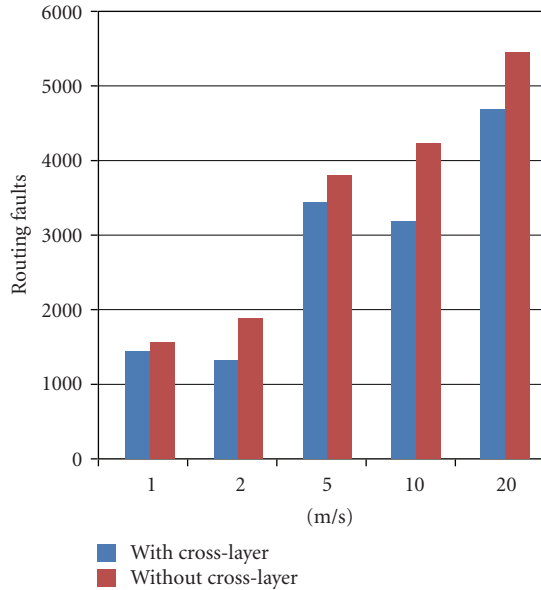


FIGURE 9: Number of routing faults for various mobility scenarios (at various speeds) with and without our proposed cross-layer implementation.

7. 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.

8. Conclusions

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] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, 2005.
- [2] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.
- [3] M. Horton and J. Suh, "A vision for wireless sensor networks," in *Proceedings of IEEE MTT-S International Microwave Symposium Digest (MWSYM '05)*, pp. 361–364, Long Beach, Calif, USA, June 2005.
- [4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS '00)*, vol. 2, pp. 3005–3014, Maui, Hawaii, USA, January 2000.
- [5] A. Dunkels, T. Voigt, and J. Alonso, "Making TCP/IP viable for wireless sensor networks," in *Proceedings of the 1st European Workshop on Wireless Sensor Networks (EWSN '04)*, Berlin, Germany, January 2004.
- [6] A. J. Goldsmith and S. B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 8–27, 2002.
- [7] I. Hakala and M. Tikkakoski, "From vertical to horizontal architecture: a cross-layer implementation in a sensor network node," in *Proceedings of the 1st International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense '06)*, Nice, France, May 2006.
- [8] T. Melodia, M. C. Vuran, and D. Pompili, "The state of the art in cross-layer design for wireless sensor networks," in *Wireless Systems and Network Architectures in Next Generation Internet*, vol. 3883 of *Lecture Notes in Computer Science*, pp. 78–92, Springer, Berlin, Germany, 2006.
- [9] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.
- [10] S. Dai, X. Jing, and L. Li, "Research and analysis on routing protocols for wireless sensor networks," in *Proceedings of the International Conference on Communications, Circuits and Systems*, vol. 1, pp. 407–411, Hong Kong, May 2005.
- [11] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 22–31, Atlanta, Ga, USA, September 2002.
- [12] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multicast in multi-hop wireless networks," in *Proceedings of the 1st International Conference on Wireless Internet (WICON '05)*, pp. 47–54, Budapest, Hungary, July 2005.

- [13] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 108–121, Baltimore, Md, USA, November 2004.
- [14] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1567–1576, New York, NY, USA, June 2002.
- [15] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas, "Cross-layer design for power efficiency and QoS provisioning in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3306–3315, 2006.
- [16] C. Suh, D. M. Shrestha, and Y.-B. Ko, "An energy-efficient MAC protocol for delay-sensitive wireless sensor networks," in *Proceedings of the Embedded and Ubiquitous Computing Workshops (EUC '06)*, vol. 4097 of *Lecture Notes in Computer Science*, pp. 445–454, Seoul, Korea, August 2006.
- [17] M. Tarique, K. E. Tepe, and M. Naserian, "A cross-layer design for passive forwarding node selection in wireless ad hoc networks," in *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, pp. 802–807, Maui, Hawaii, USA, June 2005.
- [18] R. Hashim, Q. Nasir, and S. Harous, "Adaptive multi-path QoS aware dynamic source routing protocol for mobile ad-hoc network," in *Proceedings of the Innovations in Information Technology (IIT '06)*, pp. 1–5, Dubai, UAE, November 2006.
- [19] B. DeCleene, V. Firoiu, M. Dorsch, and S. Zabele, "Cross-layer protocols for energy-efficient wireless sensor networking," in *Proceedings of IEEE Military Communications Conference (MILCOM '05)*, vol. 3, pp. 1477–1484, Atlantic City, NJ, USA, October 2005.
- [20] A. Safwat, H. Hassanein, and H. Mouftah, "Optimal cross-layer designs for energy-efficient wireless ad hoc and sensor networks," in *Proceedings of IEEE International Performance, Computing and Communications Conference*, pp. 123–128, Phoenix, Ariz, USA, April 2003.
- [21] B. A. Forouzan and S. Fegan, *Data Communications and Networking*, McGraw-Hill, New York, NY, USA, 3rd edition, 2003.
- [22] G. M. Lundy, M. Almquist, and T. Oruk, "Specification, verification and simulation of a wireless LAN protocol: MACAW," in *Proceedings of IEEE Military Communications Conference (MILCOM '98)*, vol. 2, pp. 565–569, Bedford, Mass, USA, October 1998.
- [23] T.-C. Tsai and C.-M. Tu, "An adaptive IEEE 802.11 MAC in multihop wireless ad hoc networks considering large interference range," in *Wireless On-Demand Network Systems*, vol. 2928 of *Lecture Notes in Computer Science*, pp. 97–100, Springer, Berlin, Germany, 2003.
- [24] NS-2 version 2.28, <http://www.isi.edu/nsnam/ns>.
- [25] X. Yu, "Improving TCP performance over mobile ad hoc networks by exploiting cross-layer information awareness," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)*, pp. 231–244, Philadelphia, Pa, USA, September–October 2004.
- [26] K. Nahm, A. Helmy, and C.-C. Jay Kuo, "TCP over multihop 802.11 networks: issues and performance enhancement," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, pp. 277–287, Urbana-Champaign, Ill, USA, May 2005.