Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

A Wireless Sensor Network Cross-Layer Protocol

Dr. B.Purna Satyanarayana[1], Ms.Ananya Preeti Padma[2]*
[1] Professor Dept. Of Computer Science and Engineering, NIT , BBSR*
[2] Assistant Professor,Dept. Of Computer Science and Engineering, NIT , BBSR
bpurnasatyanarayana@thenalanda.com ananyapriti@thenalanda.com*

***Abstract*— Wireless sensor networks must use energy-efficient communication protocols to achieve their application goals due to the severe energy constraints placed on battery-powered sensor nodes (WSN). The vast majority of currently available solutions, however, are built on the traditional layered protocols approach. For sensor nodes with limited resources, a unified strategy that combines common protocol layer functionalities into a cross-layer module is substantially more resource-efficient. To the best of our knowledge, there is currently no unified cross-layer communication protocol for WSNs that takes into account transport, routing, and medium access functionalities with physical layer (wireless channel) effects for effective and dependable event communication.**

A uniform cross-layer protocol is created in this paper to replace the entire conventional layered protocol architecture now in use in WSNs. As both the information and the functionalities of conventional communication layers are combined into a single protocol, our design principle is total unified cross-layering. The proposed cross-layer protocol's goals include energy-efficient, highly reliable communication with adaptive communication decisions and local congestion avoidance. To do this, the novel idea of initiative determination governs protocol operation. In order to accomplish effective and dependable communication in WSN, the cross-layer protocol implements received-based contention, local congestion control, and distributed duty cycle operation. Results of the performance study demonstrate that the suggested cross-layer protocol performs better than the conventional layered protocol architectures and greatly increases communication efficiency.*Index Terms*— Cross-Layer Protocol, Congestion Control, Routing, Medium Access Control, Wireless Sensor Networks.

I. INTRODUCTION

WRELESS sensor networks (WSN) are event-based systems that exploit the collective effort of densely deployed microsensor nodes which continuously observe certain physical phenomenon. In general, the main objective of any WSN application is to reliably detect/estimate event features from the collective information provided by sensor nodes. Nevertheless, the main challenge for achieving this objective is mainly posed by the severe energy and processing constraints of low-end wireless sensor nodes.

Clearly, the collaborative sensing notion of the WSN achieved by the networked deployment of sensor nodes help to overcome the characteristic challenge of WSN, i.e., resource constraints. To this end, there has been significant amount of research effort that aims to develop networking protocols in order to achieve communication with maximum energy efficiency.

In addition to the collaborative sensing and networking in WSN, spatio-temporal correlation is another significant characteristic of sensor networks. Dense deployment of sensor nodes makes the sensor observations highly correlated in the space domain with the degree of correlation increasing with internode proximity. Similarly, some of WSN applications such as event tracking require sensor nodes to periodically sample and communicate the sensed event features, which yields temporal correlation between each consecutive observation of a sensor node. It has been shown in [8] that exploiting the spatial and temporal correlation further improves energy- efficiency of communication in WSN.

Most of the proposed communication protocols exploiting the collaborative nature of WSN and its correlation characteristics improve energy efficiency to a certain extent. However, the main commonality of these protocols is that they follow the traditional layered protocol architectures. More specifically, the majority of these communication protocols are individually developed for different networking layers, i.e., transport, network, medium access control (MAC), and physical layers. While these protocols may achieve very high performance in terms of the metrics related to each of these individual layers, they are not jointly optimized in order to maximize the overall network performance while minimizing the energy expenditure. Considering the scarce energy and processing resources of WSN, joint optimization and design of networking layers, i.e., cross-layer design, stands as the most promising alternative to inefficient traditional layered protocol architectures.

In fact, recent work on WSN [2], [9] reveal that cross- layer integration and design techniques result in significant improvement in terms of energy conservation in WSN. There exists some research on the cross-layer interaction and design in developing new communication protocols [4]. However, as discussed in [4] in detail, these works either provide analytical results without any communication protocol design, or perform pairwise cross-layer design within limited scope, e.g., only routing and MAC layers, which do not consider all of the networking layers involving in the communication in WSN such as transport, routing, medium access and physical layers. Clearly, there is still much to be gained by rethinking the protocol functions of network layers in a unified way so as to provide a single communication module for efficient communication in WSN. To the best of our knowledge, to

communication if its initiative is 1. Denoting the initiative as
I, it is determined as fol ate, there is no unified cross-layer communication protocol $\leq \lambda$
1, if
λ_{relay}
Th_{relay}
for efficient and reliable event communication which considers
$\beta \leq \beta_{max}$
transport, routing, medium access functionalities with physical layer (wireless channel) effects for WSNs.

$\mathbf{I} =$

E_{rem}

min_{rem} $\geq E$
(1)

In this paper, a unified cross-layer module (XLM) is developed which achieves efficient and reliable event communication in WSNs with minimum energy expenditure. XLM melts common protocol layer functionalities into a cross-layer module for resource-constrained sensor nodes. The operation of the XLM is devised based on the new notion of *initiative determination*, which constitutes the core of the XLM and implicitly incorporates the intrinsic communication functionalities required for successful communication in WSN. Based on the initiative concept, XLM performs received based contention, local congestion control, and distributed duty cycle operation in order to realize efficient and reliable communication in WSN. Analytical performance evaluation and simulation experiment results show that XLM significantly outperforms the traditional layered protocol architectures in terms of both network performance and implementation complexity.

The remainder of the paper is organized as follows. Our cross-layer approach basics, overview, and protocol description are introduced in Section II. In Section III, we provide performance evaluations of the XLM solution and provide a comparative analysis with five layered suites. Finally, the paper is concluded in Section IV.

II. PROTOCOL DETAILS

Our cross-layer protocol replaces the entire traditional lay- ered protocol architecture that has been used so far in WSNs. The design principle is complete unified cross-layering such that both the information and the functionalities of traditional communication layers are melted in a single protocol. To this end, cross-layer protocol incorporates initiative determination, received based contention, local congestion control, and dis- tributed duty cycle operation as explained in the following sections in detail. Here, we first provide an overview of the cross-layer operation.

The basis of communication in XLM is built on *initiative* concept. This concept provides freedom for each node to decide on participating in communication. Consequently, a completely distributed and adaptive operation is deployed. The next-hop in each communication is not determined in advance. Instead, an *initiative determination* procedure is used for each node to decide on participating in communication. Initiative determination constitutes the core of the XLM and implic- itly incorporates the intrinsic communication functionalities required for successful communication in WSN.

A node initiates transmission by broadcasting an RTS packet to indicate its neighbors that it has a packet to send. Upon receiving an RTS packet, each neighbor of node i decides to participate in the communication or not. This decision is given through *initiative determination*. The initiative determination is a binary operation where a node decides to participate in 0, otherwise

The initiative is set to 1 if all four conditions in (1) are satisfied. Each condition in (1) constitute a certain communi- cation functionality. The first condition enures reliable links be constructed for communication. For this purpose, it requires that the received signal to noise ratio (SNR) of an RTS packet, ξ_{RTS}, is above some threshold ξ_{Th} for a node to participate in communication. The second and third conditions are used for local congestion control. As explained in Section II-D, the second condition in this component prevents congestion by limiting the traffic a node can relay. The third condition ensures that the node does not experience any buffer overflow and hence, also prevents congestion. The last condition ensures that the remaining energy of a node E_{rem} stays above a mini- mum value, E^{min}. This constraint guarantees even distribution of energy consumption.

The cross-layer functionalities of XLM lie in these con- straints that define the *initiative* of a node to participate in communication. Using the initiative concept, XLM performs local congestion control, hop-by-hop reliability, and distributed operation. The details of XLM operation are explained next.

A. *Basics and Definitions*

We assume the following network model for the operation. Each node performs distributed duty cycle operation. The value of the duty cycle is denoted by δ and defines the ratio of the time a node is active. Each node is implemented with a
sleep frame with length T_S sec. As a result, a node is active for $\delta \times T_S$ sec and sleeps for $(1 - \delta) \times T_S$ sec. Note that the start and end times of each node's sleep cycle are by no

means synchronized. As a result, a distributed duty cycle is employed. Moreover, we classify the sensor nodes in terms of two main duties. The *source duty* refers to the nodes with event information that need to transmit their packets to the sink. Hence, these nodes perform transmission rate selection based on the congestion in the network. Moreover, the *router duty* refers to the nodes that forward the packets received from other nodes to the next destination. These nodes indicate their initiative on accepting new flows through their path to the destination.

Based on these duties, each node determines its initiative to participate in the transmission of an event as explained above.

B. *Transmission Initiation*

When a node has a packet to transmit, it first listens to the channel for a specific period of time. Since a node may be spatially correlated with its neighbors, it also checks if its information is correlated with the transmitting source nodes, abandoning the transmission if a correlated node exists [10]. If

the channel is occupied, the node performs backoff based on its performs retransmission. The transmitter node times out if it

contention window size CW . When the channel is idle, the does not receive a

CTS packet after $\Sigma N pr$ CW , and performs RTS

node broadcasts an RTS packet, which contains the location of the sensor node *i* and the location of the sink. This packet retransmission.

$j=1$ j

serves as a link quality indicator and also helps the potential destinations to perform receiver-contention which is explained in Section II-C. When a node receives an RTS packet, it first checks the source and destination locations. It is clear that, in order to route a packet to the destination, the next hop should be closer to the sink than node *i*. We refer to this region where the neighbors of a node that are closer to the sink reside as *feasible region*. Similarly, the region where the neighbors of a node that are farther to the sink is referred to as the *infeasible region*. Hence, a node receiving a packet first checks if it is inside the feasible region of the transmitting node *i*. In order to save energy, the nodes inside the infeasible region of node *i* switch to sleep. The nodes inside the feasible region perform initiative determination according to (1). If a node decides to participate in communication, it performs receiver contention as explained in Section II-C.

C. *Receiver Contention*

The receiver contention operation of XLM is based on the receiver-based routing [6], [12]. After an RTS packet is received, if a node has initiative to participate in the communication, it performs receiver contention to forward the packet. The receiver contention is based on the routing level of each node which is determined by its location. The routing level of a node is decided based on the progress a packet would make if the node forwards the packet. The feasible region is divided into N_p priority regions corresponding to an increasing progress, i.e., A_i, $i = 1, ..., N_p$. The nodes with the longer progress have higher priority over other nodes. This prioritization is performed by the contention mechanism for medium access.

Each priority region, A_i, corresponds to a backoff window size, CW_i. Based on Σ the location, a node determines its region

D. *Local Cross-Layer Congestion Control*

Here, we consider two sources of traffic as an input to the buffer of each node:

- *Generated packets:* The sensing unit of a node senses the event and generates the data packets to be transmitted by the sensor node during its source duty as discussed in Section II-A. We refer to these packets as the *generated packets*. For a node i, the rate of the generated packets is denoted by λ_{ii}.
- *Relay packets:* As a part of its router duty, a node also receives packets from its neighbors to forward to the sink due to multi-hop nature of sensor networks. These packets are referred as the *relay packets*. The rate at which a node $\boldsymbol{i}$ receives relay packets from a node $\boldsymbol{j}$ is denoted as λ_{ji}.

The input rate to the buffer of node i is hence the combination of the input rates of these two types of packets.

Hence, based on the above definitions, the local cross- layer congestion control component of XLM has two main congestion control measures. The main idea of XLM cross- layer congestion control is to regulate the congestion:

- in router duty, by enabling the sensor nodes to decide whether or not to participate in the forwarding of the relay packets based on its current load due to its relaying functionality, and
- in source duty, by explicitly controlling the rate of the generated data packets.

We first analyze the upper bound for total relay packet rate a sensor node can accommodate to obtain a decision measure for congestion control in router duty. This is used in the XLM initiative determination as given in (1) in Section II.

To this end, we assume in our analysis that the generated packet rate, λ_{ii} of each node is fixed. Hence, the input packet

and backs off for

$i-1$ CW

$+ cw$, where cw

$\in [0, CW]$.

rate at the node i's buffer, λ_i, can be represented as

$=1$ j i i i

This backoff scheme helps differentiate nodes of different Σ

progress into different prioritization groups. Only nodes inside the same group contend with each other. The winner of the

$$\lambda_i = \lambda_{ii} + \lambda_{i,relay} = \lambda_{ii} +$$

$\in N_i^{in}$ j

λ_{ji} (2)

contention sends a CTS packet to node i indicating that it will forward the packet. On the other hand, if during backoff, a potential receiver node hears a CTS packet, it determines that a node with a longer progress has accepted to forward the packet and switches to sleep.

As node i receives a CTS packet from a potential receiver, it determines that receiver contention has ended and sends a DATA packet indicating the position of the winner node in the header. The CTS and DATA packets both indicate the other contending nodes the transmitter-receiver pair. Hence, other nodes stop contending and switch to sleep. Note that in the case of two nodes sending CTS packets without hearing

where N_i^{in} is the set of nodes which have node i as the next hop and λ_{ji} is the packet rate from node j to node i. Moreover,

the output rate of a node can be given by

$\mu_i = (1 + e_i)(\lambda_{ii} + \lambda_{i,relay})$ (3) where e_i is the packet error rate. A node is active on average

δ fraction of time. Hence, the average time a node spends in transmitting, receiving and listening during the active period can be given by

$$T_{rx} = \lambda_{i,relay} T_{PKT} ,$$
$$T_{tx} = (1 + e_i)(\lambda_{ii} + \lambda_{i,relay}) T_{PKT} ,$$

each other, the DATA packet sent by the node can resolve the contention. Since each time small number of nodes contend in

$$T_{listen} = \delta - (1 + e_i)\lambda_{ii} + (2 + e_i)\lambda_{i,relay} T_{PKT} ,$$

the priority regions the collision probability is small in XLM. However, in the case of CTS collisions, the transmitter node

respectively, where T_{PKT} is the average duration required to successfully transmit a packet to another node, λ_{ii} is the generated packet rate, and $\lambda_{i,relay}$ is the total input relay packet rate of node i.

In order for a node to prevent buffer overflow and maintain its duty cycle, $T_{listen} \geq 0$. Consequently, the input relay packet rate, $\lambda_{i,relay}$ is bounded by

$$\lambda_{i,relay} \leq \lambda_{i,relay}^{Th} , \quad (4)$$

where the relay rate threshold, $\lambda_{i,relay}^{Th}$, is given by

$$\lambda_{i,relay}^{Th} = \frac{\delta}{(2 + e_i)T_{PKT}} - \frac{1 + e_i}{2 + e_i}\lambda_{ii} . \quad (5)$$

TABLE I

SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Re-tx. Limit	7	P_t	5 dBm
β	2	PL(d_o)	55 dB
α	$\lambda_{ii_o}/10$	P_n	-105 dBm
Buffer Length	30	n	3
$l_{control}$	20 bytes	σ	3.8
l_{data}	100 bytes	$T_{coherence}$	16 ms
Frame Length	5s	E_{rx}	13.5 mW
Energy Threshold	100 μJ	E_{tx}	24.75 mW
ξ_{Th}	10 dB	E_{sleep}	15 μW

As a result, XLM incorporates a hop-by-hop congestion control which is devised based on this buffer occupancy analysis. Nodes participate in routing packets as long as (4) is satisfied. According to (5), the relay rate threshold is directly proportional to the duty cycle value, δ. This suggests that the capacity of the network will decrease as δ is reduced. However, since lower δ results in less energy consumption, this tradeoff needs to be analyzed carefully.

In addition to congestion control based on regulating the relaying functionality as discussed above, the XLM local congestion control component also takes an active control measure in case of network congestion, by directly regulating the amount of traffic generated and injected to the network.

During the receiver-contention mechanism described in Section II-C, node i may not receive any CTS packets. Considering wireless errors and the dynamic nature of the network due to duty cycle, δ, node i first performs retransmission in order to recover the loss and also probe the network condition. If no CTS packets are received, then node i decides that there is a congestion in the network. Then, it decreases its transmission rate by decreasing the amount of traffic generated by itself. In other words, since the traffic injected by any node due to its router duty is controlled based on (4), the active congestion control is performed by controlling the rate of generated packets λ_{ii} at the node i itself.

Therefore, in case of congestion, XLM node reduces the rate of generated packets λ_{ii} multiplicatively, i.e.,. $\lambda_{ii} = \lambda_{ii} \cdot 1/\beta$, where β is defined to be the transmission rate throttle factor. If there is no congestion detected, then the packet generation

rate can be increased conservatively in order not to lead to oscillation in the local traffic load. Therefore, XLM node increases its generated packet rate linearly for each ACK packet received, i.e., $\lambda_{ii} = \lambda_{ii} + \alpha$. Here, we select $\beta = 2$, i.e., the rate of generated packets is halved in case of congestion, and $\alpha = \lambda_{ii}0 /10$, where $\lambda_{ii}0$ is the initial value of the generated packet rate set by the sensing application. Here, note also that XLM adopts a rather conservative rate control approach. This is mainly because it has two functionalities to control the congestion for both source and router duties of a sensor node. As the node decides to take part in the forwarding based on its buffer occupancy level, it already performs congestion control as part of the XLM's forwarding mechanism. Hence, XLM node does not apply its active congestion control measures to the overall transmission rate. Instead, it only updates the generated packet rate, λ_{ii}.

III. PERFORMANCE EVALUATION

In order to gain more insight into the protocol operation, we present a comparative study between XLM and five different layered protocol suites consisting of state-of-the-art protocols. The existing sensor network simulation platforms are not suitable for cross-layer communication suite design due to their layered architecture. For this reason, we evaluate XLM and various layered protocol suites in cross-layer simulator (XLS) developed at our laboratory in C++. XLS consists of a realistic channel model and an event-driven simulation engine. We present simulation results for a sensor topology of 300 nodes randomly deployed in a 100x100m^2 sensor field. The sink is located at coordinates (80,80). The simulation parameters for both sensor nodes and the communication suites are given in Table I. In each simulation, an event occurs in an *event area* located at coordinates (20,20) with an event radius of 20m. Each source node reports its event information to the sink. To investigate the effect of duty cycle, each simulation is performed for duty cycle values of $\delta \in [0.1, 1]$. Each simulation lasts for 60s and the results are the average of five trials for each of five different random topologies.

We first identify the protocol configurations and then present the results of our comparative evaluation.

A. *Protocol Configurations*

The protocol configurations implemented for the compara- tive evaluation are shown in Table II. Note that the existing protocols that we have implemented in the layered protocol suites are usually proposed considering only their related layers with reasonable assumptions about the other layers. As an example, in the geographical routing protocols [5], each node is assumed to know the locations of their neighbors. However, actual implementation and operation of such an information exchange procedure is important especially when comparing these solutions to the proposed XLM solution. Since the receiver-based approach employed in the XLM does not require such an explicit information exchange, this constitutes a major overhead for the layered protocol suites using such an approach. Moreover, since duty cycle is deployed in our solution, each neighbor of a node may not always be active. Hence, for each protocol to work together in the protocol suites, we have made some implementation modifications.

Accordingly, in *GEO*, *PRR*, and *PRR-SMAC*, each node broadcasts a beacon to indicate its position and the remaining time to sleep. This beacon is sent at the beginning of each sleep frame when a node wakes up. Each neighbor that receives this beacon determines that the specific node will be active for the duration specified in the beacon. In the case of *PRR* and *PRR-SMAC*, this beacon also serves as a channel quality indicator. To optimize the network performance, in *GEO* and *PRR*, the beacons are piggybacked if there is a packet in the queue. In *PRR-SMAC*, a pairwise cross-layering is used and the routing beacons are sent with the SYNC packets. Similarly, SYNC packets are piggybacked if there is a packet in the queue.

TABLE II
PROTOCOL CONFIGURATIONS

Configuration	Transport Layer	Routing Layer	MAC Layer
Flooding	CBR	Flooding	CSMA w/o ARQ
[GEO]	ESRT [1]	Geographical Routing [5]	CC-MAC [10]
[PRR]	ESRT [1]	PRR-based G.R. [5]	CC-MAC [10]
[PRR-SMAC]	ESRT [1]	PRR-based G.R. [5]	SMAC [11]
[DD-RMST]	RMST [7]	Directed Diffusion [3]	CSMA w/o ARQ
XLM	XLM		

We have indicated that *DD-RMST* is used only for operation without duty cycle, i.e., $\delta = 1$. This decision is due to the fact that neither directed diffusion nor RMST consider duty cycle operation [3], [7]. Therefore, the *DD-RMST* protocol configuration is evaluated only for $\delta = 1$ for fairness and completeness of the evaluations.

We next present the results for operation with duty cycle, by changing the duty cycle δ from 0.1 to 1 in Section III-B. Since *DD-RMST* is only considered for operation without duty cycle, the performance metrics corresponding to this configuration are shown as a single point at $\delta = 1$ in the figures.

B. *Results*

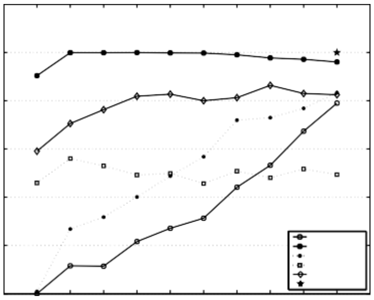
The goodput of the communication suites are shown in Fig. 1 (a). Irrespective of the duty cycle value, δ, XLM provides very high reliability. The cross-layer communication paradigm of the XLM that is adaptive to the network topology enables such high performance even when the network operates at low duty cycle. Moreover, *DD-RMST* provides 100% reliability while XLM results in a reliability of 96% for operation without duty cycle, i.e., $\delta = 1$. Note that RMST protocol uses hop-by-hop recovery with negative acknowledgments to request missing packets. On the other hand, XLM aims to first prevent link losses by constructing non-congested, high quality paths and then ensures high reliability by hop-by-hop ARQ technique. This approach results in reliability comparable to RMST at a significantly lower cost as we will discuss next.

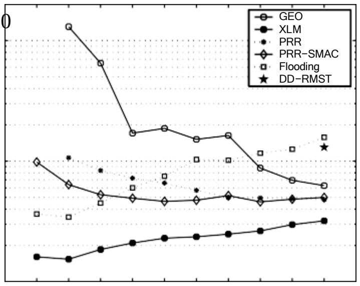
The simulation logs reveal that the decrease in reliability for the other layered protocol suites is mainly because of the significant number of packet drops due to retransmission timeouts. This suggests that nodes cannot find their intended next hops due to either low channel quality or because the nodes switch to sleep state before receiving any packets. This is exacerbated especially in the case of low duty cycle. As a result, the reliability of the network is hampered significantly. In Fig. 1 (b), the energy consumption per packet is shown. In Fig. 1 (b), the values for *GEO* and *PRR* at $\delta = 0.1$ are not shown since no packets are received by the sink. It can be seen that XLM consumes significantly less energy per packet and hence is highly energy efficient when compared to the other layered protocol suites. This difference is mainly because of the periodic broadcast of beacon packets in *GEO* and *PRR*, and SYNC packets in *PRR-SMAC*. Furthermore, the significant percentage of retransmission timeouts indicate significant energy wastage due to packets that cannot be transmitted to the sink. Since the network and MAC layers operate independently, the nodes chosen by the routing layer cannot be reached and significant energy consumption occurs. An interesting result is the significantly low energy effi- ciency of *DD-RMST*. Although this configuration provides 100% reliability as shown in Fig. 1 (a), the layered structure of the routing, transport and MAC functionalities results in a high penalty. As explained before, the routing layer, i.e., directed diffusion, incurs significant amount of overhead in order to maintain end-to-end paths between sources and the destination. On the contrary, XLM employs an adaptive routing technique that provides an energy efficient path in terms of both link quality and energy consumption distribution. Another impor- tant observation from Fig. 1 (b) is that the energy consumption per packet for XLM has a minimum at $\delta = 0.2$. Hence, we observe that the duty cycle value of $\delta = 0.2$ provides the most energy efficient performance for the operation of XLM.

The advantages of using a separate routing layer in the layered protocol suites can be seen from Fig. 1 (c), where the average hop count is shown. *GEO*, *PRR*, *PRR-SMAC*, and *DD-RMST* result in less number of hops for the packets that reach the sink than XLM. This is due to the fact that the routing algorithms in these layered protocol suites aim to find the smallest number of hops. While this result may be incorporated as a disadvantage when only routing layer is taken into account, the overall performance of XLM reveals that, routing layer performance alone does not provide efficient communication in WSNs, and other effects such as link quality, contention and congestion levels necessitate a cross-layer approach in route selection for overall network efficiency. Furthermore, as shown in Fig. 1 (d), XLM results in end- to-end latency comparable to *PRR*. *GEO* results in smaller end-to-end delay since the routing is performed based only on geographical location. On the other hand, *PRR-SMAC* results in higher end-to-end latency due to the clustered scheduling of nodes. Fig. 1 (d) also clearly shows the trade off of *DD-RMST* in achieving high reliability. This results in significantly high latency values when compared to the other configurations.

The end-to-end latency for *Flooding* is significantly higher for the limiting cases, i.e., $\delta = 1$ and $\delta \leq 0.2$. When all the nodes are active, flooding causes significant amount of contention and congestion leading to higher buffer occupancy time for each packet at each hop leading to higher latency. On the other hand, when the duty cycle is low, each time a node receives a packet, it has to go through one duty cycle

6





1

10^0

0.8

0.6

10^{-1}

0.4

0.2

0

8.5

8

7.5

7

6.5

Avg. Number of Hops

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8

Duty Cycle (□)

(a)

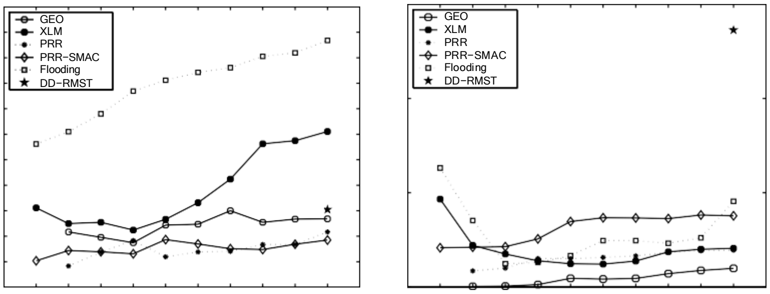
GEO XLM PRR
Flooding PRR−SMAC DD−RMST

0.9 1

10^{-2}

7

15



10

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Duty Cycle (□)

(b)

6

Latency (s)

5.5

5

5

4.5

4

3.5

3

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Duty Cycle (□)

(c)

0.1 0.2 0.3 0 0.4 0.5 0.6 0.7 0.8 0.9 1

Duty Cycle (□)

(d)

Fig. 1. (a) Average goodput, (b) average energy consumption per packet, (c) average hop count, and (d) average latency vs. duty cycle for layered protocol suites and XLM.

before it can re-broadcast the packet. This in turn increases the end-to-end latency. Similarly, the end-to-end latency of XLM increases for low δ. The reason for this increase is evident from Fig. 1 (c). Note that for $\delta = 0.1$, 14% of the transmitted packets are dropped due to retransmission timeout. This is due to the fact that, sender nodes cannot find any neighbors that satisfy the constraints in (1) discussed in Section II. As a result, the end-to-end latency increases due to retransmissions.

IV. CONCLUSION

7

XLM is a cross layer communication module for WSNs, which replaces the entire traditional layered protocol archi- tecture that has been used so far in WSNs. The design principle of XLM is complete unified cross-layering such that both the information and the functionalities of traditional communication layers are melted in a single module. The protocol operation of XLM is governed by the new concept of initiative determination. Based on this concept, XLM performs received based contention, local congestion control, and dis- tributed duty cycle operation in order to realize efficient and reliable communication in WSN. In a cross-layer simulation platform, the state-of-the-art layered protocol configurations have been implemented along with XLM to provide a complete evaluation. Analytical performance evaluation and simulation experiment results show that XLM significantly improves the communication performance and outperforms the traditional layered protocol architectures in terms of both network per- formance and implementation complexity.

REFERENCES

[1] Ö . B. Akan and I. F. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003-1016, October 2005.

[2] L. van Hoesel, T. Nieberg, J. Wu, and P. J. M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 78 - 86, Dec. 2004.

[3] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. on Networking*, vol. 11, no. 1, pp. 2 - 16, February 2003.

[4] T. Melodia, M. C. Vuran, D. Pompili, "The State of the Art in Cross- layer Design for Wireless Sensor Networks," *to appear in Springer Lecture Notes in Computer Science (LNCS)*, 2006.

[5] K. Seada, M. Zuniga, A. Helmy, B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks,", in *Proc. ACM Sensys '04*, November 2004.

[6] P. Skraba, H. Aghajan, A. Bahai, "Cross-layer optimization for high density sensor networks: Distributed passive routing Decisions," in *Proc. Ad-Hoc Now'04*, Vancouver, July 2004.

[7] F. Stann and J. Heidemann, "RMST: Reliable data transport in sensor networks," in *Proc. IEEE SNPA '03*, pp. 102-112, Anchorage, Alaska, USA, April, 2003.

[8] M. C. Vuran, Ö . B. Akan, , and I. F. Akyildiz, "Spatio-Temporal Correlation: Theory and Applications for Wireless Sensor Networks,"
Computer Networks Journal (Elsevier), Vol. 45, No. 3, pp. 245-261, June 2004.

[9] M. C. Vuran, V. B. Gungor, , and Ö . B. Akan, , "On the interdependency of congestion and contention in wireless sensor networks," in *Proc. SENMETRICS '05*, July 2005.

[10] M. C. Vuran, I. F. Akyildiz, "Spatial Correlation-based Collaborative Medium Access Control in Wireless Sensor Networks," *to appear in IEEE/ACM Transactions on Networking*, June 2006.

[11] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Co- ordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493-506, June 2004.

[12] M. Zorzi, R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Trans. Mobile Computing*, vol. 2, no. 4, pp. 337-348, Oct.-Dec. 2003.