Context Modeling in Collaborative Sensor Network Applications

Mohamed Hefeida[1], Turkmen Canli[2], Ajay kshemkalyani[3], and Ashfaq Khokhar[1]
[1]*Electrical and Computer Engineering Department, University of Illinois at Chicago*
{mhefei2,ashfaq}@uic.edu
[2] *Lemko Corporation*
turkmen@gmail.com
[3] *Computer Science Department, University of Illinois at Chicago*
ajay@uic.edu

ABSTRACT

The design of context aware protocols in Wireless Sensor Networks (WSNs) is an emerging challenge. The interpretation of the sensed information greatly depends on the context and for efficient processing and communication, context awareness can play a major role. In most of the existing WSN protocols, context awareness is exploited in a single dimension and is captured either at the application layer or at the routing layer using a single context parameter. In this paper, we develop a new WSN context model to efficiently capture multiple context parameters in multiple dimensions (i.e. context from/to different layers of the network stack) and adjust the network behavior accordingly while simultaneously balancing the network load. The new model not only considers context parameters reflecting run-time application demands from a node, but also takes into consideration the current state of the node as well as the state and demands of neighboring nodes (inter-nodal context sharing). The new model: (a) represents context demands from each layer; (b) reflects the current individual state of each layer; (c) communicates (a) and (b) to the neighboring nodes to impact the decision process; and (d) locally distributes available resources aiming to achieve an optimal load balance. We show the application of this model in realizing a cross-layer optimized protocol for routing multi-hop and multi-packet traffic in WSNs.

KEYWORDS: Wireless sensor networks, modeling, context awareness, collaboration, routing, medium access control.

1. NTRODUCTON

Wireless Sensor Networks (WSNs) are becoming part of our daily lives as a result of the continuous advances in VLSI technology which enables the development of low power, low cost and highly integrated sensing devices. Most of WSN applications, such as environmental monitoring, target tracking, medical systems and multimedia applications, are subsets of a larger network employing collaboration between heterogeneous sensor nodes. To achieve efficient collaboration in a heterogeneous WSN environment, a node needs to be completely aware of its own resources, state, and application demands, as well as those of neighboring nodes.

Context awareness in WSNs has been explored at different levels. It has been studied at the application level [1], routing level [2, 3] and MAC level [4], independently. Some cross-layer interaction has also been considered such as in [5, 6]. RMAC [5] improves the end-to-end latency problem of duty-cycle based MAC protocols in multi-hop transmissions by setting multi-hop flows using routing layer information. PRMAC [6] extends RMAC to allow transmission of multiple packets in a single flow.

In [7], the context information gathered is utilized by the routing layer after being refined by the application; this can be viewed as an application-routing cross layer optimization. In [8, 9], the context awareness problem is reduced and abstracted into a new middleware aiming to achieve adaptability of the network away from the nodes. In [3], battery life has been considered as a context parameter at the routing layer in lieu of the traditional shortest path. Application specific context aware routing has also been explored, such as in [10] which uses a combination of environmental conditions and other context criteria for routing. Canli et al. [11] proposed a cross-layer MAC protocol (BulkMAC) that cleverly

978-1-61284-639-2/11/$26.00 ©2011 IEEE

utilizes routing information and adapts its performance accordingly. Although BulkMAC can be seen as a subset of this work, like RMAC [5] and PRMAC [6], it doesn't directly incorporate the application in the protocol decision.

Although context awareness has been explored at a high level in all of the above works, full awareness and consideration of the context(s) of interest along with a nodes' available resources at all levels of the network paradigm is yet to be investigated. The impact of the application on the functionality of the lower networking layers (particularly routing and MAC), if present, is very limited. While abstraction facilitates design, compatibility and scalability, it also hides information that can be useful to other layers in the hierarchy. This can be beneficial in homogenous environments, where limited or no flexibility is required. However, in heterogeneous collaborative environments, sharing some of the hidden information between layers of the same node and between different nodes shows promising gains [5, 6, 11]. For example, if the MAC layer is aware of the context(s) of interest the routing layer is responding to, it can help the routing layer achieve the application demand by adjusting its own behavior. We refer to this type of context sharing as inter-layer context sharing.

In some cross-layer work [5, 6], limited collaboration between two layers was considered and showed significant gains when compared to regular non-collaborative work [4]. This inspired us to further investigate information (context) sharing between layers and nodes which we present in this article by formally modeling such phenomena in order to explore possible gains and tradeoffs.

This paper presents a new context aware model to exploit WSNs' full potential by:

(a) Getting the running application(s) involved in lower layer decisions and giving them the ability to control/tailor the network behavior
(b) Integrating main context parameters of interest in WSNs (e.g. battery life, delay, mobility) into one framework allowing nodes to make more informed decisions
(c) Communicating context parameters between all layers (inter-layer context sharing)
(d) Communicating context parameters and node state between nodes (inter-nodal context sharing)
(e) Distributing the load over the entire network to achieve load balancing and prolong network life time

All the above is projected to lead nodes to make more informed decisions that consider node/network status as well as application preferences and demands. This takes context awareness, particularly in collaborative WSNs, to a new level of integration, awareness and adaptability. Figure 1 shows a high level of the model functionality where context is communicated and considered within a node and between different nodes.

The rest of the paper is organized as follows: Section II describes the proposed model. Section III illustrates the functionality of the proposed model via a detailed example. Section IV evaluates the new model followed by the conclusion and future work in Section V.

2. PROPOSED MODEL

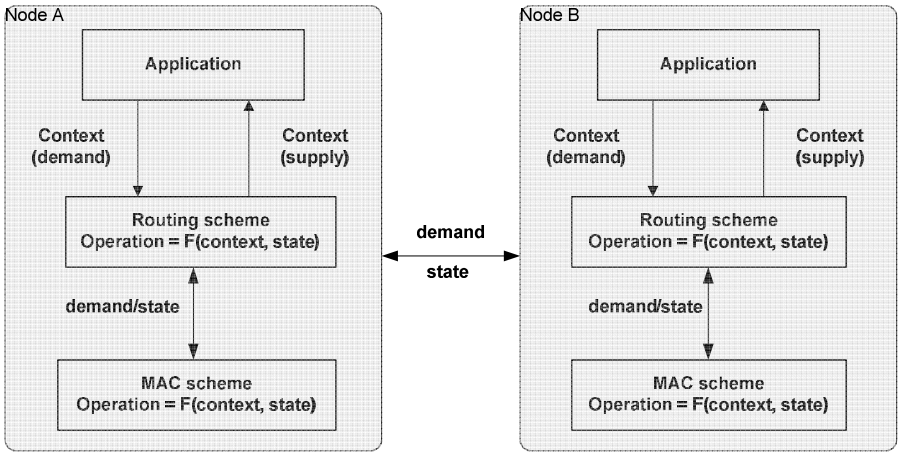
Every context parameter is represented by a coefficient which reflects its relevance to the running application. The current state of the node in regard to every context parameter is represented by a variable. The relationship between the variables and their coefficients in our model is similar to a supply and demand model. Nodes attempt to meet application demands while considering the current node/network state and balancing the load across the network as illustrated in Figure 1.

We consider only a few major context parameters in our model to illustrate it's functionality, however, the model can be expanded to include other parameters. We consider available battery power (B), delay tolerance (D), and node mobility (M). Each of these parameters reflects the node's state with respect to the associated context and is assigned a corresponding coefficient. The coefficients b_i, d_i and m_i reflect the applications' need for B_i, D_i and M_i respectively (higher values of these coefficients reflect higher needs). The coefficients assigned by the application, along with the context parameters' state on a node, generate a cost function $E_i(B, D, M)$. This function reflects the effort required by node i and is given by:

$$E_i(B, D, M) = \frac{b_i}{B_i} + \frac{d_i}{D_i} + \frac{m_i}{M_i} \qquad (1)$$

We define B as the percentage of battery remaining, D is the average delay tolerance of all packets in a node's buffer, and M is the nodes mobility (can be computed based on the rate of change of neighbors). Variables B, D and M can have a minimum value of 0.1 (to avoid undefined values for the cost function) and a maximum value of 1.

For example, consider a node with 30% of its battery remaining, having 2 delay sensitive and 1 delay irrelevant packets in its routing buffer, having one of its five



Fgure 1. Architecture of Proposed Model

neighbors change in the past 5 cycles. Such a node will have the values of 0.3, 0.33 and 0.2 for the context parameters B, D and M, respectively. This is independent of the application needs which are represented by the coefficients.

The coefficients b, d and m can have values between 0 and 1, with 0 and 1 corresponding to minimum and maximum importance/need, respectively. For example, a real time video streaming application requires very tight delay constraints (i.e. d =1) and will consume a great amount of battery power (i.e. b = 1), however mobility can be irrelevant (i.e. m = 0). A real time audio stream will indeed require very tight delay constraints (i.e. d =1) but will consume much less battery power (i.e. b = 0.5). This allows the application to directly affect how the underlying node behaves.

Load balancing is achieved by communicating the cost function E between neighbor nodes and recalculating it using:

$$E_i^{'} = f\ (E_1, E_2 \ldots, E_N) \qquad (2)$$

Function f can be application dependent reflecting context interaction between node i and its N neighbors. In its simplest form this function could be just an average. A node achieves $E_i^{'}$ by adjusting the coefficient values of E_i without compromising application needs. If not possible, a node will try to get as close as possible to the value of $E_i^{'}$. Nodes with $E_i > E_i^{'}$ will try to give some of their tasks to one of the neighbor nodes with $E_i < E_i^{'}$, hence achieving a balanced load.

Communicating essential model information, such as E_i, can greatly vary according to the level of detail required. It can vary from simply communicating one value (E_i) to communicating the entire $2C$ values constructing E_i, where C is the number of context parameters of interest (C=3 in the example above). Any combination of the above two extremes is also possible, however, determining the optimum amount of information to be communicated is out of the scope of this paper.

Based on the proposed context aware model, we develop a new collaborative routing/MAC scheme which manages available resources efficiently and optimizes its behavior to the current context demands and node states. The new scheme efficiently handles a very wide range of traffic patterns and loads. This is achieved by shifting resources when needed, such as allowing the transmission of multiple packets over multiple hops in a single duty cycle. The proposed collaborative scheme captures multiple context parameters at the application, routing and MAC layers in a single node as well as the context information coming from neighboring nodes. This multi-layer context awareness within a node along with the inter-nodal context information communicated between nodes explains its superior performance and flexibility compared to other protocols in the literature.

3. A CASE STUDY INCORPORATNG THE PROPOSED MODEL

In this section we present a Context-Aware Routing-MAC Scheme (CARMS) capable of realizing multi-hop, multi-flow, and multi-packet transmissions in a single duty cycle. CARMS considers two context parameters, traffic load and routing information. It utilizes the routing layer context of similar packets to be routed (S) to setup multi-hop flows when needed. This makes it comparable to other cross layer schemes like RMAC [5], DW-MAC [12], and PRMAC [6]. It also relies on a duty cycle based

Table 1. Notations Used

Symbol	Description
ξ_i	Number of data packets the i^{th} node wants to send to the $(i+1)^{th}$ node
p_i	Final number of data packets the i^{th} node is going to receive from the $(i-1)^{th}$ node
λ_i	Number of data packets the i^{th} node has for the final destination of the flow
μ_i	Number of packets the i^{th} node cannot receive from the $(i-1)^{th}$ node
u	Duration of time required to complete single data packet transmission from one node to another.
s_i	Send start index of the i^{th} node of the flow
e_i	Send end index of the i^{th} node of the flow
T_{SLEEP}	Start time of the sleep period
φ	Carrier sense range
r	Data communication range
T_P	Packet transmission period

MAC protocol similar to S-MAC [4] and T-MAC [13], where nodes go to sleep periodically, and coordinate their wake up times.

CARMS is more responsive to traffic load changes than both PRMAC and RMAC. This is due to its' context awareness aspect. When the traffic load is low, it functions similar to RMAC; however, as the traffic load increases, it allows transmission of multiple packets in a single cycle.

Table 1 lists explanations of the notations used in this example. Note that the communication latency between any two nodes, u, is computed as follows:

$$u = \text{durDATA} + \text{SIFS} + \text{durACK} + \text{SIFS} \qquad (3)$$

where $durDATA$ and $durACK$ are the durations needed to transmit a single data and acknowledgement packets, respectively.

In a multi-hop flow, a *multi hop frame* (MHF) is used, it includes addresses of the sender, next hop, previous hop, final destination, send start and send end indexes. Figure 2 illustrates a multi-hop flow setup and data transmission schedule. Node S has two data packets and node A has one data packet to send to node C. Furthermore, node S has to send its packets through nodes A and B, and node A has to send its packets through node B.

To setup a multi-hop flow, first node S sends an MHF, it sets the final destination field to node C, next hop field to node A, previous hop field to null, source field to its

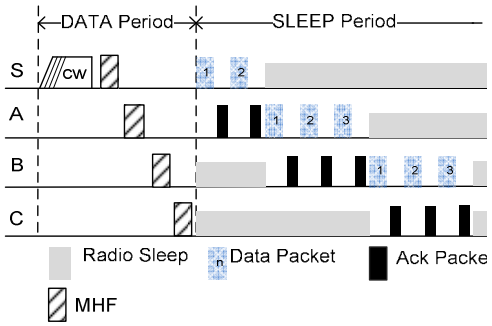


Figure 2. CARMS Multi-hop Flow

address, and send start and send end indices field to one and two, respectively. Second, node A, if capable of joining the multi-hop flow, sends an MHF. Its MHF is a confirmation for node S, and a flow setup request for node B. From the send start and send end indices field, node A understands that node S wants to send it two packets. In node A's MHF, previous hop is node S, next hop is node B, final destination is node C, and start and end indices are three and five. Node B sends an MHF in a similar way, and sets its send start end indices to 6 and 8 respectively. When node C replies back to node B, flow set up is concluded.

Node A calculates when it is going to receive data packets from node S, using node S's send start and end indices, and packet sending times from its own send start and end indices. Nodes B and C compute their packet reception and sending times similarly.

The send start and send end indices in an MHF are used to determine the number of packets each node will receive and send, as well as when it is going to receive and send. For instance using the s_i and e_i fields of the i^{th} node's MHF, the $(i+1)^{th}$ node can find the following information:

- $T_{SLEEP} + (s_i - 1)u$, the beginning of the first data packet transmission from the i^{th} node
- $T_{SLEEP} + (e_i - 1)u$, the beginning of the last data packet transmission from the i^{th} node
- $e_i - s_i + 1$, the number of packets node i wants to send

Due to limitations on packet buffer size or lack of remaining time to receive packets in the sleep period, a node may not be able to receive every packet its previous hop wants to send. Therefore, adjacent nodes in a flow negotiate the number of packets they are going to receive from or send to each other.

Let ξ_i given in equation (4) denote the number of packets the i^{th} node wants to send to the $(i+1)^{th}$ node, μ_i be the number of packets the i^{th} node cannot receive from the $(i-1)^{th}$ node out of r_{i-1} packets, λ_i be the number of packets in the i^{th} nodes buffer towards the final destination of the flow, and p_i be the number of packets node i is going to receive from the$(i-1)^{th}$ node when the flow setup ends.

$$\xi_i = e_i - s_i + 1 \qquad (4)$$
$$p_i = r_{i-1} - \mu_i \qquad (5)$$

Based on the $(i-1)^{th}$ node's MHF, node i computes the s_i and e_i fields in the MHF sent to the $(i+1)^{th}$ node as follows:

$$s_i = s_{i-1} + p_i \qquad (6)$$
$$e_i = s_i + p_i + \lambda_i - 1 \qquad (7)$$

When the $(i-1)^{th}$ node receives the i^{th} node's MHF to the $(i+1)^{th}$ node, it can compute the number of packets it can send to the i^{th} node by looking at the s_i field in the MHF, and thus finalizes the number of packets it will send and when it will send as follows:

- The first data packet transmission from the $(i-1)^{th}$ node will begin at $T_{SLEEP} + (s_{i-1} - 1)u$.
- The last data packet transmission from the $(i-1)^{th}$ node will begin at $T_{SLEEP} + (s_i - 2)u$.
- The $(i-1)^{th}$ node will send $s_i - s_{i-1}$ packets

The i^{th} node will not respond to the multihop flow set up request of the $(i-1)^{th}$ node if:

- It has overheard any flow setup packets
- It has already joined a flow
- $e_{i-1} \geq \lfloor \frac{duration\ of\ sleep\ period}{u} \rfloor$
- There is no time left in the data period to answer the $(i-1)^{th}$ node's request

If the i^{th} node is either the final destination, or its MHF sending is at the end of the data period, it sets the next hop address as null, as a result, its MHF serves as a confirmation to the $(i-1)^{th}$ node's request.

4. EVALUATION

CARMS is implemented in ns-2.29 [14], and is compared against RMAC [5] and PRMAC [6]. We assume that single schedule information is available to nodes like [5, 6]. Important simulation parameters are listed in Table 2.

Table 2. Simulation Parameters

Bandwidth	20 kbps	Communication Range	250m
Rx Power	0.5W	Interference Range	550m
Tx Power	0.5W	DIFS	10ms
Idle Power	0.45 W	SIFS	5ms
Sleep Power	0.05W	Contention window	64ms
PION(RMAC)	14B	ACK	10B
PION(PRMAC)	16 B	DATA	50B
SYNC	55.2 ms	CYCLE(PRMAC)	4.724s
DATA(PRMAC)	181 ms	SLEEP(PRMAC)	4.487s
CYCLE (RMAC)	2233ms	DATA(RMAC)	168ms

As in [6], the main performance metrics we are interested in are delivery ratio, average delay and energy consumption. *Packet delivery ratio* is the ratio of number of packets delivered to their destination to the total number of packets generated. *Average delay* is the average time for a packet to travel from its source to its final destination. Finally, *energy consumption* is the average energy spent by each node during the simulations. We divide energy consumption by packet deliver ratio to get the normalized average power consumed per packet. We compare CARMS to PRMAC and RMAC on a 12-hop chain network.

Figure 3 shows the average energy consumed to deliver one packet. CARMS shows great adaptability to traffic load variations compared to PRMAC and RMAC. It originally performs similar to PRMAC; however, as the packet rate increases it suffers fewer collisions and hence saves more energy. On the average, CARMS uses 40% less energy than PRMAC.

Figure 4 shows the average packet delay for the 12 hop chain network. RMAC outperforms PRMAC for packet

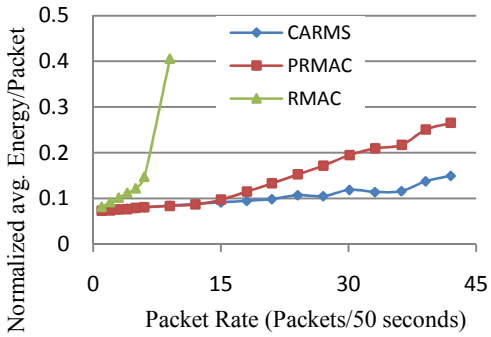


Figure 3. Avg. Power Consumed per Packet on the Chain Network

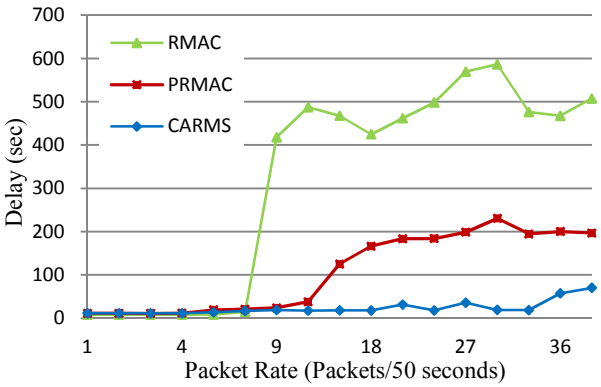


Figure 4. Avg. Delay on the Chain Network

rates lower than 9 packets/ 50 sec, due to its smaller cycle time, however, for higher traffic rates, average delay increases very rapidly. PRMAC's average delay jumps when the packet rate exceeds 15 packets/ 50 sec, as it can no longer handle the traffic load. CARMS can handle varying traffic loads better than PRMAC and RMAC due to its context awareness. CARMS takes into consideration both traffic load and routing layer information (context parameters of interest) in its decision. This allows better adaptation to network conditions, and hence reduces congestion (fewer packets in intermediate node queues).

5. CONCLUSON AND FUTURE WORK

We have presented a new WSN context model that efficiently captures application context demands, node status and communicates it across the network. This allows better resource allocation according to the current demand and network state. The new model takes context awareness to a new dimension by sharing context information between layers and hence resulting in a more informed decision. The new model also communicates context information between nodes in order to distribute the load over the network. We have developed CARMS, a new context aware routing mac scheme based on the proposed model. CARMS realizes context interaction between application, routing and MAC layers. Traffic load and routing information are used to tune the MAC scheme behavior accordingly.

Despite all potentials CARMS has shown, there are still many issues left open for future improvement. For example, the optimal granularity of the context information communicated between nodes. We are currently working on the theoretical analysis of our model to exploit its full potential and guide our investigations.

REFERENCES

[1] V.Q. Son, B.L. Wenning, A. Timm-Giel, C. Görg, "A model of Wireless Sensor Networks using Context-awareness in Logistics Applications," In proc. of ITST2009, Lille, France, pp. 2-7, Oct. 2009.

[2] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," In proc. of ACM SIGCOMM, Philadelphia, PA, pp. 133–14, Aug. 2005.

[3] S. Singh, M. Woo, C.S. Raghavendra, "Power-aware routing in mobile ad hoc networks," In proc. of ACM MobiCom, Dallas, Texas, pp. 181–190, 1998.

[4] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," In INFOCOM 2002. 21st IEEE International Conference on Computer Communications, volume 3, pp. 1567–1576, 2002.

[5] Shu Du, A.K. Saha, and D.B. Johnson, "Rmac: A routing enhanced dutycycle mac protocol for wireless sensor networks," In INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, pp. 1478–1486, May 2007.

[6] T. Canli and A. Khokhar, "Prmac: Pipelined routing enhanced mac protocol for wireless sensor networks," In the IEEE International Conference on Communications, ICC '09., pp. 1–5, June 2009.

[7] L. Shu, Y. Zhang, Z. Yu, L. T. Yang, M. Hauswirth, and N. Xiong, "*Context-aware cross-layer optimized video streaming in wireless multimedia sensor networks,*" The Journal of Supercomputing, Springer, 2009.

[8] A.Taherkordi, R. Rouvoy, Quan Le-Trung, and F. Eliassen, "Supporting Lightweight Adaptations in Context-aware Wireless Sensor Networks," 1st International COMSWARE Workshop on Context-Aware Middleware and Services. Dublin, Ireland, pp. *43–48,* June 2009

[9] L. Zhou, Naixue Xiong, Lei Shu, Athanasios V. Vasilakos, Sang-Soo Yeo, "Context-Aware Middleware for Multimedia Services in Heterogeneous Networks," *IEEE Intelligent Systems 25*(2): 40-47 (2010).

[10] B.L. Wenning, D. Pesch, A. Timm-Giel, C. Görg, "Environmental Monitoring Aware Routing in Wireless Sensor Networks," IFIP International Federation for Information Processing, Toulouse, France, Volume 284, pp. 5-16, September 2008.

[11] T. Canli, M. Hefeida, and A. Khokhar, " BulkMAC: a cross-layer based MAC protocol for wireless sensor networks," IWCMC 2010, pp. 442-446, June 2010.

[12] Y. Sun, S. Du, O. Gurewitz, and D.B. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," In Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, pp. 53–62, 2008.

[13] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," In Proceedings of SenSys'03. ACM, 2003.

[14] S. McCanne and S. Floyd. ns Network simulator. http://www.isi.edu/nsnam/ns