This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Power Efficient Adaptive Network Coding in Wireless Sensor Networks

Hnin Yu Shwe and Fumiyuki Adachi
Department of Electrical and Communication Engineering
Graduate School of Engineering, Tohoku University
Sendai, Japan

hninyu@mobile.ecei.tohoku.ac.jp

***Abstract*—Wireless sensor networks have been attracting a great attention due to their wide range of potential applications. However, due to various limitations arising from their inexpensive nature, limited size, weight and ad hoc method of deployment, the power consumption is one of the major constraints in sensor networks. Moreover, it is well known that packet communication dominates the power consumption in wireless sensor network. Therefore, it is very desirable to reduce the amount of packet transmission as much as possible. Network coding can achieve this by improving network throughput. Thus, recently researchers have been emphasizing on how wireless sensor networks can get benefits from network coding. AdapCode is one of the available works which apply the practical network coding to the wireless sensor networks. However, AdapCode has some limitations. It cannot find out all actual neighbour nodes which is critical in determining the new coding scheme. In this paper, thus, we enhance the AdapCode by deploying the power efficient neighbour discovery protocol to find out all the neighbours. Our objective is to develop a data dissemination protocol that promises the full advantage of network coding by deploying the power efficient neighbour discovery protocol.**

***Keywords*— wireless sensor network; power saving; network coding.**

I. Introduction

Wireless sensor networks (WSNs) are distributed, self-organizing solution to provide sensing and computing in various environments where conventional networks are impractical [1] [2] [3]. Such distributed sensor networks have been widely used for applications ranging from environment monitoring to military surveillance and disaster rescue. WSN is the most potential technology for very low power ubiquitous networks [4]. Recently, WSNs are widely used in monitoring of remote or hostile geographical regions, tracking of animals and objects, and monitoring in smart building and factories. Since the performance of the sensor network highly depends on the lifetime of the network, and hence to prolong the lifetime of sensor network, low power consumption is a critical design requirement for most sensor network applications. In addition, it is clear that the most power consumption action for WSN is the packet communication. Thus, in order to make optimal use of power, packet communication should be minimized as much as possible.

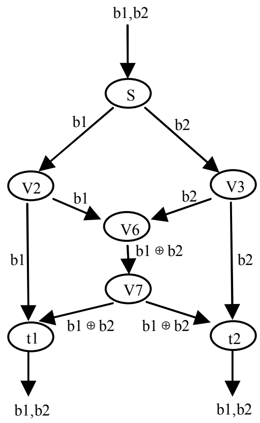

Figure 1. Network coding for throughput improvement.

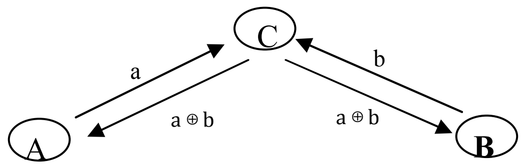

Figure 2. Network coding for power consumption reduction.

b1,b2

Network coding has been recently introduced to reduce traffic in general networks [1] [2]. Network coding is used to improve throughput or save bandwidth. The core idea of network coding is to allow the mixing of data (e.g., by an XOR operation or a linear combination) at intermediate network nodes. Encoding packets at intermediate nodes and then sending only coded packet instead of individual packets reduces the traffic without increasing delay. The network coding theory started with Ahlswede et al's seminal paper [6], where they show that the multicast capacity, which is defined as the maximum rate that a sender can communicate common information to a set of receivers, can be achieved via network coding, but cannot be achieved by routing in general. A simple example of how network coding reduces traffic in such a broadcast scenario can be seen in Figure 1 and Figure 2.

Figure 1 shows that the source, S, wants to deliver two data packets b1 and b2 to two sinks, t1 and t2. Each edge has the capacity of 1. We can see with network coding in node v6, b1 and b2 can be received by t1 and t2 simultaneously. Obviously, this cannot be achieved without coding in v6. In this case, network coding improves network throughput. Let's see another simple example in Figure 2 where sensor A and sensor B want to exchange packet information through sensor C. Suppose that information one transmission consumes one unit energy and exchange of information packets a and b needs 4 unit energy in common. However, with network coding, intermediate sensor, C, combines packets a and b into a single encoded packet and then broadcasts it. Hence, the task can be accomplished by just 3 unit energy and in this case, network coding improves power consumption.

The fundamental idea of coding in intermediate nodes in network has been shown to have advantages in other scenarios such as minimizing network resources, network diagnosis and packet communication in wireless network. Network coding technique only needs a few linear operations and several bytes storage to reduce much power consumption, increasing the lifetime of sensor networks. In addition, the broadcast nature of WSNs increases the benefits of network coding.

AdapCode is by now the most promising network coding architecture for power saving in wireless sensor network. However, the coding process of AdapCode works based on only partial neighbours of a node which may waste some potential coding opportunities. In order to address it, we enhanced AdapCode by deploying power efficient neighbour discovery protocol. The main idea of our scheme is to use the same network coding algorithm for receiving and broadcasting the packets used in the AdapCode. However, to make sure that all the neighbour nodes are detected, we introduce the beacons in neighbour discovery protocol. Moreover, received signal strength used in our scheme also help in achieving better reliability.

The rest of the paper is organized as follows. We first briefly describe some related works in Section 2. Section 3 describes the key idea of our proposed method to discover the actual neighbour nodes in order to determine the proper coding choice. Simulation results are presented in Section 4. Finally, we conclude the paper in Section 5.

II. Related Works

Code dissemination is important in WSNs since program codes running on each sensor node are needed to be upgraded frequently according to application requirements or dynamic changes of network. In this paper, we focus on code dissemination where only one source node distributes the updated code packets which are needed to be known by all nodes in the network. Power is one of the most critical resources in WSNs. Since packet communication is a very power consuming action for sensor nodes, a lot of work has focused on reducing packet communication. One of the most widely used approaches is to do data aggregation [7]. However, this approach cannot be used when all the original packets are needed at the received nodes.

Network coding has been recently introduced to reduce traffic in general networks [6]. Network coding can save much power consumption by reducing the amount of packet transmission. When using network coding, data segments can be first grouped and encoded, then transmitted through multiple paths. As long as the number of packets that sink receives is equal to or larger than a lowest bound, the original messages can be reconstructed at the sink in a desired probability.

There are two design parameters to be considered in network coding [8]. The first is the optimal number N of packets which are to form a group. If N is too small, we are unable to take advantage of the potential capability of network coding. On the other hand, if N is too large, a node might not receive the required number of packets to decode the original packet. The second design parameter is the number of packets which are needed to deliver for obtaining the desired reliability. Different applications may have different optimal value.

TABLE I.
THE CHOICE OF N ACCORDING TO AVGNEIGHBOUR

avgNeighbour	05	5~8	8~11	11~
N	1	2	4	8

The coding scheme called AdapCode [9] depends on the node density (i.e. the number of neighbours a node has). Table I shows the choice of N, optimal number of packets to combine into one packet, in AdapCode. Throughout the code dissemination process, each intermediate node dynamically decides how many packets to be encoded together, N, based on its average number of neighbours. Then, it randomly generates N coefficients and computes the linear combination of N packets. The receiver node can recover the original packets by Gaussian elimination after receiving N coded packets successfully. Thus, for each M received packet, after determining the coding scheme, N, each node combines N packets into one packet and sends out total M/N number of packets and thus it can save bandwidth up to N-1/N.

The core idea of AdapCode is to adaptively change the coding scheme according to the node density. Thus, the performance of AdapCode highly depends on the number of neighbour nodes. Generally, if nodes have more neighbours, they can encode more packets together without losing reliability since they can get enough combinations from their neighbours to decode. However, AdapCode could not find out some neighbour nodes in some situations when determining the node density. AdapCode considers just full-active nodes as neighbours since each node regards the other node as a neighbour only after receiving some messages from that node. However, there may be some neighbours who have not sent any message to that node yet and we will regard those nodes as semi-active nodes. In addition, it is very clear that if we can find out all those neighbours (both full-active and semi-active), we can get the more coding chances. Based upon these facts, we will enhance the AdapCode. The main motivation is to reduce the power consumed for the entire network and to prolong the lifetime of the network by reducing the packet communications throughout the code dissemination process.

III. System Model

A protocol running in sensor networks must be simple and easily implemented. Moreover, the dynamic environment of WSNs should be considered; nodes can temporarily disconnect or fail and the node density between nodes can vary over time. A good algorithm should be adaptive to reflect this dynamic nature.

The idea of our scheme is to use the same network coding algorithm for receiving and broadcasting the packets used in AdapCode. However, in order to make sure that all the neighbour (both semi-active and full-active neighbour) nodes are detected, we introduce power efficient neighbour discovery phase in the beginning of code dissemination process. Our coding methodology is shown as pseudo code below.

Initially nodes do not have any knowledge of the location of their neighbours, and we model the distribution of nodes as a two-dimensional Poisson process with known intensity. We assume that sensor nodes are randomly distributed. The sensors are not previously configured with the knowledge of their locations and must transmit wireless queries to discover surrounding nodes and establish a communication network. Neighbour discovery is a very first application that runs on each sensor node to form a sensor network and is also essential to find the updated network information.

Algorithm 1. networkCoding()

```
coefMatrix ← M x M matrix
invMatrix ← M x M matrix
while packet transmission is going on do
  if a packet is received then
     sender ← the sender of the packet
     pageNumber ← the page number of the packet
     construct coefMatrix using coefficients from
      packets in page pageNumber
     rank ← Gaussian (coefMatrix, invMatrix)
     if rank=M then
       solve all messages in the page by invMatix
         numNeighbour ← neighbourDiscovery()
         determine N using numNeighbour
         broadcast M/N packets
         pageNumber ←pageNumber +1
     Endif
  Endif
Endwhile
```

In our approach, we introduce the network beacons to efficiently discover neighbour. The utilization of beacons for distributing neighbour information has several benefits. First, beacon exchanges are synchronized by a MAC protocol. This provides very low idle listening and power overhead due to additional control signal exchanges. Second, beacons are quite short and therefore it allows power-efficient implementation. In addition, there is no need for a new frame type. This makes this beacon very important for wireless networks especially for self-organizing networks.

In the initial deployment of sensor network, each sensor node broadcasts a specific number of beacon messages. During this period, each of its neighbouring sensors will constantly sample received signal strength (*rss*). And at the end of this period, all neighbouring sensors will calculate the mean of the measured signal strength (q) and store it for later use. The use of this value, q, prevents packet loss and thus improves the reliability of the network. Before the code dissemination, each network node sends such a beacon message to advertise its presence in order to find out the number of neighbour nodes. Due to the broadcast nature of radio communications, each sufficiently close node receives this message and may infer that it is a neighbour of the sender node.

All nodes maintain a neighbourhood table, and any localized protocol may make decisions based on this table. During the process of neighbour discovery, the identity of a discovered node is added to the local neighbourhood tables of the nodes which have received the beacon message. The operation of the neighbour discovery algorithm is presented as a pseudo code below. The algorithm requires data memory for storing next hop neighbourhood information. As the number of neighbours is at most k, the total number of neighbour entries is upper limited to k. Thus, the required program memory space for the algorithm is typically less than 1*kB* and hence it is suitable for the memory of sensor nodes.

Algorithm 2. neighbourDiscovery()

```
N ← 0
for all i є network do
  receive a beacon from i
  if reception was successful then
    update rss (i) based on the beacon reception
    if rss (i) ≥ q then
      N ← N+1
    Endif
  Endif
Endfor
```

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed scheme. Our goal is to get a better understanding on how well our scheme works when using neighbour discovery protocol compared with AdapCode.

We performed computer simulation using NS-2. We implemented a simple WSN where a single source node distributes the source codes in order the update the network. The default parameter setting used in our simulation is shown in Table II.

TABLE II.
SIMULATION PARAMETERS

Number of nodes	100
Area (m^2)	100 x 100
Mobility model	Random waypoint
Pause time(s)	30
Transmission range	25m
Channel Capacity	1 Mb/s
Data Rate	4 packets/ Unit Time
Radio Propagation Model	TwoRayGround
Routing Protocol	AODV

To guarantee 100% reliability which means all nodes receive all the packets, we also consider for negative acknowledgement. Observe that without negative acknowledgement, a node receiving *N*-1 packets and a node receiving no packets at all are equally problematic because neither of them can decode any of the original *N* messages. Nodes receiving less than *N* packets can send out negative acknowledgement to retrieve missing data they need that have already been decoded elsewhere. Thus, here, the number of expected packets to be transmitted is defined as the total number of packets to be transmitted plus twice the number of missing packets. The intuition behind the definition is that once a node misses a packet, it must send out one negative acknowledge message to its neighbours and one of its neighbours will reply the needed packet.

The simulations were done on a uniform topology consisting of 100 nodes spread in 100 m^2 area for different packet size 128- and 1024-bytes packets. For each simulation scenario, ten runs with different random seeds were executed and the results were averaged. The simulation results are plotted in Figure 3 and Figure 4. It is shown that our protocol uses fewer packets than AdapCode does and hence can save node power consumption up to 10% than AdapCode. The number of neighbours is used to indicate the density of the network. As we expected, the lifetime extension increases when the density of the network increases.

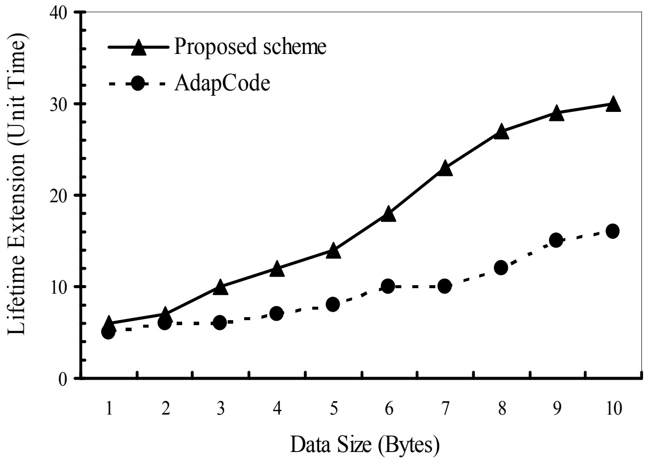

Figure 3. Lifetime extension with respect to the data size.

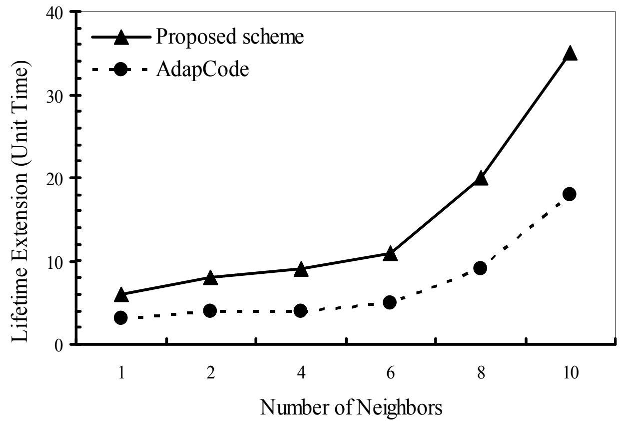

Figure 4. Lifetime extension with respect to the number of neighbours.

Finally, we will discuss the additional power consumption that neighbour discovery brings. As we can see in the simulation result shown in Figure 5, the beacon transmission overhead incurred by the neighbour discovery is very little compared with the power saving which is brought by increased chances of network coding in the proposed scheme.

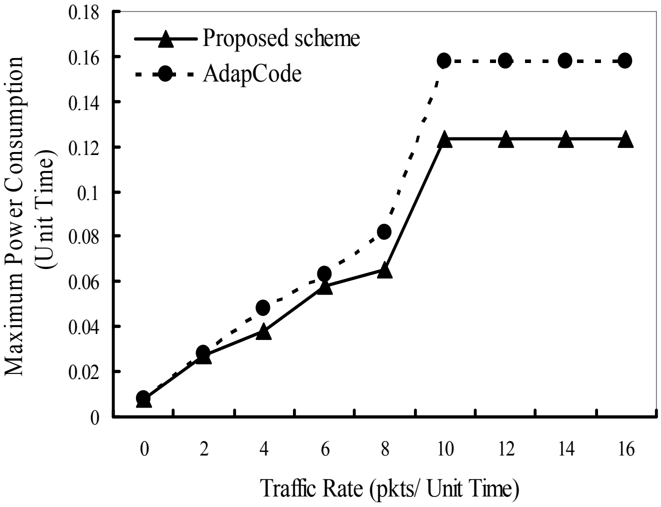

Figure 5. Comparison of the power consumption.

V. CONCLUSIONS

This paper proposed enhanced AdapCode. It was shown that our proposed scheme can guarantee the same reliability while reducing the power consumption. Power efficiency directly affects battery life and thus is a critical design parameter for WSNs. Actually; it just needs a slightly increased computational cost and a neglectable amount of power for neighbour discovery to get the power saving for the whole network in return. This feature is important in sensor networks since it can save the power for the whole network and thus increase the network lifetime.

REFERENCES

[1] F. Akyildiz et al., “Wireless Sensor Networks: a survey,” Computer Networks, vol.38, 2002, pp. 393-422.

[2] H. Karl and A. Willig, “A short survey of wireless sensor networks,” Technical Report TKN-03-018, Telecommunication Networks Group, technical University berlin, October, 2003.

[3] D. Chen and P.K. Varshney, “QoS Support in Wireless Sensor Networks: A survey,” in International Conf on Wireless Networks, 2004.

[4] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. Kaseva, M. Hannikainen, T.D. Hamalainen “Energy-efficient neighbour discovery protocol for mobile wireless sensor networks,” in Journel of Ad Hoc Networks 7 , december, 2007.

[5] M. Haenggi, “opportunities and challenges in wireless sensor networks,” in M. Ilyas, I. Mahgoub (Eds), Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, CRC Press, Florida, 2004, pp. 1.1-1.14.

[6] R. Ahlswede, N. Cai, S. -Y.R.Li and R.W.yeung, “Network Information Flow,” IEEE Trans. on Info. Theory, vol.46, July 2000, pp. 1204-1216.

[7] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “Tag: a tiny aggregation service for ad-hoc sensor networks,” in Proc. of USENIX OSDI, december, 2002.

[8] L. S. Shan, Z.P. Dong, L X. Ke, C.W. Fang and p.S. Liang, “Energy Efficient Multipath Routing Using Network Coding in Wireless Sensor Networks,” in ADHOC-NOW, LNCS 4104, 2006, pp. 114-127.

[9] I-H. Hou, Y-E. Tsai, T F. Abdelzaher and I. Gupta, “AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks,” in Proc. of IEEE INFOCOM, 2008.