EAQR: Energy Aware QoS Routing in Wireless Industrial Sensor Networks

K. Vinoth Kumar

Abstract—Most techniques for wireless sensor networks can be applied to wireless industrial sensor networks. However, for industrial applications of wireless industrial sensor networks, new requirements such as real-time, reliable delivery need to be considered. In this paper, we propose EAQR, which is a novel routing protocol for wireless industrial sensor networks. It provides real-time, reliable delivery of a packet, while considering energy awareness. In EAQR, a node estimates the energy cost, delay and reliability of a path to the sink node, based on information from neighboring nodes. Then, it calculates the probability of selecting a path, using the estimates. When packet forwarding is required, it randomly selects the next node. A path with lower energy cost is likely to be selected, because the probability is inversely proportional to the energy cost to the sink node. To achieve real-time delivery, only paths that may deliver a packet in time are selected. To achieve reliability, it may send a redundant packet via an alternate path, but only if it is a source of a packet. Experimental results show that EAQR is suitable for industrial applications, due to its capability for energy efficient, real-time, reliable communications.

Index Terms—Energy aware routing, industrial control, real-time and reliable communication, wireless industrial sensor networks.

I. INTRODUCTION

Wireless industrial sensor networks (WISNs) are used to collect data from a machine equipped with sensor nodes, and forward data to the sink node; they are generally used for industrial control applications. The sink node is connected to a control system that obtains data via the sink node, and controls actuators in a machine, or alerts users as a result of data analysis. WISNs can provide lower cable costs, and easy setup and maintenance [1] for existing industrial applications.

WISNs are useful for factory automation involving a variety of applications, such as an industrial process control predictive maintenance of a machine [2], industrial linear position. Machine vision is an application of computer vision for industrial process control. Machine vision involves input/output devices and control networks, which can be replaced with WISNs. Machine vision can improve the quality of manufacturing, by searching for product defects using information from WISNs. Predictive maintenance can improve productivity by monitoring and assessing the health status of a piece of equipment in a machine. Industrial linear position is an industrial application for measuring displacements of moving parts.

In WISNs, it is necessary to consider new QoS requirements, such as real-time, reliable communication [1]. In WISNs, sensing data from sensor nodes must be transmitted to the sink, reliably and in time. Delayed or lost data may cause industrial applications to malfunction, because the sensing data is analyzed and appropriate commands are sent to the actuator of a machine. Machine vision for industrial process control uses a multimedia stream, which is more affected by jitter or packet loss than scalar physical phenomena. A delayed packet may be useless, and it is difficult to obtain meaningful information from decoding packets if the rate of packet loss exceeds a threshold. Soft real-time communications are acceptable in WISNs, because of fault tolerance, which is one of the major advantages of WISNs. Though some packets are lost or delayed, others may be transmitted continuously, or transmitted via another path. Applications such as machine vision require soft real-time Communications for the multimedia stream, where a slight delay or a small rate of packet loss is tolerable.

Routing protocols for WISNs must be carefully designed to consider resource constraints such as low processing power, small memory and limited energy of sensor nodes [3]. In addition, WISNs must be scalable and able to tolerate dynamic network changes. They range from tens to thousands of sensor nodes. Therefore, the complexity of the routing algorithm must be independent of the size of the network or the number of sensor nodes. They would be impractical if memory utilization increases as the number of nodes increases. New nodes may be newly deployed, or some nodes may disappear, due to malfunctions. Therefore, routing algorithms must adapt to dynamic network Changes.

WISNs may use routing protocol for wireless mobile ad-hoc networks or wireless sensor networks (WSNs), because they have similar wireless networks characteristics. In several studies real-time, reliable communication has been studied for wireless mobile ad-hoc networks. However, there were problems in applying these studies to WISNs, because the scalability of WISNs or constraints of sensor nodes were not considered. There have been studies about real-time, reliable communication in WSNs [4]. Most of these studies do not provide simultaneous real-time, reliable and energy aware communication. The aforementioned WISNs require simultaneous real-time, reliable and energy aware communication. Therefore, it is necessary to design a routing protocol that can provide real-time, reliable communication, and energy awareness in WISNs.

In this paper we propose EAQR, which is designed to achieve the aforementioned requirements in WISNs. It provides real-time, reliable delivery of a packet, while considering energy. Especially, EAQR can set the packet...
reliability. Redundant packets can be used to prevent packet loss in real-time communications. However, the number of packets in networks increases, due to redundant packets. Due to the increased number of packets, there can be congestion or increased energy expenditure. Therefore, setting the reliability of a packet is essential, so that the user can achieve a trade-off between energy and reliability. EAQR estimates the expected values of the energy cost, delay and reliability of a path to the sink node. These values are computed using only information from neighboring nodes. Based on these values, EAQR selects a path that requires low energy, low delay and provides high reliability. For an even distribution of energy expenditure, sometimes EAQR selects a non-optimal path in terms of energy expenditure, but can still deliver a packet in time. This paper provides a simple approximation of the minimum delay [6].

The remainder of the paper is organized as follows. Section II describes the proposed routing protocol, EAQR. Section III presents the performance of EAQR compared to the previous protocols. Finally, concluding remarks are provided in Section IV.

II. PROPOSED ROUTING PROTOCOL

EAQR is a kind of proactive routing protocol that aims to maintain an ongoing routing table. As in other kinds of proactive routing, EAQR constructs and maintain a routing table with information from neighboring nodes. A beacon message is used to exchange information related to routing among neighboring nodes. The actual path is decided while transmitting a packet.

There are two types of messages: beacon messages and data packets. A beacon message is exchanged among neighboring nodes to construct and maintain a routing table. Upon receiving a beacon message, a routing table is constructed or updated by calculating expected values of energy cost, delay and reliability. When a path to the sink node becomes known to a node, the node begins to send a periodic beacon message. The source node sends data packets to the sink after constructing the routing table. Each intermediate node forwards a data packet to a neighboring node that can deliver the packet in time. A neighboring node for forwarding a packet is selected based on the expected delay and probability. This probability is inversely proportional to the expected energy cost of neighboring nodes. Therefore, a path that may expend less energy than other paths is most likely to be selected. To ensure reliable packet delivery, if the expected reliability of the selected node does not satisfy the required reliability, the source node selects an additional neighboring node to forward the packet.

A. Beacon Message and Routing Table

Every node exchanges a beacon message to construct and maintain a routing table of a node. A beacon message contains expected values such as energy cost ($C_i$), time delay ($T_i$), reliability ($R_i$), and residual energy ($B_i$) of a node. $C_i$ is the expected energy cost of sending a packet from node $i$ to the sink node, is the expected time of sending a packet from node to the sink node. $R_i$ is the expected reliability of sending a packet from node $i$ to the sink node. The reliability is the probability of sending a packet to the sink node without error. $C_{sink}=0$, $T_{sink}=0$ and $R_{sink}=0$ at the sink node and the expected values of the sink node are constant [5]. The beacon message also contains the position of node $i$. EAQR assumes that every node knows its own position and that of the sink node. The location information can be obtained by GPS or localization protocols for estimating the location of a node [7].

The expected values of a node may change as the residual energy of a node decreases and the state of the network changes dynamically. Once a node obtains a path to the sink node, it broadcasts a periodic beacon message to its neighboring nodes, to inform them of the change of expected values. Because the expected values of the sink node are constant, the sink node only sends a beacon message while initiating network setup and receiving an empty beacon message. An empty beacon message is a beacon message that contains nothing. Whenever a node receives an empty beacon message from another node, it responds to the node with a beacon message. A new node collects routing information by broadcasting empty beacon messages to its neighboring nodes. It constructs its own routing table with a beacon message from its neighboring nodes. When a node receives a beacon message from a neighboring node, it only adds the neighboring node to the routing table if the neighboring node is closer to the sink node than it is. If the neighboring node is

![Fig.1. Example of neighboring nodes in the routing table of node i](image)

Already in the routing table, it only updates the expected values of the neighboring node. Fig.1 shows an example of neighboring nodes in the routing tables of node $i$. Nodes in the area are neighboring nodes of node $i$. They can communicate with node via a wireless channel. Area may not be a perfect circle, because each node may have a different wireless communication range or there may be things interfering with wireless communication, such as the machine shown in the figure. In a wireless industrial environment, this kind of noise source must be considered.

Area B is a circle for which the center is at the sink node and the radius is the distance between node $i$ and the sink node. Nodes in area B are closer to the sink node than node $i$ because node is at the edge of area B. The neighboring nodes located at the intersection of areas A and B (shaded area of the figure) can be added to the routing table of node $i$.

When the expected values of neighboring nodes are updated by a beacon message, the routing probability ($P_{ik}$) is
updated $Ti$, $Ci$ and $Ri$, and are recalculated. When node $i$ receives a beacon message from neighboring node $j$, node $i$ updates these expected values as follows [5].

The beacon message contains $Cj$, $Tj$, $Rj$ and $Bj$

\[
\begin{align*}
C_{i,j} &= Cj + Ei, j \\
T_{i,j} &= Tj + Hi, j \\
R_{i,j} &= Rj + Li, j
\end{align*}
\]

where $Cj$ is the expected energy cost of sending a packet from node $i$ to the sink node via node $j$. $Ti,j$ is the expected time delay of sending a packet from node $i$ to the sink node via node $j$. $Ri,j$ is the expected reliability of sending a packet from node to the sink node via node $j$. $Ei,j$, $Hi,j$ and $Li,j$ are the energy cost, average time, and link strength for single hop communication between node and node $i,j$, respectively.

For node $k$ in $RT$—which is the routing table of node $i$—

\[
P_{i,k} = \frac{1/C_{i,k}}{\sum_{m \in RT} 1/C_{i,m}}
\]  

$P_{i,k}$ is the probability that node selects node to forward a packet. Therefore, a neighboring node with a lower energy cost is more likely to be selected.

Here, the expected values of node can be obtained as follows:

\[
C_{i} = \sum_{k \in RT} P_{i,k} C_{i,k}
\]  

\[
T_{i} = \sum_{k \in RT} P_{i,k} T_{i,k}
\]  

\[
R_{i} = \sum_{k \in RT} P_{i,k} R_{i,k}
\]  

\[
P’_{i} = \frac{1/C_{i}}{\sum_{k \in RT} 1/C_{i,k}}
\]

Fig. 2 presents a graphical example of expected values. The nodes in the region indicated by the dotted line are neighboring nodes in the routing table of node $i$. Each neighboring node $(k,j,l$ and $m)$ has its own expected values for sending a packet to the sink node.

**B. Node Selection for Forwarding a Packet**

When a node finds a path to the sink node, and a data packet is ready, a sensor node begins to send data packets received from other nodes, or its own data packets obtained from sensing. The deadline and reliability, $\epsilon$ of a packet may be predefined by user or determined by nodes at every transmission. The deadline is a relative deadline, which is the tolerable delay of delivering a data packet to the sink node. The reliability, $\epsilon$, included in a packet is the desired reliability, which is between zero and one that means that no degree of reliability is required, whereas means that a high degree of reliability is required. The laxity, $\epsilon$, which indicates the residual time until the deadline, is embedded in a data packet and recalculated at every node along a path to the sink node. EAQR selects the next node to forward a packet, based on the laxity of a packet and the expected values of neighboring nodes [9].

A path to the sink node is constructed during packet transmission. A node—including the source node—selects the next node, according to the following rules.

1) Select nodes in the routing table which can deliver a packet. Within the required deadline.

2) Calculate the probability $P’_{i,j}$ based on the $RT’$. For every node $j$ in $RT’$

\[
P’_{i,j} = \frac{1/C_{i,j}}{\sum_{k \in RT’} 1/C_{i,k}}
\]

3) Randomly select the next node by the probability $P’$ [5]. If a node is a source node $i$ and of the selected node $Rj$ is less than the required reliability $R$, randomly select one additional next node by the probability $P’$.

The selection algorithm based on probability prevents energy loss of nodes on the optimal path with the least energy cost, by distributing the load to other nodes on a non-optimal path. A maximum of two packets are sent to achieve reliability, according to the third rule. This is because the algorithm is simple, and if there was more than two paths this may result in congestion of networks, due to too many redundant packets [7].

**III. PERFORMANCE EVALUATION**

GloMoSim [8] was used to evaluate the performance of EAQR. This is a very fast, effective, discrete-event simulator for simulating wireless communications. It has a detailed propagation model, radio, and MAC layers. Table I describes the detailed simulation parameters. We used the two-ray path loss model for the radio propagation model in the simulations. The two-ray model considers both the direct path and the ground reflection path. In the simulations, we used the signal-to-noise ratio threshold (SNRT) model for the signal reception model. If signal-to-noise ratio (SNR) of a

---

*Fig. 2. Example of values for single hop communication between node $i$ and its neighboring nodes, and expected values of the neighboring nodes.*
packet exceeds the threshold, it receives the packet error-free. Otherwise, the packet is dropped [8]. We used IEEE 802.11 DCF for the MAC layer protocol.

EAQR and other protocols compared in the paper are table-driven routing protocols for static wireless networks such as wireless industrial sensor networks. The relative performance of these protocols for such networks is largely independent of MAC layer protocols such as CSMA, MACA, and 802.11 DCF. Therefore, simulation with a single common MAC protocol is sufficient for the performance comparison of the routing protocols in these networks. Every sensor node sent data to the sink node at an interval of 10 s. In EAR and EAQR [10].

a) Packet delivery ratio: The ratio of the number of packets received and the number of packets expected to receive.
b) Normalized control overhead: The total number of control message transmissions divided by the total number of received data packets. Each forwarding of the control message was counted as one transmission.
c) Normalized data packet transmission overhead: The ratio of the total number of data packet transmissions and the number of received data packets.
d) Joining delay: The average time interval between a member joining a group and its first receiving of the data packet from that group.
e) Multicast efficiency: It defined as the number of data packets delivered to multicast receivers over the number of total data packets forwarded. Higher value implies better performance.

Multicast Efficiency= total received packets / total forwarded packets

IV SIMULATION RESULTS

4.1. Average Power Conservation vs Time

The performance of average power conservation of EAQR is shown in the graph. EAQR is the modified on demand multicast routing protocol and SEAQR is the secure on demand multicast routing protocol. Compared to EAQR and MSEAQR, SEAQR is more secure and will give the more data delivery and less delay. The moving speed of nodes is uniformly set between the minimum and maximum speed values which are set as 1 m/s and 20 m/s, respectively. IEEE 802.11b was used as the MAC layer protocol. Fig 3 shows the simulation results of Average power conservation Vs Time.

4.2. Speed vs Delivery Ratio

In EAQR, the mesh structure is built on the source’s demand, and a source sends out a JOIN QUERY message periodically to refresh the mesh structure. If the nodes want to join a group, they need to wait until the next mesh refreshing period. The refreshing interval is set as 3 seconds. Fig 4 shows the simulation results of Speed Vs Delivery ratio.

4.3. Byte Sent Byte Delivered vs Speed

Using EAQR we can receive the more number of data. It is critical and challenging for a multicast routing protocol to maintain a good performance in the presence of node mobility in an ad hoc network. We evaluate the protocol performance by varying maximum moving speed from 5 to 40 m/s. Fig 5 shows the simulation results of Byte sent byte delivered Vs Speed.
4.4. Time vs Throughput

By varying the time period from 0 to 50 m/s and we can analyse the throughput. As the time period increases the throughput also increases in EAQR. But compared to EAR, EAQR has highest through put. It delivers the 95% of data send by the source. Fig 6 shows the simulation results of Time Vs Throughput.

![Fig.5. Byte sent Byte delivered vs Speed](image1)

![Fig.6. Time vs Throughput](image2)

V. CONCLUSION

This paper proposed EAQR, an energy aware routing protocol for real-time, reliable communication in WISNs. EAQR provides real-time communication without compromising the energy awareness of the existing energy aware routing protocol, EAR. It selects a path that expends less energy than others, among paths that deliver a packet in time. Sometimes, it selects a path that expends more energy than the optimal path, because the path is selected at random, according to a probability. This enables even distribution of energy expenditure to sensor nodes. In addition, EAQR provides efficient, reliable communication, because it only sends a redundant packet via an alternate path if the reliability of a path is less than a predefined value. The deadline, which is the maximum tolerable packet delay, must be carefully selected. The deadline must be the same as or longer than the minimum network delay. This paper estimated the minimum delay, to select a deadline given the density of sensor nodes and radio range. Simulation results showed that EAQR performs better than existing QoS routing protocols, in terms of reducing the number of packets that missed deadlines or were lost, while considering energy awareness.

REFERENCES


**K. Vinoth Kumar** received the B.E. degree in electronics and communication engineering from the Kurinji College of engineering and technology, Manapparai, Anna University, Chennai, India, in 2009. He received the M.E. degree in Applied electronics from the J.J College of engineering and technology, Trichirappalli, India, in 2011. Currently doing Ph.D. in communication and Networking in Karpagam University Coimbatore. His research interest includes wireless communication, Mobile Ad hoc networks, Sensor Networks, Communication networks.