Global Routing with Energy Balancing in Intra-Vehicular Environment

Gill R. Tsouri\textsuperscript{1} and Moshe Laifenfeld\textsuperscript{2}
\textsuperscript{1}Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology
\textsuperscript{2}Advanced Technical Center - Israel, General Motors

Abstract—Global routing in vehicular sensor networks is considered with the aim of balancing energy consumption across the nodes to achieve longer network lifetime. To this end, a routing protocol based on Dijkstra's routing algorithm with an augmented link cost function is used. Performance of the routing protocol is evaluated using a hardware experimental setup comprising 8 nodes positioned throughout the car and an access point placed near the dashboard. Real-time experiments implementing the routing algorithm are performed. Results demonstrate efficient balancing of energy consumption across all nodes in the network and a substantial increase in network lifetime.

Index Terms—Global Routing, Vehicular Sensor Networks.

I. INTRODUCTION

As the number of peripherals (sensors, actuators and switches) in a modern vehicle continues to grow rapidly, their wiring becomes a significant challenge to vehicle designers and manufacturers. It is estimated that a modern sedan has in total more than 3Km of wires. Reducing wires by wireless technology means, for instance, can potentially reduce weight, ease manufacturing and reduce warranty issues.

Potential wireless nodes in a Vehicular wireless Sensor Network (VSN) are required to operate under strict resource constraints. Specifically, power for transmitting data from nodes (sensors, switches and actuators) to Electrical Central Unit (ECU) must be preserved so that battery life is extended and recharging is infrequent as possible. VSNs operate in a unique signal propagation environment including shadowing, multipath propagation and relatively short distance links. An efficient routing protocol is crucial for extending battery life peripherals. It follows that efficient routing protocols should specifically be tailored to the VSN environment and requirements.

Several wireless technologies for the purpose of vehicle wiring reduction were studied in the past (see for instance [1]-[4]). However, it is only recently that network layer aspects were considered. In [5] the authors study experimentally the Collection Tree routing Protocol for intra-vehicle peripheral communications, showing performance gains in terms of packet delivery rates as well as power consumption over the traditional star topology networks. Power consumption is crucial for the VSN application as removing the power wire to remote peripherals is very much desirable. The required long lifetime of the VSN poses stringent requirements on the energy consumption of each of the peripherals, and the network as a whole. The study in [5] considers average power consumption of the network as a whole without a particular concern of a single peripheral or the lifetime of a VSN. In this study the power consumption and network lifetime are inherently considered in the design of the network protocol. Furthermore the routing protocol suggested in this study is global unlike the distributed protocol in [5].

Global routing algorithms are usually avoided in Wireless Sensor Networks (WSNs) where hundreds to thousands of nodes comprise the network and great distances are covered. The exponentially increasing complexity with the number of nodes and the need to aggregate at the ECU link-state information from all nodes in the network make such algorithms prohibitive. Unlike WSNs, a common network architecture used in VSNs is a star-topology, where an ECU is placed up to 3 meters from any node. The ECU collects data from the nodes and passes it to central processing units embedded in the vehicle. The ECU has access to substantial computation power, battery power and memory space compared to the simpler peripheral nodes. The asymmetric star-topology architecture coupled with the relatively low number of peripheral node, and the static nature of the network (nodes do not join or get removed on a regular basis) makes global routing protocols a viable option for collecting data at the ECU.

Recently, a global routing protocol based on Dijkstra’s algorithm was proposed and evaluated in a Wireless Body Area Network (WBAN) setting [6,7]. The proposed protocol in [6,7] uses a link cost function specialized for balancing energy consumption and increasing Network Lifetime (NL). NL is defined as the time it takes a single component of the network to deplete its power source from network startup. This approach views the nodes’ batteries as a distributed network resource. It follows that no single node should deplete its battery while there are other nodes with available energy. The link cost function was designed to ensure that all nodes deplete their battery at the same time and less often thereby increasing NL and reducing maintenance requirements associated with recharging batteries.

The modified Dijkstra’s algorithm makes use of a dynamic link cost function. The link cost changes periodically to reflect the change in energy usage across the nodes. Specifically, the link cost is designed to make sure that nodes with relatively low remaining energy are avoided as relays for other nodes.

There are striking similarities between the system constraints imposed by WBANs and VSNs operating environments. In both cases the signal is exposed to shadowing and fading, the nodes are required to be compact, simple and long lasting. The purpose of the work reported
herein was to explore the feasibility and benefits of the WBAN modified global routing protocol reported in [6,7] in a VSN environment. We found that the protocol scales well from the human-body environment to a vehicular environment. Experiments proved that the modified protocol is efficiently implemented in real-time, demonstrated efficient energy balancing and resulted in increased NL.

II. BALANCING ENERGY CONSUMPTION

When applying a conventional approach to power-efficient routing, the power required to transverse a link is used as the link cost. As a result the routing path from each node to ECU is the one which requires the least amount of accumulated energy across the nodes in the path. A probable outcome would be that a single node would deplete its power source before all others, thereby ending network lifetime while other nodes still have energy to use. In the link cost function of [6,7], the accumulated energy used by each node is factored in the link cost. If a node used more energy than its counterparts, its use as a relay for other nodes will be discouraged by increasing costs of its outgoing links.

As in any global routing protocol, link-cost information is periodically gathered at the ECU in the form of channel attenuation for each link in the network, and all routing calculations are performed at the ECU. Each node’s normalized energy used thus far is calculated as shown in eq. (2), where \( j \) denotes the node ID, \( i \) is the current polling round, \( a_{j,k} \) is the channel attenuation for the selected link, and RSSI \( T \) is a predefined global target Received Signal Strength Indicator (RSSI) needed to achieve the required performance level [7].

\[
a_{j,k} = \frac{\text{RSSI}}{P_{tx}} \quad (1)
\]

\[
E_{(i)}^{j} = E_{(i-1)}^{j} + \frac{\text{RSSI}}{a} \quad (2)
\]

The accumulated energy used is incremented by the energy used to transmit a single packet while maintaining RSSI \( T \). The channel attenuation for the selected link between node \( j \) and node \( k \), \( a_{j,k} \), is noted in eq. (1), where RSSI is the received power measured by node \( k \) and \( P_{tx} \) is the transmitted power used by node \( j \). Note that \( P_{tx} \), as defined in eq. (1), is a simple power control mechanism ensuring use of minimum required power to transverse a single link while meeting target RSSI. The link cost between node \( j \) and node \( k \), \( C_{j,k} \), is computed by calculating the energy that would be used by the node if that link is selected, and multiplying it by a cost factor as depicted in eq. (3) [7].

\[
C_{j,k}^{i} = \frac{\text{RSSI}}{a_{j,k}} \times \left( 1 + \frac{E_{(i)}^{j}}{P_{min}} \right)^{M} \quad (3)
\]

The cost factor is derived by dividing the accumulated energy used by the destination node, \( E_{(i)}^{k} \), with the minimum accumulated energy across all nodes, \( E_{min} \). This ratio is then raised to the power of \( M \geq 0 \), which determines how strong the effect will be. If a node’s energy is much greater than the current minimum, it will be avoided as a relay for other nodes, because its outgoing link cost would be very high. The cost factor is normalized so when \( M=0 \), it reduces to the conventional cost function which is the power required to transverse the link regardless of accumulated energy use across nodes in the network [7].

III. EXPERIMENTAL SETUP

In order to evaluate the efficiency of the cost function, it is compared to a reference system using a conventional link cost function, where the link cost is the required power to meet a link with the desired RSSI \( T \) \( (M=0 \) in eq. (3)). NL is evaluated by measuring the time it takes any node’s accumulated normalized energy to cross an arbitrary threshold. The threshold represents the amount of energy stored in a device battery. In all experiments the network was comprised of an ECU and 8 End Devices (EDs) acting as nodes. The positioning of EDs and ECU in the car is depicted in Fig. 1, where the ECU is noted as the Access Point (AP) gathering information from the 8 nodes.

The experimental setup was based on a hardware platform with Texas Instruments (TI) EZ430-RF2500. This device includes both an MSP430F2274 microcontroller along with a CC2500 2.4GHz transceiver. The CC2500 is configured to run at 250kbps. A single device, labeled as AP (the ECU), is connected via a USB to Serial link, running at 115200 BAUD, to the host computer. All other devices, labeled as EDs, are battery powered. In this implementation, the ECU acts only as a bridge between the host and the end devices. All routing and power control calculations are done on the host. The host is a laptop computer with an Intel(R) Core(TM) i5-2410M CPU and 4GB of RAM.

Eight EDs were placed throughout the vehicle (GMC Escalade) as depicted in Fig. 1. The host computer was placed near the dashboard and connected to the ECU via a
USB cable. The target RSSI was arbitrarily chosen to be $-60 \text{dBm}$.

The experimental setup was used to implement the routing algorithm in real-time in response to the changing channel conditions. The following procedure was carried out at a rate of 5Hz:

1: ECU sends synchronization beacon which includes routing and power control tables.
2: Each ED transmits its own RSSI table back to the ECU, while simultaneously listening to other ED messages and storing the received power from each.
3: Once all EDs have transmitted their data, the ECU sends a table with the RSSI data from all devices.
4: The host uses the RSSI table to compute the routes along with the required powers to meet the selected links using Dijkstra’s algorithm and eq. (1-3).
5: The host sends both routing and power tables back to the ECU so that a new cycle may begin.

To minimize the number of control packets being transmitted, the EDs are not individually polled. The only control packet sent is the synchronization beacon, which also carries the routing and power tables. Once the EDs are synchronized, they transmit their data on a pre-defined schedule to avoid collisions. Each ED has a network ID. The time between synchronization packets is divided into time-slots, where each slot is used by a single ED. The time slot used depends on the network ID of each device. This avoids the need for scheduling during runtime.

The routing table is a simple array which lists the destination for each ED packet. The ED does not need to know the entire route its packets will take, but only the next device in the path. Similarly, the power table lists the transmit power setting each device needs to use. The size of these tables is directly proportional to the number of devices in the network.

The traditional Dijkstra’s routing algorithm was run for 1500 polling rounds (5 minutes long). The modified Dijkstra’s routing algorithm was then run for the same duration using various balancing parameter values.

![Fig. 2 – Sensor positioning with rigid routing tree for traditional Dijkstra’s algorithm](image1)

![Fig. 3 – Accumulated transmitter normalized energy across polling rounds for traditional Dijkstra’s algorithm](image2)

![Fig. 4 – Accumulated transmitter energy across polling rounds for modified Dijkstra’s algorithm](image3)

IV. RESULTS

As expected, the traditional algorithm resulted in a rigid routing tree. A representative result is depicted in Fig. 2. The corresponding energy usage of the eight EDs is depicted in Fig. 3. It is clear that there is a disparity of energy use and that a single ED (Sensor 1) would deplete its battery first, thereby ending network lifetime while there is still energy in the network. This makes sense since Sensor 1 carries the heaviest relaying load: it relays packets from both Sensors 2 and 3.

A representative example for energy usage for the modified Dijkstra’s algorithm is depicted in Fig. 4 for $M=50$. In contradistinction to Fig. 3, the sensors are making balanced use of their energy and it is expected that all sensors would deplete their batteries at the same time. Note that at the end of the experiment all nodes used under 2.5 units of normalized energy where in the traditional algorithm, Sensor 1 used 8.5 units. This result coupled with the linear increase in energy use implies that the modified algorithm would have resulted in a network living 8.5/2.5=3.4 times longer. Another advantage is that all batteries would be depleted simultaneously which would require less frequent maintenance for replacing batteries.

V. CONCLUSION

The modified Dijkstra’s algorithm protocol originally designed for WBANs was shown to scale well to the VSN environment. VSNs requiring all sensors to be active in order to function would benefit from this routing method. Due to the balancing of energy use in the network, sensors...
will deplete their energy sources at approximately the same time. This offers an additional benefit when considering maintenance of the network as all sensors can be recharged or replaced simultaneously, instead of constantly monitoring and replacing individual sensors.

One possible course of action for future research is to incorporate the modified Dijkstra’s algorithm into the backbone vehicular network. If multiple ECUs are connected to a single bus and each such unit is wirelessly connecting to its own set of sensors, two possibilities emerge: the first is to implement the algorithm at each unit. The second possibility is to include branches to all units from all sensors, thereby achieving spatial diversity gain per sensor. Link costs could also be modified based on other criteria such as congestion at each unit, latency and quality of service requirements.

Another extension is to consider regenerated energy sources, i.e. energy harvesting nodes. In this approach each node has a different recharging rate on top of its current battery state. Estimating the recharging rate and including it into the routing scheme can potentially enhance network performance considerably.

REFERENCES


