Energy Efficient Approach for ConverSS: Routing Protocol for Wireless Networks

Snehal J Koparde
Department of Electronics & Telecommunication Engg.
Dr. Babasaheb Ambedkar Technological University,
Lonere, Raigad.

Anil B. Nandgaonkar
Department of Electronics & Telecommunication Engg.
Dr. Babasaheb Ambedkar Technological University,
Lonere, Raigad.

Laxman D. Netak
Department of Computer Engg.
Dr. Babasaheb Ambedkar Technological University,
Lonere, Raigad.

ABSTRACT

Recently Wireless Sensor Networks (WSNs) are a very promising research field since they find application in many different areas. In applications involving networks of sensor-equipped autonomous vehicles, it is crucial to have an energy-efficient communication protocol due to the limited on-board batteries. Unlike traditional sensor networks, vehicle sensor networks typically consist of only a small number of nodes. We exploit this fact in our protocol design by optimizing specifically for these mobile small scale networks. Our proposed solution, ConverSS, is a hybrid MAC/routing protocol that is energy-efficient for vehicle sensor networks. The results show that our schemes effectively support increased sleeping interval with low energy consumption.

Keywords

Wireless Sensor Network, Protocol design and analysis, Mobile communication system.

1. INTRODUCTION

WSNs (Wireless Sensor Networks) are systems of many low-cost, power-limited, and small size devices distributed deployed over an indoor or outdoor. Energy conservation is considered as most important factor in designing the WSNs. The nodes in WSNs are often expected to be position-aware since the sensed data without position information are meaningless in most cases. And in most of the WSN applications, most of the nodes are stationary. But the use of networked autonomous vehicles is of great value to many sensing applications such as environmental monitoring or disaster response. For such applications, the ability to dispatch these vehicles to a remote (and often hostile) location is critical. Since these vehicle applications are of a different nature than traditional sensor networks, a different approach should be taken to the networking and communication of such systems. Although we have described a specific example here, we believe this scenario covers the more generic set of vehicle based sensing applications. To define the vehicle based networking problem, we identified certain characteristics of such systems, that the cost to manufacture and deploy these vehicles is much higher than the cost for small static sensor nodes, thus limiting the scale of such networks. Therefore, scalability is not a requirement for the networking strategy, as is often assumed for traditional sensor networks. Instead, the need for scalability can be which, in some cases, set them apart from the well-studied traditional sensor networks. One characteristic of these networks is that they typically have a convergecast traffic pattern, in which many nodes report data to a single sink node. Another characteristic is that these networks have a dynamic topology, since the vehicles can be highly mobile. The impact on data routing is that routes expire quickly and the opportunity for route reuse is limited. Lastly, these networks are of a small scale, in which the number of nodes is on the order of ten to a few tens of nodes. The reason for such small network sizes is traded off for a more efficient solution. Our networking problem is to provide energy-efficient communication for small-scale, convergecast mobile networks. In our solution, we leverage the fact that these networks are small-scale for more efficient operation.

2. LITERATURE SURVEY

As ConverSS is a combined MAC and routing solution for reliable and energy-efficient convergecast for small scale, mobile networks which is designed specifically for cases where most nodes are one hop from the sink, and where mobility limits the opportunity for reusing routes[1]. Routing is usually achieved by constructing a routing tree prior to scheduling [2], but our approach differs in that our networks are mobile, and routes are discovered during data delivery. Directed diffusion routing [3] is data gathering by nature, in that individual nodes request and receive data from the rest of the network. However, directed diffusion is more efficient when routes are reused, so it is not optimized for our application. There has also been much study into routing for vehicular ad hoc networks (VANETs), but communication is typically not convergecast, and energy efficiency is not a primary focus for these automotive applications. Most routing approaches are position-based [4], [5], which rely on knowledge of the positions of the source, its neighbors, and the destination. Applications such as automotive collision avoidance flooding based techniques are used [6]. Here important performance metric is latency. Wang et al. [7] proposed an intelligent flooding technique for intervehicle communications (IVC), designed for general data services. This technique performs reactive routing approaches for small scale networks in terms of throughput and latency. Also TreeDMA protocol [8] designed for small scale convergecast network where routing tree is implicitly constructed during data delivery using beaconing and overhearing. But this protocol fails when packet errors and asymmetric links are introduced. ConverSS is robust to such real-world effects. For many of these real-time sensing applications, sensor generated data must be sent to a sink periodically. So in one cycle of protocol operation, or a sending interval, in which each node has one data packet to deliver. A sending interval occurs periodically with the data arrival period. Because these are small networks, it is feasible to use a fixed-assignment TDMA MAC, in which each node is assigned a dedicated time slot for sending. This MAC requires that the number of nodes in the network be fixed at the system initialization. However, this setup is sufficient for most sensing missions, which have a small, consistent team of vehicles. Time synchronization for the MAC is done using GPS, which has been used in existing systems and shown to achieve accuracy within 30 ns [9], [10].

2.1 Two-Phase Approach of Protocol

The idea behind ConverSS is that it functions primarily as a fixed-assignment MAC, but includes a fallback mechanism to accommodate multihop routing. We capture this idea in a two-phase approach (Fig. 1), which efficiently handles the most common cases in the first phase, and then if necessary, deals with all other cases in the second phase. Here considering same approach as per [1] but will do small change in sleeping time. We will test same for different values of sleeping interval. We will consider ConverSS protocol [1] with some changes which will show better results with respect to energy.

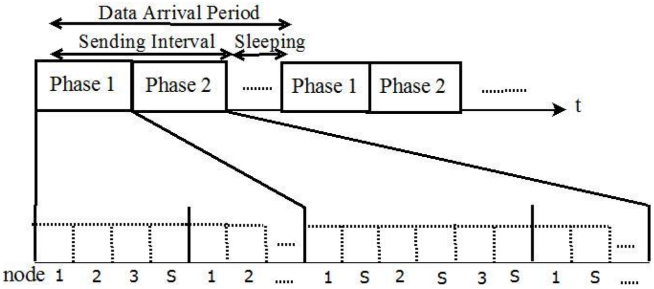

Fig 1: ConverSS two-phase approach.

In Fig. 1, we label the owner of each sending slot below every time slot, with the sink's slot indicated by an S. Phase 1 operates under the assumption that nodes are one hop from the sink, so each node attempts to send directly to the sink in its sending slot. Since there is no routing in this phase, nodes do not need to listen in the other nodes' slots. They can instead be placed in sleep mode, in which nodes turn off the radio and thus consume very low power. The sink follows by broadcasting a Delivery Status Message (DSM), which contains information on which nodes' packets have been delivered to the sink. All nodes listen to this slot to discover if their data have been delivered. In case of packet errors, more frames are allotted in which nodes can retransmit any undelivered data. By the end of Phase 1, packets from all 1-hop nodes should have been received by the sink. If this includes all nodes, as we expect in most cases, then they all go to sleep until the next sending interval. The operation described is efficient because there is no route setup, and nodes only send in their own slots and do not need to listen in the other slots. This will be the main difference with layered approaches, since a route must be established and the MAC will always assume that routing may be needed, resulting in idle listening. It is possible that there are still undelivered packets after Phase 1 because of packet errors or nodes being out of range of the sink. If that is the case, a second phase is necessary to do a best-effort data delivery. In Phase 2, any nodes whose packets were not delivered perform a type of controlled flooding, in which nodes broadcast their data and receiving nodes can rebroadcast to try to deliver it to the sink. The reason for using flooding rather than a route setup followed by data transmission relates to the presence of asymmetric links. Studies have demonstrated the problem of asymmetry in radio propagation, in which a link is stronger in one direction than in the reverse [11], [12]. In typical routing, only symmetric links can be used because a handshake is required before a packet can be sent over a link. Flooding, however, does not require a handshake, thus enabling the use of asymmetric links in routing the data to the sink. Since these are small networks, routes with asymmetric links may be the only ones available. Therefore, with more options for routing, data delivery has an improved chance of success. After the two phases have completed, the network can go to sleep until the next sending interval, when the next packet is ready for delivery. Given the small number of nodes, the two-phase sending interval is typically short enough such that the network topology will be stable during that time. Because no routes are assumed prior to Phase 1, the protocol is robust to changes in topology in between sending intervals. Only in rare cases will changes to the topology during the sending interval affect the delivery rate, and the system can recover from these by the next sending interval. The cost of such rare occurrences is small relative to the gains in energy efficiency that the protocol design affords.

2.2 Example of Simple Network

We first provide an example using the simple network topology shown in Fig. 2, in which all but one node have a one-hop route to the sink, and the remaining node is two hops from the sink. In Phase 1, the nodes transmit unicast packets to the sink in turn, and conserve energy by going to sleep in each others' slots. The sink receives the data from all but node 2 fails to transmit data. Again sink broadcasts this information in a Delivery Status Message. Nodes that receive the DSM, i.e., nodes 1 and 3, learn whose data have been received. Since node 2 did not receive the DSM, it attempts to retransmit in the remaining Phase 1 frames, but again it fails. In Phase 2, nodes whose packets have not been delivered (e.g., node 2) flood their data. Here nodes 1 and 3 have not received from node 2 since they were not listening in that slot in Phase 1.

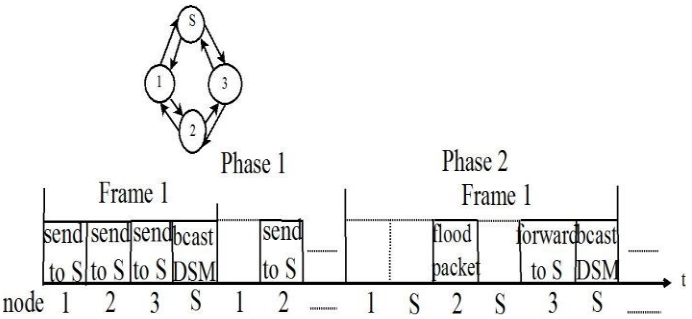

Fig 2: ConverSS example 1.

By receiving the DSM, nodes 1 and 3 know that node 2 does not have a route to the sink, so they listen in its slot. Nodes 1 and 3 receive the flooded packet, and node 3 forwards the message to the sink. Here in Phase 2, the frame structure changes and each sending slot is followed by the sink's slot, so that a DSM can be immediately sent. In this example, the sink sends the DSM after node 3, node 1 then learns that node 2's packet has been delivered, and node 1 can simply remove that packet from its queue. The DSM indicates that all nodes have delivered their data successfully and can go to sleep. Here we will use same protocol for testing and will additionally increase sleeping interval of above process. If sleeping interval is increased there should have less energy consumption. So firstly simulation will carried for different parameters of ConverSS, then will go for implementation of sleeping interval.

3. SIMULATION AND RESULT

To evaluate the performance of ConverSS, we implemented the protocol in the network simulator ns-2. For comparison purposes, we have also implemented other protocol. The second protocol that we will compare is a traditional Layered protocol. For routing, it utilizes a simplified version of one-phase pull directed diffusion [3]. Directed diffusion is a data-centric scalable routing framework that is commonly studied in sensor networks. We implemented a protocol, which expects only one type of data. The sink initiates the Layered protocol by flooding the network with an interest, which expires at the end of each sending interval. Upon receiving the interest, nodes set up gradients according to which neighbor it received from first, and packets are sent along those gradients toward the sink. When the sink has received all data, it floods the network with a message telling nodes to go to sleep. For the MAC, this protocol uses a fixed-assignment TDMA. Each node has a sending slot, and nodes listen in all other slots. Because we use a time-slotted MAC, nodes that are not sending or receiving do not need to be idle for an entire time slot. Instead, nodes can listen at the start of the slot, and if no header is received, nodes can sleep for the remainder of the slot. We handle idle listening in the same manner for our ConverSS protocol. In the simulation, we compared the protocols for various network sizes. Each node is given an error-free transmission range of 40meters. Nodes are mobile and deployed over a large field of 300x300m. Fig.3 and Fig.4 shows the average latency, and the energy consumed per packet. Latency of data packet measures time delay of packet from the source to the sink.

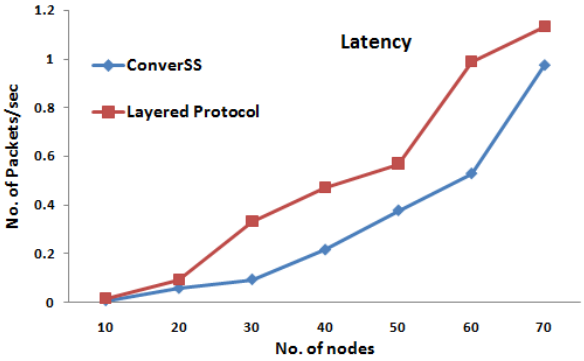

Fig 3: Latency per packet

We note that for network sizes up to 40-45 nodes, all nodes are in 1-hop range of the sink. We first comment on the results for these 1-hop networks, for which ConverSS is optimized. For ConverSS the latency is small since the data are delivered in the first frame, and the latency for Layered is a bit more because the gradient to the sink must first be established (Fig.3). For network sizes up to 40-45 nodes, all nodes are in 1-hop range of the sink so the energy consumption for ConverSS is less since all nodes immediately send to the sink and do not need Phase 2. ConverSS only consumes the less energy of Layered (Fig.4), since layered protocols pay the price of always listening in each others' slots, in case a route is needed. For network sizes greater than 40-45 nodes, we observe the performance as multihop routes are introduced. The latency increases for both the protocols (Fig.3), but it is greater for ConverSS due to the Phase 2 frame structure, which inserts a sink slot after every node's slot. Concerning energy, ConverSS consumes more energy than Layered (Fig.4) because nodes are flooding in phase 2. So here ConverSS actually becomes the less energy-efficient choice. This is due to the inefficiency of flooding when nodes get stretched out to have longer routes (Fig.4). This simulation illustrates that ConverSS is not meant as a generic solution for all types of networks. It is not designed for networks consisting of mostly multihop routes, but instead, it is meant for small networks, which consist of mostly one-hop nodes. So here concept is extended with different values of sleeping interval for ConverSS. Concerning energy ConverSS consumes less energy than layered protocol as shown in Fig.5 So ConverSS is energy efficient Mac protocol designed for convergecast small-scale network is proved.

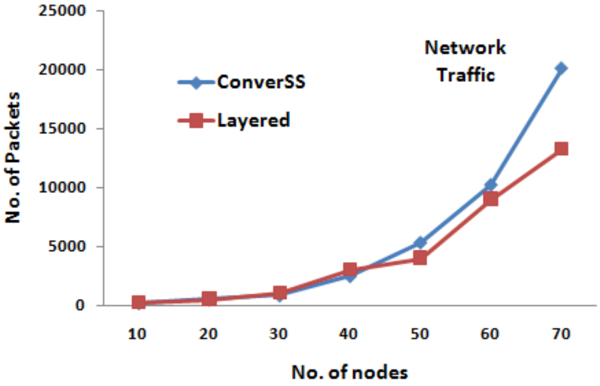

Fig 4: Network traffic per node

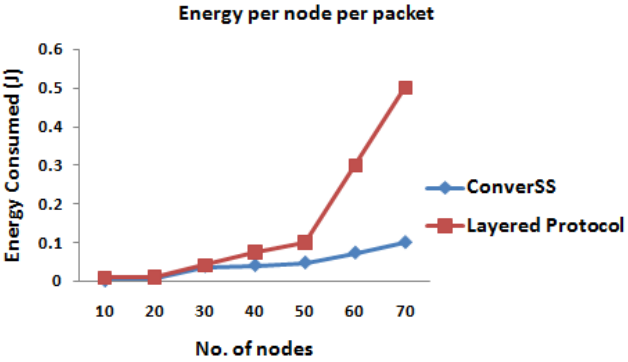

Fig 5: Energy per node per packet

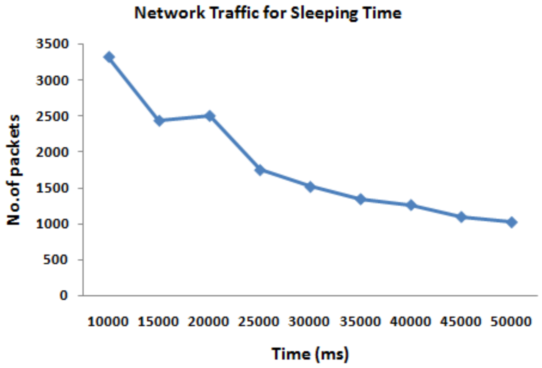

Fig 6: Network Traffic of Sleeping Time for 50 Nodes.

As ConverSS is concerned, if sleeping interval is increased then there will have better energy efficiency. So test is conducted for different values of sleeping intervals. From Fig. 6, If node size is 50 there were only 1023 packets found if sleeping interval is 50,000ms, but packets found are 3318 when sleeping interval is 10,000ms. Since network traffic is low, automatically energy consumed is low. Protocol does not work after 50,000ms, since it cannot finds routes to the sink. After the two phases have completed, the network can go to sleep until the next sending interval, when the next packet is ready for delivery. A sending interval occurs periodically with the data arrival period (Fig.1). So it is proved that protocol will be more energy efficient if sleeping interval is more.

4. CONCLUSION

In this paper we described, to enable efficient dissemination in small scale wireless sensor networks with nodes mobility. By designing a Protocol, we have achieved better performance for singlehop as well as multihop sensor-equipped vehicle networks. Also additional change with respect to sleeping interval for ConverSS protocol shows better results. So we gained energy efficient routing protocol for small-scale wireless networks.

5. REFERENCES

[1] Kam and schurgers, “Converss: a Hybrid Mac/Routing Solution for Small-Scale, Convergecast Wireless Networks,” IEEE Transactions on Mobile Computing, vol. 10, no. 9, Sept.2011.

[2] Y. Zhang and Q. Huang, “Coordinated Convergecast in Wireless Sensor Networks,” Proc. Military Comm. Conf. (MILCOM’05), 2005.

[3] F. Silva, J. Heidemann, R. Govindan, and D. Estrin, “Directed Diffusion,” Technical Report ISI-TR-2004-586, USC/Information Sciences Inst., Jan. 2004.

[4] B. Karp and H.T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” Proc. Sixth Ann. Int’l Conf. Mobile Computing and Networking, 2000.

[5] J. Luo and J.-P. Hubaux, “A Survey of Inter-Vehicle Communication,” Technical Report IC/2004/24, Ecole Polytechnique Federale de Lausanne, 2004.

[6] S. Biswas, R. Tatchikou, and F. Dion, “Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety,” IEEE Comm. Magazine, vol. 44, no. 1, pp. 74-82, Jan. 2006.

[7] S. Wang, C. Lin, Y. Hwang, K. Tao, and C. Chou, “A Practical Routing Protocol for Vehicle-Formed Mobile Ad Hoc Networks on the Roads,” Proc. Eighth Int’l IEEE Conf. Intelligent Transportation Systems, 2005.

[8] C. Kam and C. Schurgers, “TreeDMA: A Hybrid MAC/Routing Solution for Small-Scale Wireless Networks,” Proc. Third Int’l Workshop Information Fusion and Dissemination in Wireless Sensor Networks (Sensor Fusion ’09, Mobiquitous ’09), July 2009.

[9] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein, “Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet,” Proc. 10th Int’l Conf. Architectural Support for Programming Languages and Operating Systems, Oct. 2002.

[10] C. Zhu and M.S. Corson, “A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks,” Proc. IEEE INFOCOM, pp. 322-331, Apr. 1998.

[11] G. Zhou, T. He, S. Krishnamurthy, and J.A. Stankovic, “Models and Solutions for Radio Irregularity in Wireless Sensor Networks,” ACM Trans. Sensor Networks, vol. 2, pp. 221-262, May 2006.

[12] L. Sang, A. Arora, and H. Zhang, “On Exploiting Asymmetric Wireless Links via One-Way Estimation,” Proc. ACM MobiHoc, Sept. 2007.