Sustainability of Self-Configuring Wireless Sensor Networks

Bozena Kaminska
Simon Fraser University
School of Engineering Science
Burnaby, BC, CANADA
kaminska@sfu.ca

Pawel Gburzynski
University of Alberta
Department of Computing Science
Edmonton, AB, CANADA
pawel@cs.ualberta.ca

***Abstract*— Wireless Sensor Networks have recently become an important research area due to critical applications, e.g., in health monitoring and security. Those applications impose requirements for sustained, reliable, and fault-tolerant operation. We introduce a new ad-hoc wireless architecture in which forwarding is based on associativity and implicit dynamic gradients rather than explicit node-to-node connectivity. We show how our routing scheme forwards sensor data along fuzzy and intentionally redundant paths to provide for reliability and fault-tolerance.**

I. INTRODUCTION

Greater intelligence and independence from human intervention are the key attributes of future Wireless Sensor Networks (WSN). The enabling technology, comprised of the sensor nodes and the network infrastructure, requires employing advanced techniques in various areas such as MEMS sensors [1-3,9,10], microprocessors, memory, RF transceivers [4], smart antennas [5], routers, etc. New design concepts are needed to satisfy such requirements, as compactness, stability of the signal in the face of motion and other disturbances, durability and ultra-low power consumption. Wearable Sensor Networks [8] impose more constraints, such as long-term wearability with comfort, simplicity to use, reliable sensor attachment and fault tolerant functioning [2,6]. The target requirement is to have a **reliable and sustainable operation** of the WSN [7]. The factors of all environmental conditions, and associated perturbations need to be part of the definition of the WSN architecture and its modes of operation.

Consider a sample health-monitoring network. For the sake of illustration, suppose that the system's main role is to monitor the heart activity of people spread over a non-trivial (possibly open) area, with the intention of detecting and reporting anomalies. The issue of reliability and fault-tolerance occurs in several places. For example, the number and placement of electrodes on the human body may involve a degree of redundancy, to provide for sustained and reliable read-outs when some of the electrodes fall off or become disconnected. Then, the preprocessing software at the WNI (Wearable Network Interface) node interfacing the sensors to the wireless network may exhibit enough intelligence to cope with the disturbances in sensor read-outs resulting from the imperfect attachment of sensors to the body or their complete failure. At the other end, redundant data processing stations may provide for continuous attention of the interpreting agents despite hardware/software failures or intermittent connectivity problems. In this paper, we focus on the networking component and demonstrate how a certain "ad-hoc" approach to interconnecting wireless sensors may introduce a controlled amount of redundancy into the operation of forwarding data, as to render it self-healing and fault-tolerant in the face of blackouts and mobility.

A sensor network for health monitoring applications must balance the conflicting measures of cost, reliability and convenience. While the ad-hoc approach to organizing such networks may be attractive from the viewpoint of reachability, flexibility, and cost (because of their independence of infrastructure), the popular ad-hoc forwarding schemes promoted in the commercial world raise doubts regarding their resilience, robustness and reliability. The very term "ad-hoc" suggests sloppiness: few people would be comfortable betting their lives on an ad-hoc health monitoring system.

We demonstrate that ad-hoc networking for intentionally reliable sensor systems need not be tainted with "ad-hoc reliability." With our solution to low-cost, sustainable and non-intrusive monitoring, the communication paths carrying the critical sensor data are automatically safeguarded against failures of individual nodes and provide stable connectivity under node movement or occasional blackouts. This is in contrast to most of the commercially available systems (e.g., ZigBee®) where lost paths must be recovered from, which necessarily introduces episodes of uncertainty and unpredictability into the network behavior. With our approach, under reasonable conditions of general connectivity (meaning that the network is not fundamentally disconnected), paths are never lost and there is nothing to

recover from. This means that the streams of data flowing from (and to) the sensors are never disrupted. Node mobility and sporadic local failures or blackouts need not cause hiccups in the delivery of critical data. Such sustainable operation is achieved through self-configuring and automatic selection of redundant communication paths.

II. Classical Ad-Hoc Networking

The prevailing wisdom regarding the organization of an ad-hoc wireless network [11-14] assumes point-to-point communication, whereby each node forwarding the packet on its way to the destination sends it to a specific neighbor. Consider AODV [12], which lies at the heart of ZigBee. A node *S* initiating a packet exchange with node *D* broadcasts a request to its one-hop neighbors to start the path discovery operation. Based on its current perception of the neighborhood (neighborhoods are constantly monitored by all nodes) and cached information collected form previous path discovery episodes, a node receiving such a request may decide to forward it elsewhere, or respond backward with a ready path information intended for the initiating node *S*. Depending on the relative stability of neighborhoods (i.e., mobility and failures of nodes) and the availability of alive cached routes, the path setup bureaucracy may take more or less time. At the end, a single path between *S* and *D* has been established. This means that, when dispatching a packet to *D*, node *S* knows precisely which of its immediate neighbors the packet should be addressed to. Then, every node receiving such a packet knows the identity of the single next hop neighbor, and so on. A problem arises when the path is broken because such a mishap effectively demolishes the delicate structure: a new path recovery operation must be completed before communication can resume.

Generally, within the framework of point-to-point forwarding, redundancy can be implemented in two ways. First, multiple paths can be established for each end-to-end connection. Then, such paths can be used simultaneously (to avoid hiccups when one of them goes down), or one at a time (with failure recovery and a "quick" switchover to a stand-by path). Second, during path discovery, the intermediate nodes may try to internally store (cache) some information about alternative routes – to keep them handy when the primary route fails. Then, the operation of finding an alternative path may be quicker than establishing a new one completely from scratch. Note, however, that both schemes suffer from two problems. The more we try to avoid service interruption, the more resources we have to commit to the bureaucracy involved in keeping track of the alternative routes. The amount of this information per node tends to grow with the network size. Even in the ultimate case, when several paths are explored in parallel, those paths are still explicit and rigid, and their multiplicity gives us only some choice. If they cross at certain nodes, those nodes still remain a liability. If they do not cross, then their discovery and maintenance may be infeasible within the realm of small-footprint, possibly disposable, nodes.

III. Do We Need Paths at All?

In the wireless environment all transmissions are necessarily broadcast, and addressing a packet to a single neighbor does not help the fact that all other neighbors can hear it as well. Traditional schemes view this feature as a rather serious problem and try to defeat its negative consequences (like hidden or exposed terminals) via MAC-level tricks [15] facilitating point-to-point data exchange. As we argue elsewhere [16], the merits of such tricks are questionable in sensor networking, where individual packets tend to be short. Our opinion is that instead of fighting the inherent broadcast nature of a wireless channel, we should embrace it as a feature.

Suppose that node *S* wants to send a packet to node *D*. With our scheme, *S* simply transmits (broadcasts) the packet to its neighbors. A neighbor may decide to drop the packet (if it believes that its contribution to the communal forwarding task will not help) or retransmit it. This process continues until the packet reaches the destination *D*. An important property of this generic scheme (known as flooding) is that a retransmitted packet is never specifically addressed to a single next-hop neighbor. To make it useful, measures must be taken to limit the number of retransmissions to the minimum at which the desired quality of service is maintained. This part is implemented as a series of rules that determine when a node receiving a packet should rebroadcast it, as opposed to dropping. Some ideas for such rules are obvious, e.g., discarding duplicates of already received packets, limiting the maximum number of hops traveled by a packet. The key to the success of our variant of flooding is a rule that brings the paths traveled by forwarded packets down to a narrow (but intentionally fuzzy) stripe of nodes along the best route.

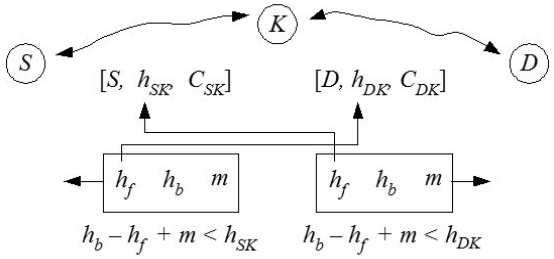

Figure 1. The suboptimal-path-discard rule.

Consider the three nodes *S*, *D* and *K* depicted in Figure 1. *K* is to decide whether an intercepted packet sent by *S* and addressed to *D* should be retransmitted or dropped. Its decision is based on simple calculations involving certain parameters cached by *K* and extracted from the packet headers.

Nodes learn about the sessions currently present in the network by monitoring packets passing by. Until the network learns about a particular session (understood as a pair of nodes that want to communicate), the forwarding for that session may be overly redundant. However, with time, the route will tend to converge to the "best" path. Note that this

mode of operation is not fundamentally less economical than one based on point-to-point forwarding. This is because no information comes out of the blue. Before a point-to-point system is able to commence forwarding, it too has to "learn" the routes some way. In our approach, the learning phase is not explicit, but occurs as an integral part of the "normal" forwarding scheme.

In addition to the identities of *S* and *D*, each packet traveling from *S* to *D* (as seen by *K*) carries the following information: h_f – the (forward) number of hops traveled by the packet so far and h_b – the (backward) target number of hops seen at the source *S* for a previous packet arriving from the opposite direction, i.e., from *D*. The latter parameter is acquired by *S* when it receives a packet from *D*. Note that as duplicates of already received packets are identified and promptly discarded, the first new packet that makes it to the destination usually has traveled along the shortest (or rather fastest) route between the two endpoints.

Owing to the inherent imperfections of the ad-hoc wireless environment, *K* should not be too jumpy with negative decisions. One possibility is to include a slack parameter *m* in the inequality – as shown in Figure 1. When $m > 0$, the rule will allow the node to forward the packet when the path passing to it appears to be longer (by up to *m* hops) than the currently believed shortest path. In essence, the slack parameter *m* determines the degree of redundancy in the population of nodes separating *S* and *D* and deemed responsible for packet delivery. When a packet departs at one end of the session, it does not know the exact path that it will travel. All it knows is that a certain (possibly fuzzy) subset of nodes will collaborate in the communal task of its delivery by pushing it in the direction of the destination.

The scheme can be augmented by simple heuristics that probe the network for new routing opportunities. For example, no matter how unsuitable a node appears to forward a packet, it may decide to do so every once in a while – just to check if the packet cannot be (better) delivered via an alternative segment of the network. Recoveries from natural lapses resulting from momentary local blackouts or reasonable mobility of nodes can be accomplished in a completely non-disruptive manner with the proper setting of the slack parameter *m*.

IV. Smooth Hand-Offs

To see how the network copes with mobility and node failures, consider the scenario shown in Figure 2. Packets traveling between nodes *U* and *V* are forwarded within the clouded fragment of the mesh network. For simplicity, we ignore whatever happens outside the cloud (the presence of other nodes along its boundary can only help). Suppose that the arrows represent neighborhoods, i.e., single-hop reachability. The path *A-B-C* (of length 2) is the shortest route through the cloud, thus, it will be considered the "yardstick" to which other nodes overhearing the packets forwarded by *A*, *B*, and *C* will calibrate their decisions. Suppose that *m* is modestly set to 1. This means that nodes *E* and *F* will also retransmit the packets because the route through them incurs a 1-hop increase over the best path. The worst that can possibly happen is the disappearance of node *B*, which is a critical component of the current best path. Note, however, that this disappearance will not disrupt the traffic because the second best path through the cloud, i.e., *A-E-F-C*, is also being used. The net outcome of this disappearance will be that a would-be duplicate arriving at *A* or *C* (from *E* or *F*), that would be discarded and ignored if *B* were in place, will be bona-fide received and forwarded towards the destination. After a short while, as the destinations update their h_b values in response to the increased number of hops along the best path, the nodes within the cloud will learn that *A-E-F-C* is the best path at this time. Subsequently, nodes *D* and *G* will add themselves to the population of forwarding nodes, as the path through them is only one hop worse than the current minimum. As we can see, assuming the changes are not too rapid or too drastic, they are accommodated without any disruption in service whatsoever. Note that with a higher setting of *m*, the network would be resistant to more drastic changes, at the cost of increased redundancy. An important property of that redundancy is its controllability. Note that *m* can be viewed as a dynamic parameter, adjustable by the nodes, whose value reflects the momentary priority of reliability over bandwidth or vice versa. Many sensor networks need little bandwidth. For them, *m* can be set high, still providing a methodologically motivated alternative to naïve and uncontrollably wasteful flooding.

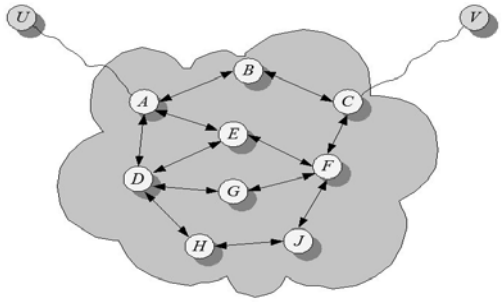

Figure 2. A hand-off scenario.

V. Application to Sensor Networks

The presented scheme constitutes the basis of TARP [16] a comprehensive add-hoc routing protocol for sensing applications. A typical network configuration involves one or more data collection points interspersed among a dynamic and possibly mobile collection of sensing nodes. Not all of those nodes have to be equipped with sensors: some may play the role of pure forwarders, although the difference is immaterial. A data collection point can be viewed as regular node (as it has to communicate with sensor nodes over the wireless channel).

Although most traffic in such a network flows from sensors to collection points, there is usually some demand for traffic in the other direction – to issue commands to the sensor nodes (switching them on and off, selecting data to

retrieve). In any case, the sessions tend to have a strongly asymmetric nature: typically, each of the collection points maintains a large number of sessions (with all their reporting sensor nodes) while a sensor node typically cares about a single session (with its target collection point). This asymmetry is OK with TARP. As long as a reasonable amount of traffic flows in either direction, the nodes along the way are able to fill their caches and take advantage of the rule illustrated in Figure 2. Typically, the collection points broadcast periodic beacons that fulfill a dual role: providing the heart beat and advertising the collection points to the sensing nodes, as well as filling their caches with the current value of h_{SK}, i.e., the number of hops separating them from the collection point.

VI. CONCLUSIONS

The unique feature of our proposed ad-hoc routing scheme is a method of introducing a controlled degree of redundancy into the width and fuzziness of forwarding paths. This redundancy puts the network component on par with the remaining parts of the entire system, i.e., the sensing equipment and the data-collection stations, for which natural reliability-enhancing techniques have been known. We believe that it will help us view the ad-hoc network as an integral part of the system, amenable to the same unified kind of sustainability control as the other components.

A wireless sensor environment with the networking component based on the concepts discussed above is being implemented for monitoring physiological parameters such as ECG, heart rate, respiration, and for activity monitoring using accelerometers and movement sensors. The personal WNI module handles 8 analog data lines from sensors sampling each of them at the rate of 500 samples per second. Then the digitized data are pre-processed in order to extract the heart rate and possible anomalies. Those extracts constituting the low-bandwidth component of the sensed data, are then transformed into 12-byte packets and transmitted to the data-processing stations at 3-second intervals. The complete samples are stored locally in NVRAM (in a circular buffer). A processing station may decide to poll a sensor node for a selected fragment of the stored data. The primary responsibility of the system is to deliver the 3-second packets on time, while offering background service for reliable (albeit less time-critical) delivery of NVRAM extracts upon demand.

The role of the underlying ad-hoc network is to cover a relatively large area, e.g., a recreation facility, with inconspicuous and inexpensive nodes, with the practical transmission range of 50-150 m. The nominal transmission rate of the wireless channel is ca. 10 Kb/s. A data packet may have to travel several hops before being delivered.

Our prototype network consists of 48 nodes, programmed in PicOS [17], which we spread in various manners as to study different node densities and network geometries. With the average node connectivity of 4, the average hop length of 40 meters and m=1, the packet delivery fraction for the critical samples is over 95%, even though the protocol does not implement acknowledgments and retransmission for this type of data. Even with occasional background transmissions of NVRAM extracts, there is enough bandwidth in the network to sustain the application under moderate mobility conditions, whereby at any time 25% of all nodes are moving at a speed between 0.5 and 4 m/s. A thorough performance study is currently under way.

REFERENCES

[1] Afridi, M. et al. MEMS-based embedded sensor virtual components for system-on-a-chip (SoC). Solid-State Electronics, vol. 48, pp. 1777-1781, 2004.

[2] Jovanov, E. et al. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. Journal of Neuro-Engineering and Rehabilitation, 2:6, doi: 10.1186/1743-0003-2-6, 2005.

[3] Steibig, H. et al. Vertically integrated thin-film color sensor arrays for advanced sensing applications. Appl. Phys. Lett. 88, 013509 (2006).

[4] Daly, D.C. Chandrakasan, A.P. Energy efficient OOK transceiver for wireless sensor networks. In Proc. Radio Frequency Integrated Circuits (RFIC) Symposium, IEEE, 2006.

[5] Xue, G., Du, D. Z., and Cao, F. Recent advances in wireless ad hoc networks. Wireless Communications and Mobile Computing, 6(2): 147-149, 2006.

[6] Virone, G. et al. An assisted living oriented information system based on a residential wireless sensor network. In Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference, Arlington, Virginia, April 2-4, 2006, pp. 95-100.

[7] Buttyan, L. and Hubaux, J. P. Enforcing service availability in mobile ad hoc WANs. In Proceedings of the First IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Boston, MA, USA, 2000.

[8] Kaminska, B. and New, W., Wearable biomonitors with wireless network communication", IEEE Engineering in Medicine and Biology, EMBS 2005.

[9] Kaminska, B. Wireless micro and nano sensors for physiological and environmental monitoring, NSTI Nanotech, Boston, May 2006.

[10] Inaudi, D. and Glisic, B. Reliability and field testing of distributed strain and temperature sensors. In Proceedings of SPIE Smart Structures and Materials Conference, San Diego, February 2006.

[11] Perkins, C. E. and Bhagwat, P. Highly dynamic Destination-Sequenced Distance Vector routing (DSDV) for mobile computers. In Proceedings of SIGCOMM'94, pages 234–244, August 1993.

[12] Perkins, C. E. and Royer, E. M. Ad hoc On-demand Distance Vector Routing (AODV). In Proceedings of the IEEE workshop on Mobile Computing Systems and Applications (WMCSA), pp. 90-100, 1999.

[13] Johnson, D. B. and Maltz, D. A. Dynamic Source Routing in ad hoc wireless networks. In Imielinski and Korth, editors, Mobile Computing, volume 353. Kluwer Academic Publishers, 1996.

[14] Hu, Y. C., Perrig, A., and Johnson, D. B. Ariadne: A secure on-demand routing protocol for ad-hoc networks. In Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom), September 2002.

[15] IEEE Standards Department. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1997. IEEE 802.11-1997.

[16] Gburzynski, P., Kaminska, B. W., "A tiny and efficient wireless ad-hoc protocol for low-cost sensor networks", DATE 2007.

[17] Akhmetshina, E., Gburzynski, P., and F. Vizeacoumar. PicOS: a tiny operating system for extremely small embedded platforms. In Proceedings of ESA'03, pp. 116-122, 2003.