World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:1, 2008

A Review of Coverage and Routing for Wireless Sensor Networks

Hamid Barati, Ali Movaghar, Ali Barati, and Arash Azizi Mazreah

***Abstract*—**The special constraints of sensor networks impose a number of technical challenges for employing them. In this review, we study the issues and existing protocols in three areas: coverage and routing. We present two types of coverage problems: to determine the minimum number of sensor nodes that need to perform active sensing in order to monitor a certain area; and to decide the quality of service that can be provided by a given sensor network. While most routing protocols in sensor networks are data-centric, there are other types of routing protocols as well, such as hierarchical, location-based, and QoS-aware. We describe and compare several protocols in each group. We present several multi-path routing protocols and single-path with local repair routing protocols, which are proposed for recovering from sensor node crashes. We also discuss some transport layer schemes for reliable data transmission in lossy wireless channels.

***Keywords*—**Sensor networks, Coverage, Routing, Robustness.

I. Introduction

A sensor network is composed of a large number of tiny autonomous devices, called sensor nodes. Each sensor node has four basic components: a sensing unit, a processing unit, a radio unit, and a power unit. All these units fit into a matchbox-sized (or even smaller) module [1]. Since a sensor node has limited sensing and computational capabilities and can communicate only within short distances, a sensor network is the corporative effort of hundreds or thousands sensor nodes. These nodes are deployed densely and coordinate amongst themselves to achieve a common task.

Sensor networks have been proposed for a wide variety of application areas, including industrial, military, biomedical, and environmental areas. Some examples of sensor network applications are as follows:

1. Intrusion detection and tracking. Sensors are deployed along the border of a battlefield to detect, classify, and track intruding personnel and vehicles [5].

2. Environmental monitoring. Specialized sensor nodes that are able to detect temperature changes and/or smoke can be deployed in high-risk areas of a forest, to give out early warning of forest fires.

3. Indoor surveillance. Surveillance sensor networks can be used to provide security in an art gallery, shopping mall, or other facilities.

4. Traffic analysis. Traffic sensor networks can monitor vehicle traffic on a highway or a congested part of a city.

These sensor networks applications differ significantly. However, the tasks performed by the sensors are similar: sensing the environment, processing the information, and sending information to the base station(s).

II. Coverage

Coverage is one of the fundamental problems in sensor networks. The most common problem is the node scheduling problem: determine the minimum number of sensor nodes that need to perform active sensing in order to provide certain coverage (with desired reliability).

On the other hand, given a sensor network, we need to find out the quality of service that can be provided by this network, e.g., finding breach areas or best observed areas.

A. Node Scheduling Protocols

A sensor network is composed of a large number of sensors deployed in high density. Each individual sensor is unreliable and short-lived. One goal is to extend the system lifetime, as well as maintain desired coverage and reliability. We can achieve this goal by keeping enough working nodes to assure system functionality and turning off redundant nodes. In this section, we study and compare two protocols, PEAS [39] and a coverage-preserving node scheduling scheme [33] proposed at the University of Ottawa. Propose PEAS [39], a probing-based density control protocol that extends network lifetime by keeping only a necessary set of sensors in working mode and putting the remaining ones into sleep state. A sleeping node wakes up once in a while to probe its neighborhood. When a working node receives a probing message, it sends a reply. If a probing node does not get a reply within a certain amount of time, the node assumes that there is no working node within its probing range and starts to work. Otherwise, the node goes back to sleep. Fig. 1 shows the state transition at each node.

H. Barati is with the Islamic Azad University Dezful Branch, Dezful, Iran (e-mail: hbarati@iaud.ac.ir).

A. Movaghar is with the Sharif University of Technology, Tehran, Iran, (e-mail: movaghar@sharif.edu).

A. Barati is with the Islamic Azad university, Dezful Branch, Dezful, Iran, (e-mail: abarati@iaud.ac.ir).

A. Azizi Mazreah is with the Islamic Azad University, Sirjan Branch, Sirjan, Iran (e-mail: aazizi@iausirjan.ac.ir).

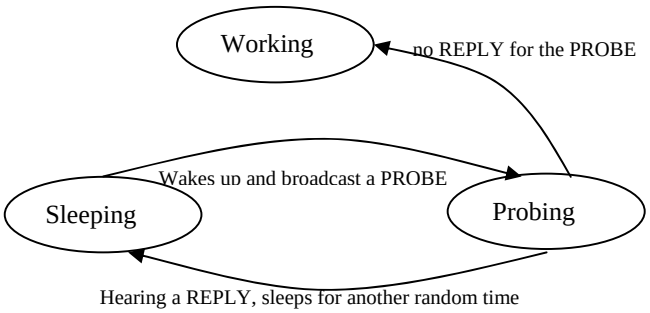

Fig. 1 PEAS: state transition at each node (adapted from [39])

Two parameters decide the performance of the algorithm: the probing range and the wakeup rate.

The desired redundancy can be achieved by setting the corresponding probing range. The number of wakeups decides the overhead, thus it should be kept low. However, if the wakeup rate is set to be too low, when a working node fails unexpectedly, there can be large "gaps" during which no working node is available. PEAS allow the application to decide the desired frequency of wakeups. Each probing node adjusts its wakeup rate according to the observation of its sleeping neighbors, so that transient node failures are tolerated by the application.

In [33], propose a coverage-preserving node scheduling scheme, which guarantees that the sensing coverage is maintained after turning off redundant nodes. In their approach, each node decides on whether to turn itself on or off based on the off-duty eligibility rule: a node is eligible to turn off itself if its neighbors can cover its sensing area. The node scheduling operation is divided into rounds. Each round includes a self-scheduling phase, followed by a sensing phase. In the self-scheduling phase, nodes exchange position information among neighbors. Each node computes its neighbors' sponsoring sensing area, and decides whether it is eligible for off-duty.

Blind points may appear if all the nodes make decisions simultaneously, as shown Fig. 2. Node 1 finds that its sensing area can be covered by node 2, 3, and 4, thus it turns itself off according to off-duty eligibility rule. At the same time, node 4

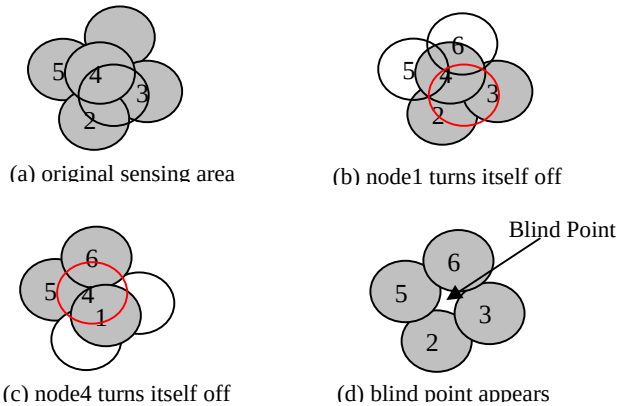

Fig. 2 Blind Point Occurrence

also finds that its sensing area can be covered by node 1, 5 and 6. Node 4 turns itself off too. Since both node 1 and node 4 are turned off, a blind point occurs.

A back-off scheme is used to solve this problem. Each node waits for a random back-off time period before determining its eligibility and waits for a short time, if it decides to turn off. A node announces its decision to turn off by broadcasting a Status Advertisement Message (SAM). When receiving a SAM, a node will delete the sender's information from its neighbor lists. Thus, the nodes that have decided to be turned off will not be considered by their neighboring nodes.

Compared to the probing-based approach, PEAS, the coverage-preserving node scheduling scheme in [33] focuses more on static scenarios, that is, calculating the sensing area that neighboring nodes can help to monitor. PEAS assume that all nodes have the same sensing range, while the node scheduling scheme in [33] allows nodes to have different sensing ranges. The advantage of PEAS is that the algorithm is simple, no location information is needed, and no per-neighbor state needs to be kept. However, PEAS does not ensure sensing coverage, and blind points may appear after turning off some nodes. On the other hand, the approach in [33] ensures sensing coverage, while it requires exchange of location information among neighboring nodes. Comparison of these two approaches is shown in Table I.

TABLE I
COMPARISON OF PEAS AND COVERAGE-PRESERVING NODE SCHEDULING SCHEME

PEAS	Coverage-Preserving Node Scheduling
focuses on dynamic scenario	focuses on static scenario
requires all nodes have the same sensing range	nodes can have different sensing ranges
no location information is needed	requires exchange of location information among neighboring nodes
simple probing approach, no per-neighbor state is kept	nodes according to their location information
does not guarantee coverage, blind points may occur	ensures sensing coverage

B. Quality of Service

Sensing coverage decides the monitoring quality provided by a given sensor network. In [27], define two types of coverage: worst case coverage and best case coverage.

Deciding worst case coverage of a particular sensor network is to detect breach regions in the network, that is, the areas of low observability from sensor nodes. On the other hand, deciding best case coverage of a sensor network is to identify the best monitored regions. The authors propose an algorithm for calculating the maximal breach paths (worst case coverage) and maximal support paths (best case coverage) in a sensor network. The maximal breach path is a

path where its closest distance to any of the sensors is maximized. The maximal support path is a path where its farthest distance from the closest sensors is minimized. The key idea is to use Voronoi diagram and Delaunay triangulation, combined with graph search algorithms for finding the optimal paths in each case. The Voronoi diagram is composed of line segments that are equidistant from neighboring sensors, while the Delaunay triangulation is formed by connecting nodes that share a common edge in the Voronoi diagram. The maximal breach path must lie on the lines of Voronoi diagram, since any point deviating from the Voronoi line segments must be closer to at least one sensor. Similarly, the maximal support path lies on the lines of the Delaunay triangulation of the sensors.

III. Routing Protocols

Routing techniques are needed to send data between sensor nodes and the base station. Several routing protocols are proposed for sensor networks. These protocols can be divided into the following categories: data-centric protocols, hierarchical protocols, location based protocols, and some QoS-aware protocols.

A. Data-Centric Protocols

The first category of routing protocols we consider at this review are data-centric protocols, including SPIN (Sensor Protocols for Information via Negotiation) [11], directed diffusion [14], GRAdient Broadcast (GRAB) [38, 40], and Rumor routing [2]. Data-centric routing protocols can be further divided into event-driven, query-driven, or a combination of both, depending on whether sources or destinations initiate data flow. SPIN is the first data-centric routing protocol. It includes a family of protocols used to efficiently disseminate information in a wireless sensor network. SPIN-1 is a source-initiated protocol. It applies a 3-stage (ADV-REQ-DATA) handshake interface for disseminating data. SPIN nodes assign high-level names to their data, called meta-data. They use meta-data to negotiate with each other before transmitting data. This avoids transmitting redundant data in the network.

We show an example of SPIN-1 in Fig. 3. Node A sends an ADV message to its neighbors, saying that it has new data to disseminate. When node B receives the ADV packet, it checks to see whether it has all the advertised data (a). If not, node B sends a REQ message back to node A (b). When node A receives the REQ packet, it responds by sending a DATA message, containing the requested data (c). After receiving the data from node A, node B sends ADV messages advertising the new data to all its neighbors except node A (d). These nodes send requests to node B, and the protocol continues.

Directed Diffusion is a destination-initiated protocol. Its features include attribute-based naming, data-centric routing, and in-network aggregation. Each sensor node names its data with one or more attributes. A destination node sends interests requesting for data, based on these attributes.

Interests are flooded over the network. When a node receives an interest from a neighbor, it sets up a gradient to send data to the neighbor.

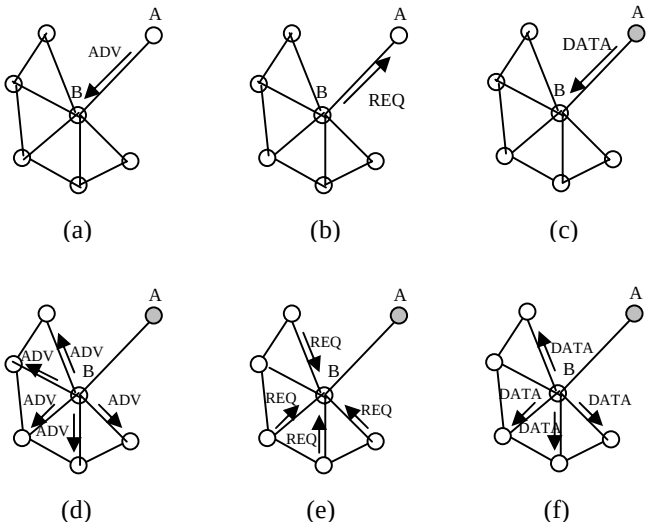

Fig. 3 The SPIN-1 Protocol (adapted from [11])

Each node only knows the neighbor from whom it got the interest. It is possible that each node would receive the same interest from more than one neighbor. In this way, multiple paths can be set up from the source node to the destination node. Among these paths, one or a few high rate paths are defined and other paths remain low rate. Depending on the number of paths that are reinforced, directed diffusion can be single-path or multi-path routing. If a better path emerges, the protocol will change high rate paths to low rate ones using negative reinforcements. Further, directed diffusion allows intermediate nodes to cache and aggregate data to improve data accessibility and energy efficiency.

Both SPIN and directed diffusion address the data-centric characteristics of sensor networks. SPIN is a source-initiated (or event-driven) protocol. Information dissemination starts as source nodes advertising the availability of new information. It reduces network traffic and energy consumption through meta-data negotiation. The limitation of SPIN is that it does not guarantee data delivery.

GRAdient Broadcast (GRAB) [38, 40] is built on directed diffusion. However, GRAB nodes record the cost, when the interest is diffused through the network. The goal is to deliver messages along the minimum-cost path in a large network. Each node has a cost field indicating the minimum cost from this node to the sink, which can be defined as the height of the node. A message carries the minimum cost from the source to the sink, and the cost that it has consumed so far from the source to the current node. A source node broadcasts the message. A neighboring node forwards this message only if the sum of the consumed cost and the cost at this node equals to the source's cost. GRAB uses a delay-based algorithm to compute the minimum cost. A node defers broadcasting interest for some time, and chooses the message which leads to the minimum cost. Thus, the delay used by a node is

critical. They prove that the delayed time should be proportional to the node's minimum cost.

GRAB improves energy efficiency by forwarding data only along the minimum-cost path. Rumor routing [2] is proposed to fill the region between query flooding and event flooding. It is useful only when the number of queries compared to the number of events falls into a certain range. And it is designed to be adjustable to support different query to event ratios.

The goal of rumor routing is to avoid expensive flooding operations. Unlike Directed Diffusion and GRAB, which try to find an optimal path by flooding queries for gradient setup, rumor routing sends a query on a random walk until it finds the event path. In rumor routing, each node maintains a neighbor list and an event table. When a node witnesses an event, it adds the event to its event table. Nodes that have recently observed an event generate an agent with a certain probability.

An agent is a long-lived packet, roaming through the network and propagating information. Each agent carries a list of events it has encountered, along with the number of hops to that event. When it arrives at a node, it synchronizes its list with the node's list. The agent travels the network for some number of hops, and dies. Any node can generate a query that is destined to a particular event. If it has a route to the event, then it will transmit the query. If it does not, it will forward the query to a random neighbor. If the query has not reached the destination node when the query Time-to-Live (TTL) expires, then it will retransmit the query, or flood the query.

B. Hierarchical Protocols

The next category of routing protocols is hierarchical protocols. LEACH (Low-Energy Adaptive Clustering Hierarchy) [10] is one of the first hierarchical routing protocols in sensor networks. The basic idea of LEACH is to organize sensor nodes into clusters based on the received signal strength and use local cluster-heads as routers to base station. LEACH performs local data fusion and aggregation at cluster-heads to further reduce energy consumption. Sensor nodes elect themselves to be local cluster heads with a certain probability.

Each non-cluster-head node determines which cluster it wants to join by choosing the cluster-head that requires the minimum communication energy. Once all nodes are organized into clusters, each cluster-head creates a schedule for the sensors in its cluster. The cluster-head position is randomly rotated among the various sensors in order to not drain the battery of a single sensor.

Threshold-sensitive Energy Efficient sensor Network protocol (TEEN) [26] and Adaptive Periodic TEEN (APTEEN) are the follow-up work of LEACH. TEEN is designed to be responsive to sudden changes in the sensed attributes. TEEN uses two thresholds (hard threshold and soft threshold) to reduce message transmission. At every cluster change time, the newly elected cluster-head broadcasts hard threshold and soft threshold to its members. Sensor nodes sense their environment continuously, and report data to the cluster-head only when the value of sensed attribute is equal to or greater than the hard threshold, or the change in the value of sensed attribute is equal to or greater than the soft threshold. Since these thresholds are broadcast afresh at every cluster change time, the user can change them as required. The main drawback of TEEN is that if the thresholds are not reached, the nodes will never communicate.

APTEEN allows the user to set threshold values and a count time interval. The count time interval is the maximum time period between two successive reports sent by a node; that is, if a node does not send data for a time period equal to the count time, it is forced to sense and retransmit the data.

PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [23] is a chain-based protocol that is an improvement over LEACH. In PEGASIS, each node communicates only with a close neighbor. When a node on the chain receives data from a neighbor, it aggregates the data with its own data and send the data to the next neighbor on the chain. Rather than multiple cluster-heads sending data to the base station as LEACH, only one node on the chain is selected to transmit to the base station.

LEACH and its follow-up work (TEEN, APTEEN and PEGASIS) are all built on the assumption that the base station is fixed and located far away from the sensors. And they also assume that every sensor can reach the base station directly. These assumptions severely limit the applicability of these protocols.

C. Location-based Protocols

Some of the routing protocols in sensor networks require location information for sensor nodes. Location information can be derived from GPS signals, received radio signal strength, etc. By using location information, a relatively optimal path can be found without flooding. In this section, we look at three location-based routing protocols in sensor networks: Greedy Perimeter Stateless Routing (GPSR) [16] and Geographical Energy Aware Routing (GEAR) [41], and Two-Tier Data Dissemination (TTDD) [25].

In GPSR, a sensor node makes packet forwarding decisions based on the positions of its immediate neighbors. Each node uses greedy forwarding to forward packets to its neighbor that is closer to the destination. When a packet reaches a region where a greedy path does not exist, GPSR recovers by routing around the perimeter of the region. GPSR scales well since sensor nodes only keep state about the local topology. It can find correct new routes quickly under topology changes.

GEAR forwards packets to all the nodes in the target region in two steps: (1) forwarding the packets towards the target region; (2) disseminating the packet within the region. In the first step, a node forwards packets to the nearest neighbor to the target region. If there is no neighbor closer than the node, GEAR picks a node that minimizes some cost value of this neighbor. If the packet has reached the target region, it can be diffused in that region by recursive geographic forwarding.

The target region is divided into four sub-regions and four copies of the packet are created. This recursive splitting and

forwarding procedure continues until there is one node inside this sub-region.

TTDD is a two-tier data dissemination model that is designed to deal with the sink mobility problem. In previous work, mobile sinks have to broadcast their location information continuously, so that all sensor nodes get updated with the new position of sinks. This continuous broadcasting incurs increased collisions in wireless transmissions and excessive energy consumption. TTDD solves the multiple mobile sinks problem by proactively building a grid structure. The sensors closest to grid points are called dissemination nodes. Instead of propagating query messages to all the sensors, only the dissemination nodes need to acquire the forwarding information. A query traverses two tiers to reach a source. The lower tier is within the local grid square of the sink's current location, and the higher tier is made of the dissemination nodes at grid points.

The sink floods its query within a cell. The query is then received by the nearest dissemination node, which forwards it on to its upstream dissemination node toward the source, which in turn further forwards the query. This process continues until the query reaches either the source or a dissemination node that is receiving data from the source. In this way, TTDD localizes the impact of sink mobility on data forwarding. Only a small set of sensor nodes need to maintain forwarding state.

D. QoS-aware Protocols

QoS-aware protocols consider end to end delay requirements while setting up the paths in the sensor network. We discuss sample of these protocols in this section.

SAR: Sequential Assignment Routing (SAR) is the first protocol for sensor networks that includes the notion of QoS in its routing decisions. It is a table-driven multi-path approach striving to achieve energy efficiency and fault tolerance. The SAR protocol creates trees rooted at one-hop neighbors of the sink by taking QoS metric, energy resource on each path and priority level of each packet into consideration. By using created trees, multiple paths from sink to sensors are formed. One of these paths is selected according to the energy resources and QoS on the path. Failure recovery is done by enforcing routing table consistency between upstream and downstream nodes on each path. Any local failure causes an automatic path restoration procedure locally. Simulation results show that SAR offers less power consumption than the minimum-energy metric algorithm, which focuses only the energy consumption of each packet without considering its priority. SAR maintains multiple paths from nodes to sink.

Although, this ensures fault-tolerance and easy recovery, the protocol suffers from the overhead of maintaining the tables and states at each sensor node especially when the number of nodes is huge.

Energy-Aware QoS Routing Protocol: A fairly new QoS aware protocol for sensor networks is proposed by Akkaya and Younis. Realtime traffic is generated by imaging sensors. The proposed protocol extends the routing approach in [15] and finds a least cost and energy efficient path that meets certain end-to end delay during the connection. The link cost used is a function that captures the nodes' energy reserve, transmission energy, error rate and other communication parameters. In order to support both best effort and real time traffic at the same time, a class-based queuing model is employed. The queuing model allows service sharing for real-time and non-real-time traffic. The bandwidth ratio r, is defined as an initial value set by the gateway and represents the amount of bandwidth to be dedicated both to the real-time and non-real-time traffic on a particular outgoing link in case of a congestion. As a consequence, the throughput for normal data does not diminish by properly adjusting such "r" value. The queuing model is depicted in Fig. 4, which is redrawn from [22]. The protocol finds a list of least cost paths by using an extended version of Dijkstra's algorithm and picks a path from that list which meets the end-to-end delay requirement. Simulation results show that the proposed protocol consistently performs well with respect to QoS and energy metrics. However, the r-value is set initially same for all the nodes, which does not provide flexible adjusting of bandwidth sharing for different links. The protocol is extended in [49] by assigning a different r-value for each node in order to achieve a better utilization of the links.

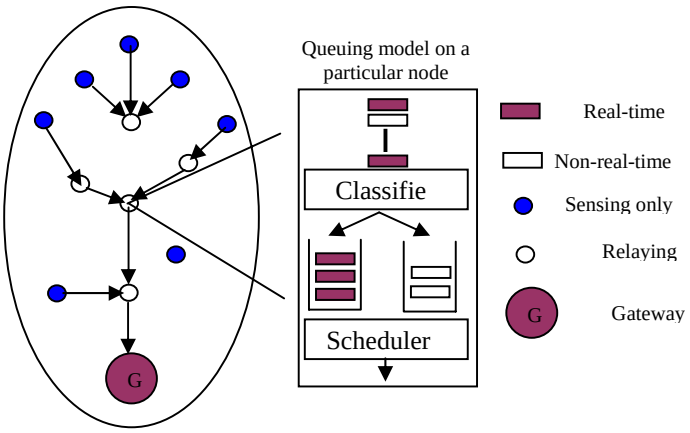

Fig. 4 Queuing model in a particular sensor node

SPEED: A QoS routing protocol for sensor networks that provides soft real-time end-to-end guarantees. The protocol requires each node to maintain information about its neighbors and uses geographic forwarding to find the paths. In addition, SPEED strive to ensure a certain speed for each packet in the network so that each application can estimate the end-to-end delay for the packets by dividing the distance to the sink by the speed of the packet before making the admission decision. Moreover, SPEED can provide congestion avoidance when the network is congested.

The routing module in SPEED is called Stateless Geographic Non-Deterministic forwarding (SNFG) and works with four other modules at the network layer, as shown in Fig. 5. The beacon exchange mechanism collects information about the nodes and their location. Delay estimation at each

node is basically made by calculating the elapsed time when an ACK is received from a neighbor as a response to a transmitted data packet. By looking at the delay values, SNGF selects the node, which meets the speed requirement. If such a node cannot be found, the relay ratio of the node is checked. The Neighborhood Feedback Loop module is responsible for providing the relay ratio which is calculated by looking at the miss ratios of the neighbors of a node (the nodes which could not provide the desired speed) and is fed to the SNGF module. If the relay ratio is less than a randomly generated number between 0 and 1, the packet is dropped. And finally, the backpressure-rerouting module is used to prevent voids, when a node fails to find a next hop node, and to eliminate congestion by sending messages back to the source nodes so that they will pursue new routes.

When compared to Dynamic Source Routing (DSR) and Ad-hoc on-demand vector routing (AODV), SPEED performs better in terms of end-to-end delay and miss ratio. Moreover, the total transmission energy is less due to the simplicity of the routing algorithm, i.e. control packet overhead is less, and to the even traffic distribution. Such load balancing is achieved through the SNGF mechanism of dispersing packets into a large relay area. As explained earlier, similar energy saving technique is used in GBR by spreading traffic uniformly through the network. SPEED does not consider any further energy metric in its routing protocol. Therefore, for more realistic understanding of SPEED's energy consumption, there is a need for comparing it to a routing protocol, which is energy-aware.

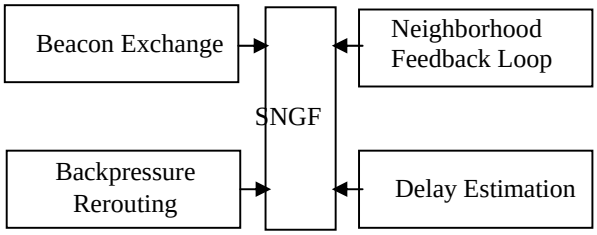

Fig. 5 Routing Component of SPEED

IV. Conclusion

In this survey, we first gave an overview of wireless sensor networks, their application areas, characteristics and distinct requirements. We then covered two topics in sensor networks: coverage and routing. The first problem in coverage is the node scheduling problem. We presented and compared two node scheduling protocols: PEAS, a probing-based density control protocol [39], and the coverage-preserving node scheduling protocol [33], which focuses on calculating the sponsoring sensing area by neighboring nodes. We then discussed the question of the quality of service that can be provided by a particular sensor network, which is quantified by calculating maximal breach path and maximal support path. Next, we classified the routing protocols for sensor networks into four categories: data-centric protocols, hierarchical protocols, location-based protocols, and QoS-aware protocols. Due to the data-centric nature of sensor networks, most routing protocols belong to the first category. Among these protocols, some are source-initiated, some are destination-initiated, some focus on energy efficiency, and some emphasize robustness. Hierarchical protocols are a special family of routing protocols.

The idea is to organize sensor nodes into clusters and use cluster-heads as routers to aggregate and transmit data to base station. There are groups of routing protocols that require location information. By using location information, they can find a relatively optimal path without flooding. We also presented several QoS-aware protocols, which address energy efficiency and real-time requirements.

References

[1] Limin Wang," Survey on Sensor Networks", Department of Computer Science and Engineering Michigan State University.

[2] Kemal Akkaya and Mohamed Younis, "A Survey on Routing Protocols for Wireless Sensor Networks".Department of Computer Science and Electrical Engineering University of Maryland, Baltimore County.

[3] Crossbow Technology, Inc. Mote In-Network Programming User Reference Version 20030315, 2003.

[4] Stefan Dulman, Tim Nieberg, Paul Havinga, and Pieter Hartel. Multipath routing for data dissemination in energy e±cient sensor networks. Technical Report TR-CTTT-02-20, Center of Telematics and Information Technology, University of Twente, The Netherlands, July 2002.

[5] A. Arora et. al. A Line in the sand: a wireless sensor network for target detection, classification, and tracking. Technical Report OSU-CISRC-12/03-TR71, The Ohio State University, December 2003.

[6] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACMTransactions on Networking, 5(6):784{803, December 1997.

[7] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. Mobile Computing and Communications Review, 1(2), October 2002.

[8] Tian He, John A. Stankovic, Chenyang Lu, and Tarek F. Abdelzaher. SPEED: a real-time routing protocol for sensor networks. Technical Report CS-2002-09, University of Virginia, March 2002.

[9] Wendi Beth Heinzelman. Application-Specific Protocol Architectures for Wireless Networks. PhD thesis, Massachusetts Institute of Technology, June 2000.

[10] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd HawaiiInternational Conference on System Sciences (HICSS), Maui, Hawaii, USA, January 2000.

[11] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 174{185, Seattle, Washington, USA, August 1999.

[12] J. Hill and D. E. Culler. Mica: A wirleess platform for deeply embedded networks. IEEE, 2002.

[13] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In Proceedings of The Ninth International Conference on Architectural Support for Programming Language and Operating Systems(ASPLOS-IX), pages 93{104, November 2000.

[14] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed Diffusion: a scalable and robust communication paradigm for sensor networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, pages 56{67, Boston, Massachusetts, USA, August 2000.

[15] Chris Karlof, Yangping Li, and Joseph Polastre. ARRIVE: algorithm for robust routing in volatile environments. Technical Report UCB CSD-03-1233, University of California, Berlekey, March 2003.

[16] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In Proceedings of International Conference on Mobile Computing and Networking (Mobicom), Boston, Massachusetts, USA, August 2000.

[17] Bhaskar Krishnamachari, Yasser Mourtada, and Stephen Wicher. The energy-robustness tradeoff for routing in wireless sensor networks. Technical Report Autonomous Networks Research Group (ANRG) Technical Report TR02-001, University of Southern California, September 2002.

[18] Sandeep S. Kulkarni and Karun N. Biyani. Correctness of component-based adaptation.Proceedings of the 7th International Symposium on Component-Based Software Engineering(CBSE), 3054, 2004.

[19] Sandeep S. Kulkarni, Karun N. Biyani, and Umamaheswaran Arumugam. Composing distributed fault-tolerance components. Proceedings of the International Conference on Depend-able systems and Networks (DSN), Supplemental Volume, Workshop on Principles of Dependable Systems, pages W127{W136, June 2003.

[20] Sandeep S. Kulkarni and Limin Wang. Mnp: Multihop network reprogramming service for sensor networks. Technical Report MSU-CSE-04-19, Michigan State University, May 2004.

[21] Sung-Ju Lee, Elizabeth M. Belding-Royer, and Charles E. Perkins. Scalability study of the ad hoc on-demand distance vector routing protocol. International Journal of Network Management, 13:97{114, March/April 2003.

[22] Philip Levis, Neil Patel, Scott Shenker, and David Culler. Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. Technical report, University of California at Berkeley, 2003.

[23] Stephanie Lindsey and Cauligi S. Raghavendra. PEGASIS: power-e±cient gathering in sensor information systems. In Proceedings of IEEE Aerospace Conference, March 2002.

[24] Ching-Gung Liu, Deborah Estrin, Scott Shenker, and Lixia Zhang. Timer adjustment in SRM. Technical Report USC 97-656, Univ. Southern California, July 1997.

[25] Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zhang. TTDD: a two-tier data dissemination model for large-scale wireless sensor networks. In Proceedings of International Conference on Mobile Computing and Networking (MobiCom), Atlanta, Georgia, USA, September 2002.

[26] A. Manjeshwar and D. P. Agrawal. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, in conjuntion with 2001 IPDPS, Hyatt Regency, San Francisco, USA, April 2001.

[27] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mina B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In Proceedings of INFOCOM, pages 1380{1387, Anchorage, Alaska, USA, April 2001.

[28] Christos Papadopoulos, Guru Parulkar, and George Varghese. An error control scheme for large-scale multicast applications. In Proceedings of INFOCOM, San Francisco, USA, March-April 1998.

[29] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing.In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pages 90{100, New Orleans, Louisiana, USA, February 1999.

[30] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, USA, March 2002.

[31] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J. Pottie. Protocols for self-organization of a wireless sensor network. IEEE Personal Communications, pages 16{27,October 2000.

[32] Fred Stann and John Heidemann. RMST: reliable data transport in sensor networks. In Proceedings of The First International Workshop on Sensor Net Protocols and Applications(SNPA'03), Anchorage, AK, USA, May 2003.

[33] Di Tian and Nicolas D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta, Georgia, USA, September 2002.

[34] Di Tian and Nicolas D. Georganas. Low-cost, reliable data delivery in large wireless sensor networks. Technical report, University of Ottawa, 2003.

[35] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. PSFQ: a reliable trans-port protocol for wireless sensor networks. In Wireless Sensor Networks and Applications (WSNA), Atlanta, Georgia, USA, September 2002.

[36] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity con¯guration in wireless sensor networks. In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SENSYS),Los Angeles, CA, USA, November.