

Communication in WSN

Vishal Arora*, Sunny Behal**, Charanjit Singh***

*BCET, Gurdaspur,** SBSCET, Ferozpur,,*** RIMT-Polytechnic

Abstract- A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. The power of wireless sensor networks lies in the ability to deploy large numbers of tiny nodes that assemble and configure themselves. Usage scenarios for these devices range from real-time tracking, to monitoring of environmental conditions, to ubiquitous computing environments. In addition to drastically reducing the installation costs, wireless sensor networks have the ability to dynamically adapt to changing environments. Adaptation mechanisms can respond to changes in network topologies or can cause the network to shift between drastically different modes of operation. Unlike traditional wireless devices, wireless sensor nodes do not need to communicate directly with the nearest high-power control tower or base station, but only with their local peers. Instead, of relying on a pre-deployed infrastructure, each individual sensor or actuator becomes part of the overall infrastructure. The flexible mesh architectures envisioned dynamically adapt to support introduction of new nodes or expand to cover a larger geographic region. Additionally, the system can automatically adapt to compensate for node failures. The vision of mesh networking is based on strength in numbers. Unlike cell phone systems that deny service when too many phones are active in a small area, the interconnection of a wireless sensor network only grows stronger as nodes are added. As long as there is sufficient density, a single network of nodes can grow to cover limitless area. With each node having a communication range of 50 meters and costing less than \$1 a sensor network that encircled the equator of the earth will cost less than \$1M.

I. COMMUNICATION IN WIRELESS SENSOR NETWORKS

Data communication in sensor networks can happen in the following ways:

A. Single packet sensors-to-sink data delivery: In this communication scenario the sensors deliver critical packets for example some highly aggregated data or an important item tag information in inventory tracking applications [14].

B. Block of packets delivery: In applications where the sink disseminates new code or new queries into the network for retasking [1], large blocks of packets will need to be transported from the sink to the sensors.

• Stream of packets: Applications in which the sensor nodes periodically report data to the sink is the primary example of stream of packets communication.

II. SINGLE PACKET DELIVERY

Single packet delivery is important when the sensor nodes send a critical piece of information to the sink node. The critical piece of information could be an aggregation of data collected over a period of time or detection of an important event. Reliable single packet delivery can be achieved with the following approaches –

A. Single path delivery with MAC – layer retransmissions

B. In order to provide reliability with MAC – layer retransmissions the following issues needs to be resolved –

- Who detects losses and what are the indicators used?
- Who requests retransmissions?
- Who actually carries out these retransmissions?

In single packet delivery using MAC – layer retransmissions the transmitting node has to detect losses by using timers and has to carry out the retransmissions. The receiver will notify successful reception by sending acknowledgments. Block or stream delivery has more flexibility as it is possible to let the receiver detect losses for example, by checking for holes in the sequence number space and request retransmission of missing packets by using negative acknowledgment (NACK) packets. If additionally NACKs are understood as carrying implicit acknowledgments then there is no necessity to send positive acknowledgments for every packet, thus saving lots of energy. MAC - layer retransmissions: When a node in the routing path forwards the data packet, it expects to receive a MAC-layer acknowledgment. For setting timers single-hop propagation delays and packet processing times have to be considered. Typically, the transmitter makes a bounded number of trials to successfully forward the packet and drops it after this number has been exhausted. A CSMA (Carrier Sense Multiple Access) based adaptive rate control scheme for media access in sensor networks: A.Woo et. al in [4] propose a Media Access control scheme for sensor networks by studying the unique application behavior and tight constraints in computation power, storage, energy resources and radio technology. The protocol design is as follows –

Carrier Sense Multiple Access (CSMA) and collision detection schemes found in Ethernet are examples of listening mechanisms. Listening is effective when all nodes can hear each other, (i.e. without hidden nodes). Though listening is simple, it does come with an energy cost, because the radio must be on to listen. To conserve energy, it is important to shorten the length of carrier sensing. The highly synchronized nature of the traffic imposes new criteria for CSMA. Since there are no mechanisms for detecting collisions, nodes that

happen to send at the same time will corrupt each other. If the traffic pattern is independent, this situation is not likely to repeat. However, detecting of common events by nodes will synchronize data transfers from these nodes which can lead corruption of all data. The solution is to use random delay for transmission to unsynchronize the nodes.

- *Backoff Mechanism*

Random backoff is used to reduce contention. The idea of backoff is to restrain a node from accessing the channel for a period of time and hopefully the channel will be free after the backoff period.

- *Rate Control Mechanism*

The tension between originating traffic and route-thru traffic has a direct impact in achieving the fairness objective. The adaptive rate control idea is simple and is best explained with an analogy of metering traffic onto a freeway where the route-thru traffic is like traffic on the freeway and each node originating data is like cars trying to enter. Periodically a node attempts to inject a packet. If the packet is successfully injected it becomes part of the route-thru traffic. As it is routed by the node's parent, it signals that the road still has capacity for more traffic and thus, the node can increase its transmission rate. However if the injection of the packet wasn't successful, it signals that the road is jammed and the node decreases its rate of originating data and backoff to achieve a phase change effect.

III. THE HIDDEN NODE PROBLEM

Figure illustrates the hidden node problem. Suppose that node A wants to transmit to node B located at a distance x from A. By only sensing the medium, node A will not be able to hear the transmissions by any node (C) in the dashed area denoted by $A(x)$, and will start transmitting, leading to collisions at node B.

This is the well known hidden terminal problem, where the hidden nodes are located in the area $A(x)$.

A. S-MAC Protocol: W.Ye et. al in [5] propose S-MAC a medium access control protocol for wireless sensor networks which has the following features –

- A low-duty-cycle scheme for multi-hop networks that reduces energy consumption due to idle listening. Idle listening refers to the nodes listening to the wireless channel even when it is not expecting any messages. Every node maintains a schedule for sleep and listens cycles.

- A RTS/CTS (Request to send, Clear to send) mechanism for collision avoidance. This mechanism solves the hidden node problem by exchanging the control information packets – RTS and CTS before actually sending the data packet. This increases the reliability of packet delivery for large data message sizes (100 bytes – 200 bytes) when compared to the CSMA protocol [6]. However for smaller data sizes the exchange of control packets – RTS and CTS – is an overhead in terms of energy consumption and latency.

B. Single path delivery with end-to-end retransmissions

In this the source node needs to buffer the packet until an acknowledgement from the sink node arrives. The number of retransmissions that the node does is typically bounded. However setting timers is harder in this case as reasonable guesses would need knowledge on the number of hops, the per-hop delay and the effect of current cross-traffic. Another drawback in using end-to-end retransmissions is the overhead of retransmitting the packets along the entire routing path even when the packet loss has occurred close to the destination. The HHR approach (Hop-by-Hop Reliability) described in [7] relies on sending multiple copies of the same packet which are unicast by each node back to back to the next upstream node. The required number of copies is determined from a locally estimated packet error rate, the desired packet level probability and the hop-distance to the sink. Alternatively, packets are repeated until a local acknowledgment has been received (HHRA). These schemes are sub-optimal since multiple copies are simply wasted when copies are transmitted during good channel periods i.e., when the bit-error rate is low, when a single or a few packets would likely suffice.

C. Using multiple paths

One of the approaches in using multiple paths is to set them up in advance and declare one of them as the default route. Once problems occur, another route is used [8] or the route is repaired locally by a rerouting scheme [9]. However in either case extra route maintenance is required which does not necessarily pay off in single delivery applications.

Other approaches send not only a single packet over one of the paths but transmit multiple packets over multiple paths in parallel. In the ReInForM scheme [10], multiple copies of the same packet are transmitted over randomly chosen routes. Specifically, it is assumed that a packet is destined to a sink node and that each node knows its hop distance to the sink as well as the hop distances of all its immediate neighbors. Packet duplication can occur at every intermediate node, not only at the source node. An intermediate node has to decide two things: the number of copies to create and the upstream nodes to which the packet is actually forwarded. With respect to the latter choice, ReInForM prefers nodes which are closer to the sink but otherwise the choice is random. This distributes the load over many nodes and avoids quick depletion of nodes along a "good" route. The number of duplicates is determined from the locally estimated error rate, the hop distance to the sink and the target delivery probability.

Packet Block Delivery

Block transfers are needed when large amounts of data (e.g., code updates) have to be transported. One important feature of such block transfers is that NACKs (Negative acknowledgments) can be used. This potentially reduces the number of acknowledgment packets. A NACK is regarded as a retransmission request issued by the receiver. When an intermediate node caches the segments, it can serve such a request as well as the original source node could but with the

benefit that the NACK and the following retransmitted segment do not need to travel the whole distance between source and sink node. Such a node is also called a recovery server [11] [12]. In an extreme case, all nodes in the network could spend buffer for caching. Let us discuss some of the schemes incorporating the above ideas.

A. PSFQ: Pump Slowly, Fetch Quickly

The PSFQ protocol presented in [1] is a transport level protocol designed to deliver a number of segments from a single source node to a subset of receiver nodes or even to all nodes within a sensor network. It provides guaranteed delivery, eg., for code updates.

The protocol consists of three basic primitives: a pump operation, a fetch operation and a report operation. In the pump operation, the sink node transmits all the segments making up the block one by one, using MAC-layer broadcasts. The time T_{min} between the different segments is comparably large and hence the pumping operation is considered slow pumping. Each segment is equipped with a sequence number. All other nodes behave as follows:

- When a node A receives a new segment not yet seen, it stores it in an internal cache. When the segment has already been received before, the new segment is simply dropped.
- When a new segment is received in-sequence, node A waits for some random time and forwards it further. However, forwarding is suppressed when A finds that four or more of its neighbors have already forwarded the same segment, since the expected additional coverage achieved by A forwarding the segment tends to be small.
- When a packet is received out-of sequence, it is also stored, but instead of forwarding it, the node requests immediate retransmission of the missing segments from any upstream neighbor using a NACK message indicating the missing segments. This is the process of quickly recovering from the error in the packet sequence. As soon as the node receives the missing segments, it starts forwarding the segments in-sequence in the pumping mode, i.e. with long delays in between.

The decision to forward packets only in-sequence has the advantage that loss events do not propagate: Suppose that node A pumps packets $x_1, x_2, x_3, \dots, x_n$. Node A's downstream neighbor B has received and forwarded packet x_i and receives the packet x_{i+2} afterwards. Node B triggers a fetch operation. If B were to forward x_{i+2} further to some downstream (w.r.t B) node C, C would also trigger a fetch operation which is useless and a waste of energy. The fetch operation corresponds to a NACK or a retransmission request and is triggered by missing sequence number. When segments from the end of the block are missing, there is no higher sequence number and this method of detecting packet losses fails. To attack this problem, a node A proactively triggers the fetch operation by sending the NACK packet. If the upstream neighbors do not possess the packets, they forward the NACK

packet further upstream until it eventually reaches the node having the missing segments.

The NACK packets themselves are broadcasted and any upstream neighbor having some of the missing segments is invited to respond. To avoid collisions among these packets, the nodes use random delays before sending the answer.

The report operation is requested from the sink node. The most far-away nodes from the sink issue report packets indicating their own address and the received/missing segments. This way the sink can judge the progress of the code block dissemination.

B. GARUDA

The scheme developed in the GARUDA project [11] addresses a similar problem as PSFQ, namely the reliable transfer of block data from the sink to all sensors or a significant part of the network. GARUDA uses a NACK based scheme and additionally takes great care that the first packet in the block is reliably delivered to all sensors. This solves the problem of NACK based schemes that a receiver needs to receive at least one packet from the block to detect losses of further packets at all.

GARUDA constructs an approximation to the minimum dominating set of the sensor network topology and the members of this set (called core members) act as recovery servers for downstream core members and neighboring non-core members. Only those nodes are candidates for the core that have a hop distance to the sink which is an integral multiple of three. A candidate core member refrains from becoming a core member when there are enough core members in its neighborhood. On the other hand, non-core members having no core member in their range can request a candidate core member to really become a core member. All core members know at least one upstream (i.e. closer to the sink) core member from which they request retransmissions. The reliable delivery of a block of data from the sink proceeds in two steps:

First within the core, and then the non-core nodes fetch missing data from their associated core members. GARUDA is based on out-of-order delivery. A core member x requests missing segments from its upstream core member y , but does this only when x knows that the missing segment is indeed available at y . To achieve this, node y includes into every forwarded packet a bitmap indicating the segments that y already has, and x can use this knowledge to suppress NACKs for packets missing at y . A non-core member a associated to x suppresses all retransmission requests until x has all the segments present, indicated by a full bit-map.

C. RMST: Reliable Multi-Segment Transport

The RMST scheme [3] adds reliable data transfer to directed diffusion [13]. RMST is designed for delivering large blocks of data in multiple segments from a source node to a sink node. This is required for applications where in time series data has to be transmitted. RMST combines several mechanisms to enforce reliability:

- In RMST's cached mode the sink node and all intermediate nodes on an enforced path cache segments and check the cache periodically for missing segments. When a node detects missing segments, it generates a NACK message which travels back to the source along the reinforced path. The first node A having missing segments in its cache forwards them again towards the sink (and thus towards the requesting node). If A can retrieve all requested segments from its cache, then A drops the NACK packet, otherwise it is forwarded further upstream. Both the segments and the NACK packets are represented in terms of attributes, to be compatible with directed diffusion. In the noncached mode of RMST only the sink node has such a cache but not the intermediate nodes; therefore, NACK's travel back to the source node (which also needs to cache the segments).
- Use of application layer redundancy: the source sends out the whole data block periodically until the sink explicitly unsubscribes.
- By frequently repeating interest propagation, dissemination of exploratory events and subsequent establishment of (new) reinforced routes some resilience against node failures is achieved. The authors investigate different combinations of the above mechanisms for their total number of bytes (data plus overhead) needed to transmit 50 segments of 100 bytes size, it showed up that MAC-layer retransmissions are helpful in case of higher packet loss rates. But interestingly, using the cached mode without MAC-layer retransmissions (and thus without MAC layer overhead like acknowledgments or RTS/CTS handshakes) is the cheapest approach (given that all intermediate nodes cache segments).

D. Packet Stream Delivery

Some sensor network applications require the sensor nodes to generate and report their data periodically. An important reliability target in such a setup is to ensure that the sink receives a sufficient number of packets per unit time to achieve desired information accuracy. Since many environmental processes vary only slowly, repeated sensor readings of the same or neighborhood sensors are often correlated and accordingly some lost packets are acceptable. Therefore, the key mechanism to ensure delivery of the desired number of packets at the sink node are not retransmissions, but instead to control either the packet generation rate of the sensor nodes or alternatively the number of nodes generating packets at a fixed rate.

E. ESRT: Event to Sink Reliable Transport [2]

The ESRT protocol works by adjusting the reporting sensors packet generation rate such that sufficient number of packets arrives at the sink without producing congestion. It is assumed that the sink requires this minimum number of packets to achieve desired information quality. The situation considered by the algorithm is that of a single sink node to which all the sensor nodes direct their readings. The sink node is not energy constrained and can transmit with sufficient power to reach all the sensors. It uses this ability to control the rate f_n by which

sensors generate data packets in the n -th round of the algorithm. The control strategy is based on a certain relationship between the generation rate f_n on the one hand and the observed sink quality (given as the rate of delivered packets per unit time) and congestion state on the other hand. Following are some of the scenarios –

- For very low generation rates there is no congestion and the quality is insufficient.
- When the data generation rate is increased, the desired quality is reached within some fraction ϵ and without causing congestion. So the network is not congested and the sink receives just the right number of packets to achieve the desired quality, not much more or less. This is the target region.
- When the data generation rate is increased, more packets than needed are delivered without causing congestion.
- A further increase in data generation rate results in a decrease in the number of delivered packets as congestion starts to build up and the packets are dropped.

The sink node collects congestion signs and observes the rate of incoming packets for a certain time, called a round. Based on this information it determines the current scenario and computes a new desired generation rate for the next round – which can be smaller, larger or equal to the current generation rate – and broadcasts this to all sensor nodes. The control strategy strives to reach the target region. The congestion state is detected by sensors from their local buffer occupancy, taking the current occupancy and the growth trend of buffer occupancy with respect to previous rounds into account. Upon congestion detection the sensor node sets a congestion notification bit in outgoing packets. The sink infers a congestion state when any incoming packet has this bit set. Under the assumption that in the non-congested scenario there is a linear relationship between the reporting rate and the number of packets received at the sink per unit time, it can be shown that the protocol always converges to the target region, with the convergence speed depending on it. The protocol does not require the sink or the sensor nodes to have global knowledge like for example the current number of available sensor nodes. One of the drawbacks of the ESRT scheme is that all sensor nodes are controlled at once, treating interesting regions or regions with higher node density in the same way as uninteresting regions or regions with low node density.

F. Wireless communication requirements

To communicate over a wireless link, protocols must be built up from the raw electro-magnetic signaling primitives. A transmitter must carefully modulate the RF carrier while receiver performs demodulation and signal analysis. Figure 1-1 illustrates the key phases of a packet-based wireless communication protocol. It is important to note that many of the operations must be performed in parallel with each other. This can be seen in the distinct layers that overlap in time. The first step in the communication process is to encode the data for transmission. The coding schemes are designed to increase the probability of a successful transmission by preventing and correcting slight errors. For efficiency reasons the encoding

process is pipelined with the actual transmission process. Once the first byte is encoded, transmission may begin. The remaining bytes can be encoded as preceding bytes are transmitted. Coding schemes can range from simple DC-balancing schemes, such as 4b-6b or Manchester encoding, to complex CDMA schemes. In either, a collection of one or more data bits, called a *symbol*, are coded into a collection of radio transmission bits called *chips*. Manchester encoding has two chips per symbol which represents 1 bit of data. Direct sequence spread spectrum and CDMA schemes often have 15 to 50 chips per symbol with each symbol containing 1 to 4 data bits.

The actual transmission begins with the execution of a media access control protocol (MAC). MAC protocols are designed to allow multiple transmitters to share a single communication channel. One of the simplest MAC protocols is carrier sense media access (CSMA) where each transmitter first checks for an idle channel prior to each transmission. If the channel is busy, it waits for a short, random, delay after which it reinitiates the transmission. The first piece of data to be actually transmitted over the radio link is a synchronization symbol or start symbol. The start symbol signals to the receiver that a packet is coming and is used by the receiver to determine the timing of the arriving transmission. The start symbol is immediately followed by the encoded data transmitted as a serial stream. As the transmission proceeds, the transmitter must precisely control the timing of each bit transition so that the receiver can maintain synchronization. Skewed bit transitions can cause the sender and receiver to get out of synch, resulting in an unsuccessful transmission or corrupted data. For a receiver, the first part of data reception is to detect that a transmission has begun. The channel is monitored continually in order to filter out background noise and detect the special start symbol. The length and format of the start symbol can be optimized for the expected noise levels. In order to properly detect the start symbol, the receiver must sample the channel at least twice the radio chip rate. Otherwise, the relative phase of the sampling and the transmission may result in the receiver missing the start symbol. Once detected, the receiver must then synchronize itself to the exact phase of the incoming transmission. This synchronization step is critical in allowing the receiver to determine the start and end of each bit window being used by the transmitter. Synchronization requires the incoming transmission to be sampled higher than twice the bit rate so the timing of the bit transitions can be determined. Once synchronized, the receiver then samples the value of the incoming signal at the center of each bit. Precise care must be taken to minimize skew in the sampling rate and timing. As the individual bits are extracted from the radio, they are assembled into blocks that are the encoded version of actual data messages. Finally, the blocks are decoded back into the original data and assembled into a packet. The decoding process can often correct bit errors in the received signal and reproduce the original data.

REFERENCES

- [1] C. Wan, A. Campbell, L. Krishnamurthy. PSFQ: A Reliable Transport Mechanism for Wireless Sensor Networks. ACM International Workshop On Wireless Sensor Networks and Applications, Atlanta, Georgia. Sept 2002.
- [2] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," presented at the ACM MobiHoc, Annapolis, MD, Jun. 2003.
- [3] R. Stann and J. Heidemann, "RMST: Reliable data transport in sensor networks," in *Proc. 1st IEEE Int. Workshop Sensor Net Protocols Appl. (SNPA)*, Anchorage, AK, May 2003, pp. 102–112.
- [4] A. Woo and D. Culler, "A transmission control scheme for media access in sensor networks," in *Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 221–235.
- [5] Wei Ye, John Heidemann and Deborah Estrin, "Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3, June 2004.
- [6] Wei Ye, John Heidemann and Deborah Estrin, "A Flexible and Reliable Radio Communication Stack on Motes," USC/ISI Technical Report ISI-TR-565.
- [7] B. Deb, S. Bhatnagar, and B. Nath, "Information assurance in sensor networks," in *Proc. 2nd ACM Intl. Workshop on Wireless Sensor Networks and Applications (WSNA)*, San Diego, CA, Sept. 2003.
- [8] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks," *Mobile Computing and Communications Review (MC2R)*, vol. 1, no. 2, 2002.
- [9] D. Tian and N. D. Georganas, "Energy efficient routing with guaranteed delivery in wireless sensor networks," in *Proc. IEEE Wireless Communications and Networking Conference 2003 (WCNC'03)*, Institute of Electrical and Electronics Engineers. New Orleans, USA: IEEE Press, Mar. 2003.
- [10] B. Deb, S. Bhatnagar, and B. Nath, "Reinform: Reliable information forwarding using multiple paths in sensor networks," in *Proc. 28th Annual IEEE Conference on Local Computer Networks (LCN 2003)*, 20-24 Oct. 2003 Page(s):406 - 415.