# Reliability in Wireless Sensor Networks: Survey and Challenges Ahead

Muhammad Adeel Mahmood and Winston Seah

*School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand*

**Abstract**

This paper presents a survey on existing data transport reliability protocols in wireless sensor networks (WSNs). Most of the existing research employs retransmission mechanisms to achieve reliability, ignoring the use of redundancy schemes such as erasure codes to enhance *event* reliability. We review the data transport reliability schemes in terms of *event* and *packet* reliability using retransmission or redundancy mechanisms. Furthermore, we analyse and compare the existing data transport reliability protocols based on several attributes, specially the importance of *event* reliability over *packet* reliability through the use of retransmissions or redundancy. On the basis of the analysis, we finally point out some future research directions and challenges ahead in order to enhance reliability in WSNs.

*Keywords:* Reliability, redundancy, retransmission, erasure codes.

## 1. Introduction

Wireless Sensor Networks (WSNs) have been the preferred choice for the design and deployment of next generation monitoring and control systems [33, 34]. It has now become feasible to deploy massive amounts of low-cost sensors to monitor large regions over ground surface, underwater, or atmosphere [35]. The most significant benefit of sensor networks is that they extend the computation capability to physical environment where human beings cannot reach [36]. They can operate for prolonged periods in habitats that are hostile, challenging or ecologically too sensitive for human visitation

[37]. Moreover, it has the potential to provide wealth of data about the environment in which they are deployed and send their results across the network to the end-users [38, 39].

At the heart of WSNs are the sensor nodes. A sensor node is a tiny device that is capable of sensing, computation and communication capabilities. Depending on their sensing components, sensor nodes can be used to monitor different phenomena like temperature, light, and humidity. The processing module, of the sensor node, is able to do computation on the sensed value and also on the other received values from their neighbors. The communication module in sensor nodes is used to send and receive the information from the neighboring nodes [40]. Since a single sensor provides only limited information; a network of these sensors is used to manage large environments.

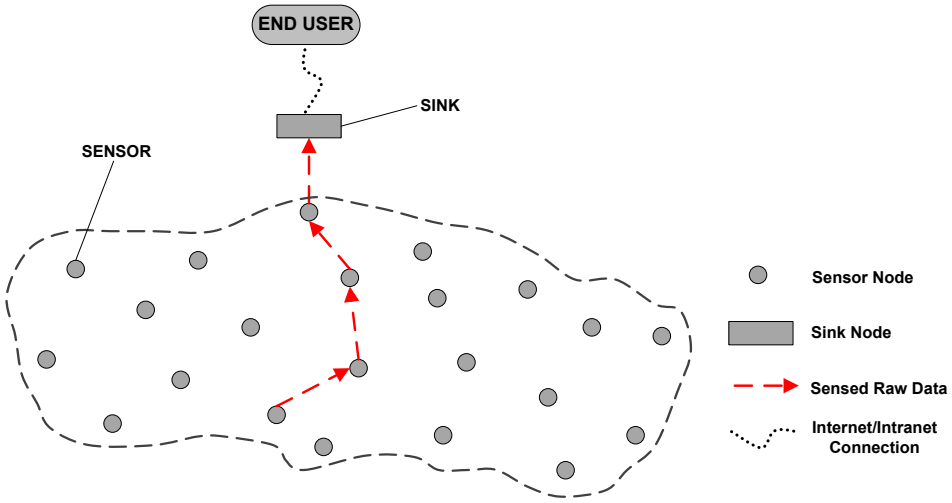


Figure 1: Wireless Sensor Network

Figure 1 shows a Wireless Sensor Network (WSN) comprising of several sensor nodes and a sink node. The sensor nodes continually sense data from the environment and send them to the sink node. In most settings, tens to thousands of such sensor nodes are distributed throughout a region of interest, where they self-organize into a network through wireless communication and collaborate with each other to accomplish a common task [36]. The sink receives all the information from these sensor nodes, processes it and sends them to the end user.

In WSNs critical event data collected by the sensor nodes need to be reliably delivered to the sink for successful monitoring of an environment. Therefore, given the nature of error prone wireless links, ensuring reliable transfer of data from resource constrained sensor nodes to the sink is one of the major challenges in WSNs. Reliable transfer of data is the surety that the packet carrying event's information arrives at the destination. In WSNs, reliability can be classified into different levels i.e.

- *Packet* or *Event* reliability level
- *Hop-by-Hop* or *End-to-End* reliability level

*Packet* or *event* reliability is concerned with how much of information is required to notify the sink of an occurrence of something happening in the environment. *Packet* reliability requires all the packets carrying sensed data from all the sensor nodes to be reliably transported to the sink. Whereas, *event* reliability ensures that the sink only gets enough information about a certain event happening in the network instead of sending all the sensed packets.

In addition to *packet* or *event* reliability; the successful recovery of certain event information can be reliably achieved either at *hop-by-hop* or *end-to-end* level. In *hop-by-hop*, the next hop is responsible for ensuring the reliable transmission of information to the destination i.e. the sink node. Whereas, in *end-to-end* reliability, only the end points (i.e. only the source and destination nodes) are responsible for ensuring the successful transmission of information.

To achieve *packet* or *event* reliability in terms of recovering the lost information at the *hop-by-hop* or *end-to-end* level is through the use of retransmissions or redundancy mechanism. Retransmission is simply the retransmissions of the lost information. Retransmissions can either be performed on *end-to-end* or *hop-by-hop* basis. *End-to-end* retransmission requires only the source node that generated the packet to retransmit the lost information. *Hop-by-hop* retransmission allows the intermediate nodes to perform retransmission of lost information by caching it in their local buffer. In redundancy mechanism redundant information is added to the original data that allows the receiver to recover from information loss. Like retransmissions, redundancy-based reliability can also be performed on *end-to-end* or

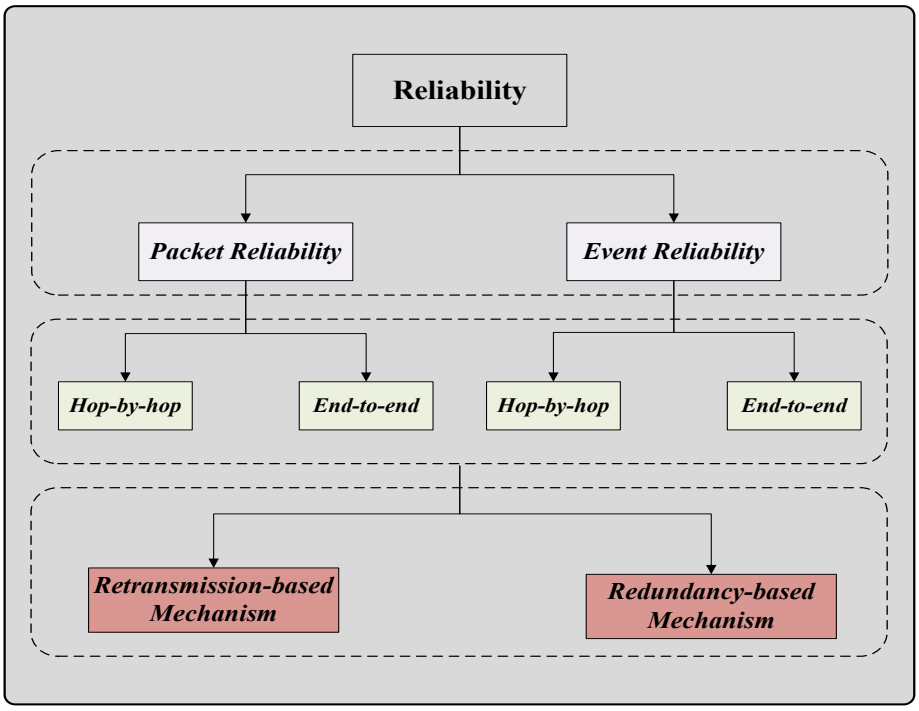


Figure 2: Classifications of Reliability in WSNs

*hop-by-hop* basis. In *end-to-end* redundancy, encoding/decoding is performed only at source and the sink node. Whereas in *hop-by-hop* redundancy, encoding/decoding is performed at each intermediate hop in order to reconstruct the lost bits at each hop.

In this paper, we aim to present a survey on reliable data transport protocols based on retransmissions and redundancy mechanisms. Section 2 gives a background of the key terms used in this paper. Section 3 presents some of the existing data transport protocols based on retransmissions mechanisms. Section 4 addresses some of the existing protocols that aim to achieve reliability through the use of information redundancy. Finally, we enumerate several problems that can be further studied in future and concludes the paper in section 5.

## 2. Background

A number of protocols have been proposed to address the reliability issue in wireless sensor networks, each of them proposes different ways of reliably transporting data packets from sensor node to the sink. Due to the convergent nature of traffic in WSNs, all sensor nodes in the network tends to inject their captured data towards the sink. With the increase of traffic flow the networks starts getting congested, whereas, the area around the sink becomes more congested as all the traffic is flowing towards the sink. This becomes the cause of packet loss, because there is not enough buffering space in sensor nodes. Besides losing packets through congestion, packets are also lost due to transmission error, packet collision, interference, node failure (due to energy depletion) or any other unforeseeable reasons. Furthermore, due to the short range of sensor nodes, data might have to travel a large number of hops which in turn introduces a lot of entry points for errors that can also become the cause of packet loss. Thus, to ensure reliability, the lost packets need to be recovered which requires either a retransmission or redundancy mechanism.

One way of achieving reliability in terms of recovering the lost packets is through the use of retransmissions of the lost packets. Retransmissions can either be performed on *end-to-end* or *hop-by-hop* basis. *End-to-end* retransmission requires only the source node that generated the packet to retransmit the lost packet. Whereas, *hop-by-hop* retransmission allows the intermediate nodes to perform retransmission of lost packets by caching the information of data packets in their local buffer. In a multi-hop WSNs, retransmission-based reliability can be achieved on the basis of acknowledgement mechanism, where the sender node requires an acknowledgement of its data packets from the next hop relay nodes in the path to the sink. The major kinds of acknowledgement mechanisms are explicit acknowledgement (eACK), negative acknowledgement (NACK) and implicit acknowledgement (iACK). The eACK mechanism is a traditional way of ensuring the absolute reliability guarantee for every single packet transmitted. The eACK rely on a special control message which the receiving node after successfully receiving a packet, sends back to the sender as a receipt of the sent packet. Like eACK, NACK also corresponds to a special control message that allows the receiver to notify the sender to retransmit a particular missing sequence of packets. The eACK and NACK mechanisms result in high transmission overhead and

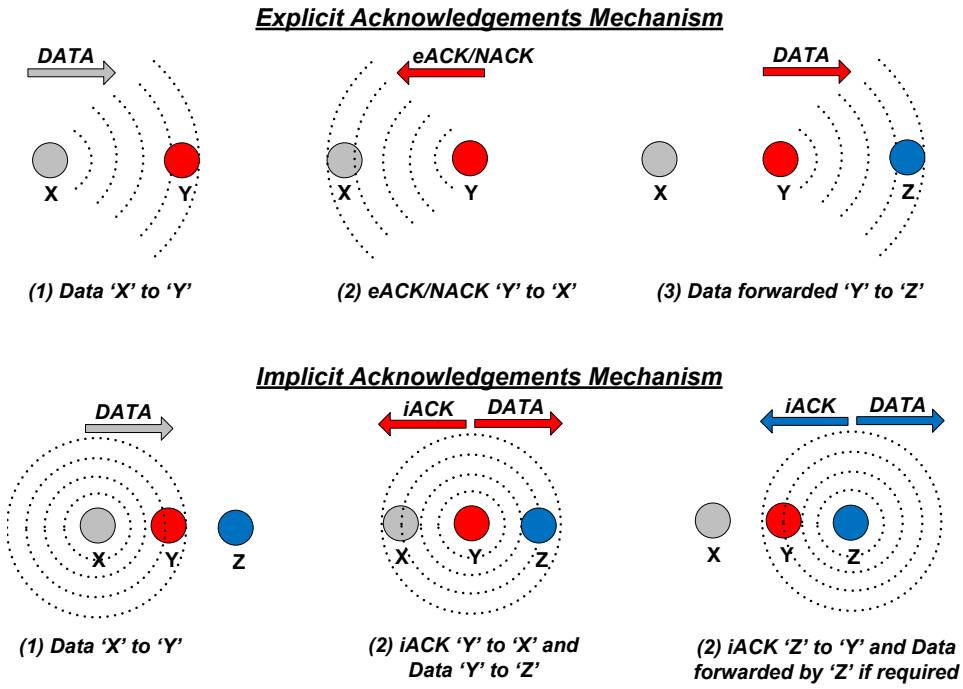


Figure 3: Implicit & Explicit Acknowledgement Mechanisms for Retransmissions [41]

energy wastage which may not be feasible for most of the WSN applications. On the other hand, in alternative iACK mechanism the sender after transmitting the packet, listens to the channel and interprets the forwarding of its sent packet by the next hop node as a receipt of acknowledgement. In this way, iACK exploits the broadcast nature of wireless channel avoiding the additional transmission overhead and energy wastage caused by the transmission of control messages (eACK/NACK).

Any of the acknowledgement schemes require the node to cache its transmitted packets for any possible retransmission request. Thus to prevent the stale packets occupying the buffer which might also become the cause of delay, the packets needs to be immediately retransmitted as the loss is detected. The immediate retransmission would further aggravate the network condition, in case the network is congested. This requires an intelligent retransmission mechanism to be introduced. However, given the limited memory of the sensor node, large number of packets may not be cached at the sensor node in terms of providing retransmission-based reliability. This gives the

idea of providing data transport reliability based on information redundancy, which solves the problems caused by data caching at the nodes.

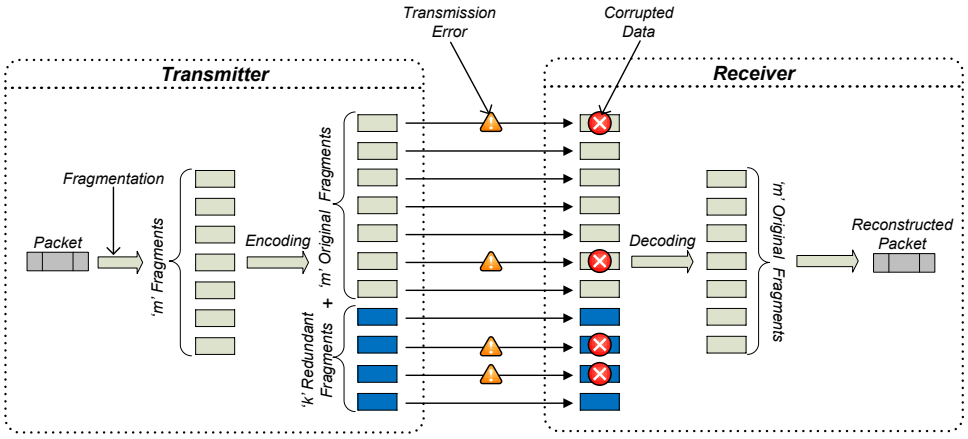


Figure 4: Redundancy Mechanism using Erasure Coding

Apart from retransmissions, another way to achieve data transport reliability is by introducing information redundancy where multiple copies of the same packet are transmitted based on erasures codes that allows receiver to recover from independent packet loss [2]. In erasure coding, the sender divides the packet into $m$ fragments, which are further encoded by adding $k$ redundant fragments. These $m+k$ fragments are transmitted towards the sink, where the sink after receiving the encoded set of fragments, reconstructs the original data out of it. The receiver is able to reconstruct the original data packet only on a condition that the number of fragments it has received are equal to or greater than the number of original fragments i.e. $m$. Like retransmissions, erasure coding can also be performed on *end-to-end* or *hop-by-hop* basis. In *end-to-end* erasure coding the encoding/decoding of data is only performed on the source and the sink nodes, whereas, the intermediate nodes just relays the packets. However, in *hop-by-hop* erasure coding, encoding/decoding of data is performed at every intermediate node to reduce redundant data. Thus *end-to-end* erasure coding increases the overall network traffic by producing high number of redundant fragments as compared to *hop-by-hop* erasure coding.

### *2.1. Summary*

This section has presented an overview of data transport reliability in WSNs along with the challenges to efficiently achieve reliability. After introducing reliability, two ways of achieving data transport reliability namely, retransmissions and redundancy are described that are exposed in this research. This section also describes different levels of reliability and their combinations in order to achieve data transport reliability in WSNs based on either retransmissions or redundancy. Different terminologies that are used throughout this paper are also briefly explained.

## 3. Retransmissions-based Reliability

Reliable transmission of data is one of the major characteristic of most of the existing transport protocols in traditional networks or WSNs. In traditional networks, TCP (Transport Control Protocol) is responsible for the reliable delivery of data by providing *end-to-end packet* reliability [3]. In TCP, connection establishment between sender and receiver, *end-to-end* acknowledgements and retransmissions, and longer transmission time are few of the reasons that TCP would not be a feasible choice for the resource constrained WSNs. Despite these problems, attempts have been made to make TCP more suitable for WSNs. Some of the applications in sensor networks require guaranteed delivery, such as reporting the critical data from sensor nodes to the sink (upstream reliability) and distribution of new software codes or queries from the sink node to downstream sensor nodes (downstream reliability). Most of the WSN data transport reliability protocols focusses on providing upstream reliability, while some application requires downstream reliability guarantee [31], [32], [16] where the sink needs to perform some management and control tasks. Common techniques to ensure reliability includes the use of explicit, implicit or negative acknowledgements through the use of caching mechanisms while others regulate the flow of data through the use of event reporting frequencies; in addition, some of them aims to achieve reliability and event detection accuracy by increasing the node density to detect an event. A brief overview and shortcomings of the major data transport reliability protocols in terms of providing *event* or *packet* reliability are discussed as follows.

### *3.1. Event-to-Sink Reliable Transport (ESRT)*

Akan *et al.* [17] were the first one to introduce the concept of *event* reliability by presenting the most notable *event-to-sink* reliability protocol (ESRT). It guarantees *end-to-end* upstream delivery of individual event information to the sink, not individual packets from each sensor node. It measures reliability by the number of packets carrying information about a particular event that are delivered to the sink. The sink after receiving a number of packets within a time interval, calculates that how reliably a particular event has been detected based on the amount packets it receives for that particular event. For this, ESRT uses a centralized rate control mechanism running at the sink, where the sink achieves required reliability level by periodically controlling the event reporting frequency *(f)* of the nodes. When the required reliability is not achieved, the sink will increase the event reporting frequency *(f)* of the nodes. On the other hand, it will reduce *(f)* when there is congestion in order to bring back the reliability required at the sink. In the event that the required reliability is observed at the sink, it will decrease *(f)* to reduce the nodes' energy consumption. ESRT aims to achieve the optimal point where the event is reliably received at the sink without making the network congested, while adjusting *(f)* according to four different network conditions such as *(reliability, congestion), (reliability, no congestion), (no reliability, congestion) and (no reliability, no congestion).* A key assumption of ESRT is that the sink has a high power radio that can transmit the latest value of *(f)* to all sensor nodes in a single broadcast message, where each node adjusts its *(f)* accordingly.

ESRT is evaluated analytically and on the basis of simulations, where one of the serious limitations is that it requires all the nodes in the network to be one hop away from the sink. This is a serious limitation because conventionally the sink is not one hop away from all the sensor nodes. ESRT increases the source sending rate in order to guarantee *event* reliability; however, this central rate control method is not as energy efficient as *hop-by-hop* retransmission-based loss recovery schemes. This would further deteriorate the overall bandwidth utilization and introduce high energy wastage as different parts of the network have different network conditions. Furthermore, ESRT assumes the event area to be congested, if any of the intermediate nodes between source and the sink got congested. This assumption is not fair, as a clear event area is considered as congested based on congestion at intermediate nodes. Accurate event detection based on in-network data

processing would be more reliable and energy efficient as compares to a centralized system.

### *3.2. Delay Sensitive Transport (DST)*

Gungor and Akan [18] proposes DST, which is simply an extended version of ESRT [17], where application specific delay bounds are added, which was not considered in ESRT. As the main focus of DST is to achieve desired event detection reliability at the sink, restricted by the application-specific delay bound; therefore, it introduces a Time Critical Event First (TCEF) scheduling policy. According to TCEF, the data packets within each node's queue with the minimum remaining time to deadline are given high priority for the transmission. The remaining time to deadline is calculated by the piggybacking the elapsed time information of that event with the data packets, which eliminates the need for global time synchronization.

The predefined event locations (x, y) are randomly distributed, where all the nodes surrounding event's location, lying within the radius of the event are supposed to be the sources for that event. The event is considered to be reliably detected, if the observed reliability (i.e. number of packets received at sink in a decision interval within a delay bound) is higher than the desired reliability (i.e. minimum number of packets required for reliable event detection within a delay bound). Like ESRT, DST configures the reporting rates under same four network states as described in ESRT and aims to achieve the optimal state (i.e. adequate reliability, no congestion) with desired *event* reliability with delay consideration and no congestion.

One of the major criticisms in DST is that the whole functionality is totally dependent on the sink to regulate the flow of data, which results in increase in traffic and more energy usage. One question that needs to be asked is that what will happen in case of multiple events in close proximity, happening in parallel, as the protocol has just considered a single event and the reporting rates of the nodes within the range of that event. The consideration of multiple random events without prior knowledge would rather be a more practical approach.

### *3.3. Sensor Transmission Control Protocol (STCP)*

Iyer *et al.* [19] proposed Sensor Transmission Control Protocol (STCP) which is a generic *(sensor-to-sink) end-to-end* transport protocol providing both congestion control and reliability. STCP is also dependent on sink's central processing, where the sink is notified of congestion in the network by intermediate nodes using a congestion notification bit in packet header which they set based on the queue lengths. The sink then notifies the affected source nodes of the congested paths and to find the alternative paths for their packets to be reliably transmitted. STCP provides different degree of reliability for different kinds of applications, such that it handles different kind of data flows for event-driven and continuous data flow applications in terms of providing reliability. For event-driven applications, source nodes use ACK based *end-to-end* retransmissions to get a confirmation of packet reception from the sink; however, for continuous data flow applications, the sink uses NACK based *end-to-end* retransmissions.

Congestion is detected and notified to the sink by the intermediate nodes by looking at the queue length; whereas to maintain reliability, source node keeps the packets in its cache, until it gets an acknowledgement from the sink. In case of large hop count from source to the sink, the long waiting of the source node for an acknowledgement from the sink can become the cause of latency and cache overflow. STCP assumes that all the nodes in the network have their clocks synchronized with the sink, but the authors have not discussed that how this clock synchronization will be performed. Proper synchronization of nodes with the sink might play an important role in the performance of this protocol.

### *3.4. Price-Oriented Reliable Transport Protocol (PORT)*

The Price-Oriented Reliable Transport Protocol (PORT) reported by Zhou *et al.* [20] aims to provide finer fidelity in *event* reliability as compared to ESRT while minimizing energy consumption. Unlike ESRT [17], PORT performs the adjustment of frequency reporting rate in differentiated manner, as the different reporting rates of source nodes at different regions are adjusted accordingly. In addition to the adjustment of network-wide source-specific reporting rates, PORT provides an energy efficient congestion control mechanism by avoiding the nodes to travel through the paths having high loss rates based on the node price. Node price is actually the number

of transmission attempts made before the packet is being successfully delivered. The energy cost for the communication is also evaluated with the help of node price. To achieve this source specific need, PORT assumes that sink knows the location and the information contained in the packets.

PORT provides reliability in terms of sink getting enough information about the area of interest, instead of relying on the ratio of total number of incoming packet rate to the desired rate. To ensure reliability, PORT either uses a dynamic source report rate feedback and *end-to-end* communication cost mechanism. The first one allows the sink to adjust the reporting rate of every source on the basis of the node price piggybacked with the packets sent by each source. The sink then performs the adjustment of the source reporting rates based on the node price and information about the sensing area. The sink sends its feedback through the reverse path to the source. The other mechanism gives sink the overall *end-to-end* communications cost of transmitting information from source to the sink. The sink regulates the reporting rates of the particular source based on the communication cost information accordingly. Where, the sink decreases the reporting rates of sources having higher communication cost and vice versa, while maintaining reliability.

The simulation results show that PORT is more energy efficient as compared to ESRT. PORT works well for the networks having diversified reporting rates at different part of the network; however, if the reporting rates of different areas of the network are similar, then PORT will also give the similar results as the other existing schemes. Like the other *event* reliability schemes PORT also requires a high powered sink to regulate the flow of data. PORT provides *end-to-end* reliability guarantee, but it does not provide any packet loss recovery mechanism which may be vital in high loss environments.

### *3.5. Asymmetric and Reliable Transport (ART)*

Tezcan and Wang [21] introduced Asymmetric and Reliable Transport (ART) that chooses some essential nodes *(E-nodes)* within the sensor network that work together as cluster heads to provide upstream *end-to-end* reliability and downstream query reliability. Thus ART is one of the few protocols to provide bidirectional reliability. In addition, ART provides upstream congestion control mechanism in a decentralized way to regulate the data flow of intermediate nodes in an energy efficient way. Most of the work

is done by *E-nodes*, thus, all the nodes other than *E-nodes* do not face *end-to-end* communication overhead. The *E-nodes* are selected by looking at the remaining battery power of the nodes, which should be better than *non E-nodes*.

To achieve upstream reliability ART uses a light weight ACK mechanism between *E-node* and the sink. The *E-nodes* enable the event notification bit, start the timer and send the first event message to the sink and waits until an ACK is received. Whereas, the sink is enforced to send an ACK, if it receives a message with its event notification bit enabled. In case, *E-node* doesn't receive an ACK as the timer expires, it again enables the event notification bit, retransmits the message and reset the timer. On the other hand, ART performs reliable query propagation using NACK mechanism. After the sink sends query to *E-nodes*, it is the duty of the receiving *E-node* to check the sequence order and sends back a NACK to the sink, detecting the query fragment loss. Finally, Congestion is also controlled with the help of *E-nodes*. If the *E-nodes* don't receive any ACK from the sink as the time expires, it is assumed that the network is congested. The *E-nodes* then regulates the data flow by restricting its neighboring *non E-nodes* for sending any data until the congestion is cleared.

As the congestion control and *event* reliability is controlled by the only *E-nodes*; therefore, the packet loss due to congestion at non-essential nodes cannot be recovered by this protocol. ART assumes that there is congestion when ACK is not received within a timeout and thus the ACK is considered to be lost. This is not a reasonable assumption as it is not necessary that the ACK is lost due to congestion, it might be lost due to the link loss or some transmission errors. ART performs reliability and congestion control only for the essential node, whereas the reliability of major portion of the network that is non essential nodes are ignored.

### *3.6. Loss-Tolerant Reliable Event Sensing (LTRES)*

Recently, Xue *et al.* [22] identified a Loss-Tolerant Reliable Event Sensing transport protocol (LTRES) to satisfy the *end-to-end* transport reliability requirements for sensing applications with heterogeneous sensing fidelity requirements over different event areas within a wireless sensor network. LTRES addresses *event-to-sink* reliability by guaranteeing *end-to-end* transport reliability requirements of an event area instead of each sensor node. In

addition, it aims to provide sources rate adaption based on the network capacity. This network capacity awareness is calculates by measuring the event goodput at the sink. LTRES also employs a light weight congestion control mechanism based, where instead of relying at the nodes to report congestion; the sink itself suppresses the transmission of the aggressive nodes based on the data it receives.

LTRES maintains a certain level of loss tolerant reliable (LTR) data transport requirements on the basis of current network conditions and providing best-efforts service where that certain reliability level is not met. The sink after getting the events of interest from *E-nodes* covering the event area, measures the required event sensing fidelity on the basis of the observed sensing fidelity and pass it on to the set of *E-nodes*. Each *E-node* changes its source rate accordingly to deliver enough event goodput to the sink. Furthermore, a loss rate based congestion control mechanism is implemented at the sink, where the loss rate at each *E-node* is calculated by the sink in a periodic manner. The routing path is considered to be congested if the loss rate observed by the sink becomes too high. As a result, sink sends congestion notification to congested *E-node*, so that they start congestion avoidance mechanism by decreasing their source rates. When the sink assumes that an event area is of no more interest, then it broadcasts a session close packet to the *E-nodes* to make their source rates at default, and LTRES operation finishes.

This protocol is totally dependent on the sink, where the sink sends the updated source rates to the downstream *E-nodes*, but it doesn't employ any downstream reliability or congestion control mechanism. Lack of loss recovery mechanism and reliability considerations for the non-essential nodes would be some of the important issues that are ignored.

### *3.7. Quality-based Event Reliability Protocol (QERP)*

Hosung *et al.* [23] proposes Quality-based Event Reliability Protocol (QERP), which unlike the existing quantity-based *event* reliability protocols emphasize on the quality of event data reported to the sink. This recent study by Hosung *et al.* claims that the quantity-based *event* reliability protocols are not suitable for the resource constrained wireless sensor networks, as they aims to achieve reliability by increasing or decreasing the frequency of data reporting rates. These varying frequencies would sometimes increase the overall data rate more than the desired rate, which would aggravate the

network conditions. QERP as an *end-to-end event* reliability protocol claims to provide better reliability than ESRT and MERT [25] by using the concept of Contribution Degree (CD). This means that the difference of sensor data in CD for event detection is due to the different environmental conditions like varying distance of nodes from the events.

The data packets are marked with a CD value, where the nodes, located closer to the event's location are assigned with the high CD values and assumed to be critical data values. Similarly, the lower CD values show the farther nodes within the event's region that also detected the event. The CD and Full Selection (FS) fields are set in the packet header, where FS field is used when all the data packets within the event's region are required by the sink. The node closest to the event is selected as a leader node, while the rest of the nodes in the event's region report their data to the sink through the leader nodes. The sink compares the received reliability with the desired event reliability (DER) and may require the leader node to select and send the new data on the basis of the new DER value. The data with higher CD values are transmitted on priority by using CD-based buffer management and CD-based load balancing (high value CD on high link quality) mechanisms.

The simulation results show that QERP performs better than ESRT and MERT in term of energy efficiency and reliability. The flow of data in QERP is controlled by the sink, however, the buffer management and load balancing mechanisms are done at the node level. QERP makes the clusters by selecting the leader nodes within the event's region and makes the leader node work as a cluster head. If the node density is high near the event's location, then the process of selecting the leader node would be longer, resulting waste of energy. After the completion of transport process, if the revision of the DER is required by the sink, using ACKs, this would result in overhead. In case of packet loss, there is no packet loss detection mechanism discussed by the authors. QERP is an *end-to-end event* reliability protocol; having its data flow mechanism is based on the event's location. However, have not described or discussed any mechanism to find the event's location on the basis of which CD values are defined.

### *3.8. Event Reliability Protocol (ERP)*

Most recently Mahmood *et al.* [24] proposed ERP, an upstream *hop-by-hop event* reliability protocol aims to reliably deliver the information about

events happening to the sink. It suppresses the transmission of redundant packets containing information about the similar events based on spatio-temporal correlation. This reliable transmission of unique events to the sink is performed with the help an intelligent region-based selective retransmission mechanism. Apart from minimizing the transmission of unnecessary duplicate events, ERP further reduces the traffic by exploiting the broadcast nature of wireless channel through the use of implicit acknowledgement (iACK) mechanism. Instead of relying at the sink to control the data flow, ERP performs in-network data processing that saves the overall network cost in terms of energy, data flow and congestion.

The sensing regions of densely deployed sensor nodes often overlap with each other; therefore, the data sensed by a node might also be sensed by other neighboring nodes in close vicinity. A reliable transfer of events information from each sensing region would be sufficient for the sink. Therefore, the sink would not require receiving packets with the same event information from all the nodes belonging to same region. Thus ERP uses implicit acknowledgements with region-based selective retransmission mechanism. Where if a node does not receive an iAKC for its transmitted packet within a certain time frame, it will retransmit that packet based on a condition that another packet from the same region of the unacknowledged packet is not present in the node's queue. The retransmissions of the unique packets are performed until a certain threshold level is reached to prevent stale packets occupying the queue.

ERP gives better performance in terms of energy efficiency with low traffic overhead by minimizing the unnecessary transmission of duplicate events. The authors also look into the problem of declining network coverage with increasing node density which is still ignored by most of the existing data transport reliability protocols. The proposed approach highlights the importance of in-node data processing while minimizing the dependability on the sink. The authors have compared their protocol against SWIA which is simple and straightforward as already mentioned in the paper. Comparison with another efficient reliability protocol would further validate its performance. The use of eACKs might also be a suitable option, since eACKs do not force the sender to listen on the channel for possibly long time before it can overhear the packet transmission.

### *3.9. Stop-and-Wait-Implicit Acknowledgement (SWIA)*

Most of the work that we have discussed up till now focused on *event* reliability; however, upstream and downstream *packet* reliability also plays an important role. Upstream *packet* reliability requires all the packets carrying sensed data from all the sensor nodes to be reliably transported to the sink. While, downstream reliability requires 100% reliability guarantee, as if an application/code update for the sensor network is needed, then it must be installed in all the sensor nodes in the network in a broadcast fashion. Some of the important work on *packet* reliability is mentioned as follows.

Maróti [1] proposed a Directed Flood-Routing Framework (DFRF) using Stop-and-Wait Implicit Acknowledgement (SWIA) approach as a packet recovery mechanism in wireless sensor networks. In SWIA, the node will not begin sending another packet before it received and ACK for the previous packet. It makes use of the notion of implicit acknowledgement (iACK). This refers to the fact that a sending node, upon forwarding a packet, cab listen to the channel to see if the receiver has tried to further forward the packet. If this this iACK is heard by the sender within a certain time limit, the sender assumes that the receiver has received the packet, otherwise, the sender assumes that the packet is lost. The advantage of SWIA is that iACK makes efficient use of the broadcast nature of the wireless channel without any additional packet overhead.

SWIA retransmits every packet that gets lost. In a congested network, such indiscriminate retransmission may aggravate the problem of channel contention and network congestion. Moreover, most sensor network applications aiming to achieve reliability can afford some degree of packet loss. Hence, attempting to retransmit every single lost packet may be both unnecessary and detrimental to the reliability of the network.

### *3.10. Reliable Bursty Convergecast (RBC)*

Zhang *et al.* [26] proposed Reliable Bursty Convergecast (RBC) protocol, designed for challenging situations where a large burst of packets from different locations needs to be reliably transported to the sink in a real time. The main design considerations for RBC are the need to improve channel utilization, to reduce ACK loss, alleviate retransmission incurred channel contention and implementing adaptive timers mechanism for retransmission

timeouts. It employs iACKs, using the list of queues as window-less blocks to place the packets that have been transmitted and have a priority scheme which allows the packets that have been retransmitted less number of times, to be transmitted first, if there are some nodes requesting for them.

The window-less block acknowledgement mechanism detects and notifies the packet loss more reliably and energy efficiently. The receiver piggybacks information of all successfully received packets in the header of each data packet to be further forwarded. The sender listens to get such information and starts retransmission, if required. If the sender does not get any expected information for a specific time interval, and if the timer for each transmitted packet is expired then sender will start *hop-by-hop* retransmissions. The value of the retransmission timer depends on the queue length at the next hop. RBC performs intra-node and inter-node packet scheduling to avoid congestion caused by retransmission. In intra-node packet scheduling, each node maintains a virtual queue giving high priority to the packets with less number of retransmissions. However, inter-node packet scheduling implements contention control by allowing a node with more packets in buffer to have higher priority to have more chances to access the channel.

In RBC, reliability is achieved by retransmission of missing packets. The retransmission is done for all missing packets which might be unnecessary if packets carrying similar information have already reached the sink. Moreover, the use of priority schemes to schedule transmission could make the packets with the low priority to wait forever in the queue, resulting in high latency. Energy modeling is one of the most important issue of WSNs, which is ignored by the authors. In case of a fragment loss, in RBC, the receiver would not be able to request the retransmission of a fragment without having a NACK mechanism.

### *3.11. Energy-efficient and Reliable Transport Protocol (ERTP)*

Le *et al.* [27] proposes ERTP, an energy efficient transport protocol providing *end-to-end* statistical reliability (determined by the quantity of data packets received at the sink rather than the reliability of each data packet) of the data from sensor to sink for a WSN application producing streaming data. It uses Stop-and-Wait Implicit Acknowledgement [1] mechanism to recover the lost packets while *end-to-end* reliability is achieved by checking the reliability at each hop and use of ACK to confirm packet reception at the

sink. ERTP dynamically minimizes the number of retransmissions to reduce the energy consumption. Thus, introduces a retransmission timeout estimation mechanism in which the node's waiting time for iACK at each hop is dynamically adjusted in terms of time taken for the downstream nodes to forward the packet. This results in significant decrease in energy consumption which is the one of the main goal of ERTP.

One of the major criticism of Le's work is that it assumes low transmission rate where network congestion is negligible and it assumes to use low power listening (LPL) MAC protocol for implicit acknowledgements where instead of idle listening the node periodically sleep, wake up and listen and then return to sleep. This results in low overhearing, where insufficient number of retransmissions results in loss of packets. The authors ignore the presence of network congestion which might be an interesting observation.

### *3.12. Tunable Reliability with Congestion Control for Information Transport (TRCCIT)*

Shaikh *et al.* [28] reported Tuneable Reliability with Congestion Control for Information Transport (TRCCIT), aiming to provide desired reliability in varying network conditions and application requirements by using an adaptive retransmission mechanism. This helps to reduce the unnecessary retransmissions which in turn help to reduce congestion; whereas, spatial path redundancy and congestion is further controlled by adapting different paths for information flow. A localized hybrid acknowledgement (HACK) mechanism and a timer management system are used to provide required reliability in an adaptive fashion. A congestion control mechanism is implemented that detects and reduces the congestion while maintaining the required reliability level.

The HACK is used as a combination of implicit and explicit acknowledgement mechanisms. In HACK, if the sender does not get any iACK from the next hop and the timer is expired then sender will retransmit the information. However, if the next hop after receiving the information from the sender, decides to suppress the information in order to maintain hop level reliability; it will send an eACK to the sender. In this way, the sender will not perform unnecessary retransmissions and will be informed that the information is being suppressed to fulfil the reliability requirements. The adaptive retransmission timeout mechanism is based on the buffer occupancy and time

to send outgoing messages of the sender node. It calculates the number of incoming and outgoing messages at each node and creates a balance between them to control congestion. Once the number of incoming messages becomes greater than the outgoing messages, the nodes then increase the outgoing flow by sending the messages on multiple paths to control congestion. It uses RTS/CTS to avoid the situation where due to the busy channel; nodes are not able to send the messages.

Shaikh et al. claims to provide tuneable reliability for a large scale sensor network; however, it performs simulations only for 100 nodes which are scattered in an area of $60m \times 60m$ which is not large enough in terms of both number of nodes and network area.

### *3.13. Reliable Transport with Memory Consideration (RTMC)*

Zhou *et al.* [29] proposes Reliable Transport with Memory Consideration (RTMC), a reliable transport protocol considering limited memory and unreliable links in wireless sensor networks. RTMC performs segmentations on the files to be transmitted, where the node doesn't transmit the files until the next hop node's memory gets free. Furthermore, it makes sure that sink has reliably received all the segments. To prevent memory overflow, the header of the packets are loaded with the memory information which is shared among the neighbors.

RTMC inspired from the idea of pipe flow, where liquid can flow from source container to the destination container until it is full. Similarly, in RTMC source node transfers the segments of a file to destination node through the relay nodes, until the memory of a relay node is full. To deal with this situation, the transmission rate at the source is dynamically adjusted. The segments currently stored are the ones which are received and not acknowledged by the next relay node. When the current node's memory is empty then it asks the previous node to send packets. When all the segments are sent, then the source node tells the intermediate nodes that the transmission is finished.

One of the limitations in with this protocol is that, if more number of nodes starts sending the file segments at the same time then it might become a cause of huge time delay and will need more memory space. In RTMC the experiments are performed on real time, but only six sensor nodes are

deployed in a line, where the first node is assumed to be the source node and the last is the base station. This scheme might not work properly if a number of senor nodes are increased and there are many sources sending the segments to the base station. The authors' claims to achieve congestion control; however there is no explicit congestion control mechanism used in this scheme. The authors tried to avoid congestion, but they did not take the long data transfer delay into consideration.

### *3.14. Reliable Multi-Segment Transport (RMST)*

Stann and Heidemann's [30] Reliable Multi-Segment Transport (RMST) is a transport protocol designed to make directed diffusion reliable. It benefits from diffusion routing and guarantees delivery and provides fragmentation and assembly for applications that require them. It has both caching and non-caching mode where the caching mode will store packets that can be retransmitted when other nodes requests for it.

Reliability is achieved by retransmission of missing packets when their timers expire. These retransmissions are done regardless of what and where the packets are from. As such, packets which have similar data from others which have successfully reached the sink would be retransmitted even though missing some of these similar packets would not have made the data collection any inaccurate. RMST uses NACK as a control message but does not handle the possibility of NACK implosion when downstream nodes issue a chain of NACK requests for holes detected in the non-caching mode. It also does not look into the problem where NACK based schemes can only work if the receiver receives at least one packet from the sender. Reliability is hence also not guaranteed for single packet messages. RMST only seems suitable for the some multimedia applications needs to send large size sensory data such as image files which requires fragmentation at the source and reassembly at the destination (sink).

### *3.15. Pump Slowly Fetch Quickly (PSFQ)*

Wan *et al.* [31] identifies a *hop-by-hop* downstream reliability mechanism, Pump Slowly Fetch Quickly (PSFQ). It aims to achieve 100% reliability guarantee in order to disseminate the control messages or software code segments towards downstream sensor nodes, while performing the loss detection and

recovery operation, and local retransmissions of the lost data. As the name indicates, in PSFQ, the source node sends data at a slow speed (pump slowly), whereas nodes that has packet losses will fetch any missing packets from its immediate neighbor very aggressively (fetch quickly). It is designed for the applications that require full data delivery reliability such as reprogramming of sensors. It uses three operations, including pump operation, fetch operation and report operation.

The base station broadcasts the code segments slowly to the downstream nodes at regular intervals, during the pump operation. Then the fetch operation starts as the sequence number gap is found by any intermediate nodes. The nodes will stop forwarding new packets when there are missing packets and will send NACKs to their neighbors to ask for retransmission of the missing packets. It will only resume the forwarding process when the missing packet has been retrieved from its neighbors. Finally, report operation provides the status report to the use when a report message is sent from the farthest node in the network towards the requesting user. Whereas, each node along the way adds its report to the original report message in an aggregated manner till it reaches the user.

Computation overheads on each node is quite high because there a lot of timers to keep track of, whereas as timers are crucial to start pump and fetch operations. PSFQ requires all the nodes to cache all the packets received in order for the reprogramming of sensor nodes to succeed. In a normal sensor network, nodes have limited memory and thus they will not be able to cache all the packets that they received. PSFQ does not take into account of the temporal and spatial information of each packet. If the sink has already received the information about a particular area, extra information about that area would be just an incentive but not necessity. Broadcasting done in both pump and fetch operation, making the network congested and results in energy wastage. PSFQ is more suited for nodes reprogramming and this application is not that common because the nodes are seldom programmed.

### *3.16. GARUDA: Achieving Effective Reliability for Downstream Communication*

Like PSFQ, Park *et al.* [32] presented another downstream reliability protocol, GARUDA, which aims to reliably deliver query-metadata and the control codes to downstream nodes. Control codes are transmitted for the

reprogramming of the downstream nodes, whereas query-metadata is saved in the downstream sensor nodes for query management. GARUDA appoints the nodes in the network as core nodes for retransmission of lost packets and makes use of A-map to prevent NACK implosion. A-map is a bit map which states which of the packets a core node has and is able to retransmit to the nodes that request for it. It uses the Wait-for-First-Packet (WFP) pulse, so that it can also send data that has only one single packet.

| RETRANSMISSIONS-BASED SCHEMES | | | | | |
|---|---|---|---|---|---|
| **Protocols** | **Traffic Flow** | **Reliability Level** | **Loss Recovery** | **Loss Detection & Notification** | **Sink Centric** |
| Mahmood *et al.* [ERP] | Up | Event | Hop-by-hop | iACK | No |
| Akan *et al.* [17] ESRT | Up | Event | End-to-end | ~ | Yes |
| Gungor *et al.* [18] DST | Up | Event | End-to-end | ~ | Yes |
| Iyer *et al.* [19] STCP | Up | Packet/Event | End-to-end | ACK/NACK | Yes |
| Zhou *et al.* [20] PORT | Up | Event | Hop-by-hop | ~ | Yes |
| Tezcan *et al.* [21] ART | Up/Down | Event | End-to-end | ACK/NACK | Yes |
| Xue *et al.* [22] LTRES | Up | Event | End-to-end | ACK/NACK | Yes |
| Hosung *et al.* [23] QERP | Up | Event | End-to-end | ~ | Yes |
| Maroti [1] SWIA | Up/Down | Packet | Hop-by-hop | iACK | No |
| Zhang *et al.* [25] RBC | Up | Packet | Hop-by-hop | iACK | Yes |
| Le *et al.* [26] ERTP | Up | Packet | Hop-by-hop | iACK/ACK | Yes |
| Shaikh *et al.* [27] TRCCIT | Up | Packet | Hop-by-hop | iACK/ACK | Yes |
| Zhou *et al.* [28] RTMC | Up | Packet | End-to-end | ~ | Yes |
| Stann *et al.* [29] RMST | Up | Packet | Hop-by-hop | NACK | Yes |
| Marchi *et al.* [10] DTSN | Up | Packet | End-to-end | ACK/SACK | Yes |
| Wan *et al.* [30] PSFQ | Down | Packet | Hop-by-hop | NACK | Yes |
| Park *et al.* [31] GARUDA | Down | Packet | Hop-by-hop | NACK | Yes |

Figure 5: Comparison Table for Retransmission-based Reliability Schemes in WSNs

GARUDA achieves reliability through the retransmission of packets from the core nodes. Hence, there is an extra core construction stage for GARUDA. The WFP required also means that special hardware has to built into the sensors and generating these pulses would also consume extra energy. This is a better approach for a small size network; however, loss recovery and core

creation would create high latency in case of large scale sensor networks. It does not offer any congestion control system and the reliability guarantee is provided only for the transfer of the first packet.

### *3.17. Summary*

In this section, we briefly described the most important approaches proposed to solve the problem of reliability faced by WSNs through the use retransmissions mechanism. We described, critically analyzed and compared related work according to different functions provided at different levels. These functions such as achieving *packet* or *event* reliability through *hop-by-hop* or *end-to-end* retransmissions serve as a criterion for comparison between existing research work.

We discussed the most of the existing schemes responsible for providing data transport reliability in WSNs. Different schemes aim to achieve different levels of reliability based on the application requirements. [17, 18, 19, 21, 22, 23] aims to achieve *event* reliability while recovering the lost packets based on the *end-to-end* retransmissions, whereas [17, 18, 23] doesn't explicitly provides any loss detection and notification mechanism. *Event* reliability tends to be a more energy efficient strategy when performed with *hop-by-hop* retransmissions, but requires an efficient cache management system at each hop. Whereas, in *end-to-end* retransmissions the packet stays in cache of the source node till acknowledged by the sink which consumes more energy especially in large networks. Thus *hop-by-hop event* reliability [24, 20] performs better than *end-to-end event* reliability in terms of less energy consumption, lower communication cost, quicker loss recovery and fast congestion alleviation. [24] is the only *hop-by-hop event* reliability mechanism that performs loss detection and notification by utilizing the broadcast characteristic of wireless channel using implicit acknowledgements while [19, 21, 22] uses ACK/NACK mechanism that puts extra message overhead over the network. However, [20] doesn't provide any loss detection and notification system. All the *event* reliability schemes [17, 18, 19, 20, 21, 22, 23] except [24] rely at the sink to regulate the flow of data to attain certain application specific required reliability instead of taking simple decisions at the node level and saving the cost of communication.

Although the use of *event* reliability in WSNs is more energy efficient but, many applications require the guaranteed delivery of each and every

packet through the use of *packet* reliability. Major portion of the existing data transport reliability schemes [19, 26, 27, 28, 29, 30, 10, 31, 32] aims to achieve *packet* reliability. Like *event* reliability, *hop-by-hop packet* reliability schemes [26, 1, 27, 28, 30] performs better than *end-to-end packet* reliability schemes [19, 29, 10]. The performance of *packet* reliability schemes [26, 27, 28] are further improved by using iACKs, while [19, 30, 10] faces high traffic overhead due to ACK/NACK/SACK mechanisms.

Like the upstream reliability, downstream reliability schemes [31, 32] also plays an important role. Among all the existing schemes only [21] performs both upstream and downstream reliability, whereas [19] is the most flexible in terms of providing both *event* and *packet* reliability depending on the application requirement.

## 4. Redundancy-based Reliability

In a wired or wireless network, retransmission is the most commonly used techniques in order to recover from packet loss and achieve data transport reliability. Apart from wired and wireless networks, wireless sensor networks are also able perform retransmission of the lost packets to achieve reliability as employed by most of the schemes discussed in the previous section 3. Unlike wireless networks, wired networks consider a bit level loss or corruption within a packet as a loss of the entire packet, which requires the retransmission of the whole packet in order to achieve reliable data transmission. However, in wireless networks these small bit errors in a packet can be recovered by performing Forward Error Correction (FEC) techniques. FEC techniques are used for controlling errors caused by the data transmission over unreliable or noisy communication channels [6]. The FEC techniques are able to correct only the lost or corrupted bits within the packet. This significantly reduces the transmission overhead caused by the acknowledgements and retransmissions of the entire packet even if just a single bit of that packet was lost. Keeping in view the resource constrained nature of WSNs, employing FEC will reduce the overall cost of network in terms of energy, memory usage and increase network lifetime at the cost of information redundancy.

Rizzo [2] highlights the importance of using Forward Error Correction (FEC) techniques instead of commonly used Automatic Repeat reQuest

(ARQ) techniques using acknowledgements and retransmissions in order to provide better reliability in computer communications. Erasure coding as one of the FEC techniques is described, implemented and evaluated by Rizzo [2] for a computer network. FEC techniques are used to perform error detection and corrections at the lower layers of the protocol stack for telecommunication and computer communication (e.g. CRC in computer communication) systems. However, when it comes to upper layers (e.g. link, transport and application layers), erasure codes are needed to deal with the missing packets since it is easier to figure out the exact position of the lost data at these layers. The author provides a good introduction to Reed-Solomon codes in his paper, but he has done the implementations on desktop computers network. However, by carefully choosing the optimization parameters, this work can provide help in implementation of erasure codes in resource constrained wireless sensor networks.

In erasure coding, the sender divided the data packets into $m$ fragments which are further encoded by adding $k$ redundant fragment. These encoded $k$ redundant fragments in addition to $m$ original fragments *(m + k)* are transmitted towards the destination. The destination (i.e. the sink), decodes the encoded set of fragments it has received and tries to reconstruct the original data out of it. The destination is only able to reconstruct the original data out of the received encoded segments if the fragments it received i.e. *m'* is greater than or equal to the number of original fragments i.e. *m*. This process of *end-to-end* erasure coding is adapted only on the source and the destination (i.e. the sink), while all the intermediate nodes act as simple relay nodes. However, if we perform erasure coding in *hop-by-hop* fashion, the encoding/decoding is performed at every intermediate node to reduce redundant data which is also known as full coding.

Traditional *end-to-end* erasure coding calculates the number of redundant fragments on the basis of the probability of successful data arrival for the whole path through to the sink. However, in *hop-by-hop* erasure coding the number of redundant fragments to be added at the $i^{th}$ hop is calculated on the basis of the probability of successful data arrival at $i^{th}$ hop instead of the whole path. Therefore, *end-to-end* erasure coding produces high number of redundant fragments as compared to *hop-by-hop* erasure coding, increasing the overall network traffic with uneven load balancing. Most of the existing research, employing information redundancy in order to achieve reliability in WSNs are depicted below.

### *4.1. Retransmission or Reliability: Transmission Reliability in WSNs*

Wen *et al.* [7] theoretically evaluates the performance of retransmission and redundancy mechanisms in order to provide energy efficient data transmission reliability. Data delivery reliability is evaluated in terms of packet arrival probability by checking it against required reliability, while energy efficiency is evaluated in terms of energy consumed in successful delivery of one packet.

The authors model the loss behavior using Gilbert model [8], which is basically designed for one hop channel. However, for a multi-hop network it is assumed that the two-state Gilbert model is adopted independently at each hop of a multi-hop transmission. Successful packet arrival probability through multiple *(n)* hops and average energy consumption for one successful packet arrival is theoretically calculated and analyzed for simple transmissions, retransmissions and erasure coding mechanisms. The packet arrival probability and energy consumption are mainly compared against the unconditional loss probability *(ulp)*, which is the probability that the next packet will be lost based on the condition that the previous packet was lost. Finally, erasure coding is evaluated to be better than the *hop-by-hop* retransmission mechanism in terms of energy efficiency only on a condition that *ulp* value is below a certain threshold value.

This theoretical analysis shows that erasure coding mechanism performs better than the retransmission mechanism in terms of reliability and energy efficiency in an environment with a low packet loss rate and less number of hops. However, the performance of erasure coding can be improved in a lossy environment by increasing packet redundancy at the cost of energy consumption. The results in this paper are evaluated and analyzed only on the basis of theoretical calculations which may be different when performed practically.

### *4.2. Optimum Reed-Solomon Erasure Coding*

Ali *et al.* [9] proposed another theoretical analysis to achieve reliability using erasure codes in a distributed wireless sensor network. The source node creates and transmits the fragments of a packet using Reed-Solomon codes, a class of erasure codes, along multiple paths towards the sink where they are reconstructed. The authors give the idea of calculating the cost

of transmitting these fragments under different network conditions where Genetic Algorithms *(GAs)* are used to calculate the optimized number of fragments to be transmitted accordingly.

This scheme assumes a query-based wireless sensor network, where the sink sends the query to the prongs which are the one hop nodes from the sink. The prongs then further broadcast the queries throughout the network. The relay nodes keep track of the path through which they received the queries. This helps the relay nodes to send the response of the queries back the sink through prongs, while selecting the path with short hop counts or high reliability. Whereas, the authors have not employed any path selection process and assumes that it is already implemented.

It is observed in the numerical analysis that by increasing the number of parity fragments (additional fragments) reliability is increased at the cost of high fragments transmissions; however, the probability of packet loss decreases. Therefore, intelligent selection of number of redundant bits becomes the optimization factor in Reed-Solomon codes, which results in increase in overall reliability and energy efficiency. The author aims to use genetic algorithms for the selection of the optimized number of fragments to minimize the overall cost of high fragment transmission and packet loss.

This is a good idea of using $GAs$ in orders to select the optimized number of fragments to perform erasure coding. However, the authors have ignored the fact that intensive processing required by the selection process of $GAs$ which might not be suitable for resource constrained WSNs. Moreover, the proposed idea is not validated by any simulations or practical implementation on real test beds, which raises a question that how well $GAs$ performs with WSNs. The encoding/decoding is performed only at the source and destination which would result in high probability of packet loss in case of high hop count. The authors have not introduced any downstream reliability mechanism in order to perform query processing, whereas, this query/response mechanism would further increase the network traffic. The idea of introducing prongs doesn't seem to be suitable as it will create an extra processing overhead on the nodes acting as prongs, while making no use of the extra processing capabilities of the sink.

*4.3. Distributed Transport for Sensor Networks (DTSN)*

Marchi *et al.* [10] proposed DTSN, a reliable transport protocol that claims to provide energy efficient upstream reliable data transmission for wireless sensor networks in a distributed fashion. DTSN provides reliability in terms of full and differentiated reliability services. The full reliability mode is aimed to provide full *end-to-end packet* reliability when all the packets are required to be delivered to the sink. However, in cases where full reliability is not required, a differential reliability mechanism is added to the basic DTSN with addition of a Forward Error Correction (FEC) strategy.

Full reliability level is provided with the help of Selective Repeat ARQ mechanism, using ACK/NACK for loss recovery and packet control, where the packets are cached at the intermediate nodes to reduce *end-to-end* retransmissions. The source transmit packets until the number of packets becomes equal to the size of Acknowledgement Window (AW) and sends an Explicit Acknowledgement Request (EAR) to the destination for the confirmation message delivery (ACK or NACK). An ACK is sent if the sequence of the packets is in order and those packets are suppressed from the output buffer. However, if NACK is received then retransmissions are required for the missing sequence of packets.

On the other hand, differential reliability level is employed using ARQ along with Enhancement Flow and FEC strategies. Enhancement Flow strategy is used to transfer a block of data like still image, the basic image (i.e. core portion of the image with minimum resolution) is buffered at the source and reliably transferred using FEC. However, rest of the image (remaining part of the complete block of data) is not provided with any reliability guarantee, because, it is the enhanced part of the data (i.e. image with maximum resolution). This Enhancement Flow along with the FEC mechanism results in high reliability and improved throughput as compared to full reliability strategy with the complete block transfer.

Caching at intermediate nodes requires large storage which would not be suitable for resource constrained WSNs; therefore, a cache management mechanism is required to minimize the unnecessary cache occupancy. Furthermore, the ACK/NACK mechanism results in high message overhead which affects the overall energy efficiency. An overhearing mechanism can be introduced to reduce this message overhead. DTSN provides a mechanism to transfer the core part of data, but does not give the details of how

the size of the core data is determined and reliability level is maintained in changing network conditions. The authors have not provided the details of which FEC techniques they have used for transferring the core part of data. It is mentioned that encoding/decoding is performed at source and the sink node, where intermediate nodes act as relay nodes. This is not suitable for a sensor network with large number of hops between source node and the sink, as this will introduce a lot of entry points for errors and packet losses along the way.

### *4.4. Reliable Transfer on WSNs*

Although most of the studies on introducing information redundancy to achieve reliability in WSNs have only focused on the theoretical analysis, while Kim *et al.* [11] have practically validated the use of erasure codes in WSNs on real test beds. The authors claimed to provide high *packet* reliability in a multi-hop wireless sensor network by employing the right combination of retransmission; erasure coding and route fix mechanisms. The aim is to provide *packet* reliability, where the maximum numbers of packets are required to be received at the sink. This can be done either by increasing the number of packets sent or by achieving the probability of successful packet transmission. The number of sent packets can either be increased by increasing the redundant data or number of retransmissions. However, probability of successful data transmission can be achieved by minimizing the packet loss ratio. In link-level retransmissions, implicit acknowledgements makes the best use of wireless channel, but for that the nodes require buffer space to keep the packets until they get acknowledged. This results in high latency and buffer overflow in case of *end-to-end* retransmissions. To counter this high cost latency and buffer overflow for retransmission, redundancy scheme such as erasure coding can be applied at the cost of high traffic. However, the performance of erasure coding deteriorates in a high loss environment; therefore, finding alternative routes when there is continuous packet loss at the current path would be a better option.

The authors use systematic codes, produced by Vandermonde matrix which is a property of Reed-Solomon codes, a particular erasure coding algorithm. In systematic codes, the node does not need to perform great deal of computation for restructuring the original message. This is because, if a part of encoded message is the original message itself then the receiving

node can recover it without performing any decoding. This will not only reduce the node's processing, but also increase reliability and reduces node's memory usage, as all the processing is done on the fly. Systematic codes are practically implemented on real test beds by labeling the transmitted packets either as an original or a redundant packet. It is observed that increasing retransmissions would improve reliability till a certain level at the cost of network congestion and queue overflow; however, increasing redundancy would also add the network cost but will also significantly increases reliability and loss tolerance. It is also observed that erasure coding performed better than the retransmissions mechanism at the cost of high bandwidth and overhead; however, in case of continuous losses, finding alternate route would be a better choice. Therefore, it is required to combine retransmissions, redundancy and route-fix in an intelligent manner to have perfect reliability mechanism.

The authors implemented the proposed work on real test beds which provide more authentic and reliable results. The tests are performed on a specifically designated source and destination, but not for the case where a number of sources from the network are randomly sending data to the sink. This might become the cause of high traffic and delay as more nodes will try to contend the channel simultaneously to transmit data.

### *4.5. Reliable erasure-coding based Data Transfer Scheme*

Most recently, Srouji *et al.* [12] introduces RDTS, an efficient *hop-by-hop* data transfer reliability mechanism using erasure coding scheme. Unlike other schemes that performed erasure coding only at source and the sink nodes, RDTS implements erasure coding at each hop to attain *hop-by-hop* reliability. In addition to erasure coding, RDTS applies the partial coding mechanism in terms of reducing the computational overhead triggered by erasure coding at each hop.

Partial coding mechanism further reduces the effect of performing complete erasure coding process at each intermediate node. This makes sure that receiver receives and transmits enough data fragments to the next hop instead of always decoding, reconstructing and encoding the missing fragments at each hop. Thus, partial coding performs decoding and encoding only when the original fragments are missing and the received fragments are not enough.

RDTS is compared with *hop-by-hop* erasure coding and *end-to-end* erasure coding, where it shows better performance in terms of energy consumption, traffic overhead, load-balancing and network lifetime. RDTS calculates number of fragments needed for the next hop on the basis of probability of successful data arrival, but this probability is not actually calculated using a specific technique rather taken as random values from 0.7 to 0.95. There should be a mechanism to calculate these probabilities dynamically.

### *4.6. On Improving Wireless Broadcast Reliability for Sensor Networks using Erasure Codes*

Most of the schemes discussed so far, provide high reliability using erasure codes where the data is flowing in an upstream direction. However, Kumar *et al.* [13] claims to provide high reliability, especially in terms of transmitting data in a downstream traffic flow in a large WSN. When the software needs to be updated, the software codes are broadcasted to the sensor networks results in high message overhead, and thus affects reliability due to lossy wireless channel [14]. Whereas, software update in WSNs requires 100% reliability for proper functioning of the network. The existing baseline approaches for broadcasting requires extra information about the network, which is expansive. Keeping in mind the improved processing capabilities of sensor nodes, Kumar *et al.* [13] proposes FBcast, a baseline broadcasting protocol based on FEC and compared it with probabilistic broadcast mechanism [15]. FBcast provides high reliability with low message overhead as compared to probabilistic broadcast, without processing extra information about the network.

Kumar *et al.* uses fountain codes, a class of erasure codes, producing encoded block of data on-the-fly without having any limitation on the packet size or symbol size, where a single unit of data may comprise of any number of bytes. The encoded data is broadcasted to the neighbors, which is further rebroadcasted, only on a condition that the received fragments are new. This simple FBcast protocol assumes that all the nodes are one hop away from the source, thus source broadcasts the data in a single broadcast message. This is not a feasible solution for a multi-hop WSN, thus an extended version of FBcast protocol is also proposed with the concept of repeaters. The repeater functionality is applied on the motes, where the motes after receiving enough data fragments, reconstructs the original data and then encode

it again before further rebroadcasting to the remaining hops. However, for probabilistic broadcasting, the repeater mote simply retransmits the original packets. FBcast with repeaters gives high reliability as well as better coverage as compared to simple FBcast and probabilistic broadcast mechanisms.

The authors proposed a downstream reliability protocol in shape of FBcast, but they have not considered comparing it with other existing WSNs downstream reliability protocols. Doing this would give a better picture of how good the proposed protocol is. The idea of implementing erasure codes in terms of ensuring downstream reliability might be new, but performing encoding/decoding at each and every downstream node of the network for successfully broadcasting the data throughout the network would be an expensive choice.

| REDUNDANCY-BASED SCHEMES | | | | | | |
|---|---|---|---|---|---|---|
| **Protocols** | **Traffic Flow** | **Reliability Level** | **Encoding / Decodong** | **Coding Scheme** | **Performance Evaluation** | **Supportive Mechanisms** |
| Wen *et al.* [7] | Up | Packet | End-to-end | Erasure Codes | Theoretical Analysis | ULP |
| Ali *et al.* [9] | Up | Packet | End-to-end | Erasure Codes | Theoretical Analysis | Genetic Algorithms |
| Marchi *et al.* [10] | Up | Packet | End-to-end | Erasure Codes | Simulations | Enhansement Flow |
| Kim *et al.* [11] | Up | Packet | End-to-end | Erasure & Systematic Codes | Real test beds | Alternate Route-Fix |
| Srouji *et al.* [12] | Up | Packet | Hop-by-hop | Erasure Codes | Simulations | Partial Coding |
| Kumar et al. [13] | Down | Packet | Hop-by-hop | Erasure Codes | Simulations | FB Cast |

Figure 6: Comparison Table for Redundancy-based Reliability Schemes in WSNs

### *4.7. Summary*

This section highlights the importance of FEC techniques in WSN in order to achieve reliability, followed by a brief overview of how encoding and decoding is performed through the use of erasure coding mechanism. This section mainly provides an overview and critical analysis of the existing schemes using erasure coding in order to achieve better reliability.

There are very few schemes in the field of WSNs relying on redundancy mechanisms in order to achieve data transport reliability. We have discussed some of the redundancy based schemes using erasure codes, where [2] first tested the erasure codes in computer communication which gave the idea

of implementing erasure codes in resource constrained WSNs. Among the existing redundancy based schemes, [7, 9, 10, 11, 12] aims to achieve upstream *packet* reliability, while only [13] focuses on downstream reliability. All of them claim to achieve high reliability at the cost of message overhead, where [7, 9] performed theoretical analysis and [10, 12, 13] performed simulations to evaluate their schemes. However, only [11] practically tested erasure codes on real test beds in a WSNs and claims to achieve almost 100% reliability. This gives the confidence of using erasure coding mechanism in terms of further enhancing data transport reliability in field of WSNs. Most recently, only [12] popped up the idea of performing erasure coding in a hop-by-hop fashion which performed better than *end-to-end* erasure coding in terms of reduced coding overhead and increased network lifetime.

## 5. Conclusion & Challenges Ahead

WSNs hold the promise of many new applications in the field of monitoring and control systems. With the advent of cheap and small sensors, monitoring systems can make use of them to monitor numerous characteristics of the environment. All these applications are built for specific purposes, where maintaining data transport reliability is one of the most important challenges until now.

To address the reliability issue in sensor networks, we surveyed the various existing protocols that manage data transport reliability and each of them has its unique way of ensuring reliability. Some of them require full *end-to-end packet* reliability while others can tolerate some degree of packet loss depending on the nature of application they are working on. The most commonly used techniques to ensure *packet* or *event* reliability includes the use of retransmissions mechanism while others use information redundancy to ensure reliability.

We have provided the summarized comparison of exiting data transport reliability protocols, where figure 5 shows most of the protocols achieve reliability on the basis of retransmissions, while figure 6 show others that are based on redundancy using FEC techniques in order to enhance reliability. Retransmission mechanism is pretty straightforward where schemes using *hop-by-hop* retransmissions [24, 1, 20, 26, 27, 28, 30] seems to be more suitable for wireless sensor networks as compared *end-to-end* retransmission

based schemes [17, 18, 19, 21, 22, 23, 29, 10].

*Hop-by-hop* retransmission mechanisms perform better as compared to *end-to-end* mechanisms in lossy environments, especially when there is a high hop count from source to destination. The high hop count introduces more entry points for errors which become the cause of packet loss, thus affecting reliability. *Hop-by-hop* mechanism performs better, as loss recovery is performed at each intermediate hop from source to the destination. The *hop-by-hop* mechanisms can further be improved when it is performed for achieving *event* reliability [24, 20]. Requirement of a dynamic retransmission timeout mechanism and efficient buffer management system for preventing stale packets occupying the queue would be an advantage to the existing protocols in order to save sensors' limited resources.

Most of the existing *event* reliability protocols have ignored the importance of accurate event detection which is one of the major characteristic of an event driven wireless sensor network. They do not consider the complexity of Identifying more information about the event with the help of which the overall network traffic could be reduced by transferring only the information about the unique events towards the destination. Once the events are accurately identified then events should be efficiently prioritized on the basis of their information, where high priority events should be transmitted first. Therefore, the accurate event detection and redefining the priorities before its reliable dissemination is one of the important research areas that needs to be further explored. Additionally, most of the protocols do not specifically deal with the problem of declining network coverage in high levels of congestion. To address this unique challenge, the nodes required to suppress the retransmission when the lost packets are found to be in the same sensing area as other packets soon to be transmitted. This idea can be exploited through the use of spatio-temporal correlation [42] among the closely located sensor nodes.

Most of the existing reliability protocols have been evaluated on the basis of simulation. However, real time implementations are needed to be performed to have more realistic evaluation of the protocols. Developments in cross layer designs is another future research direction, where interaction with the MAC layer would help in solving the channel contention problems while network layer would help to find the best route or the alternative route in case of congestion. In addition to reliability, congestion control also plays an

important role in minimizing the packet loss which can enhance reliability. Although few protocols implemented both congestion control and reliability, but this area also needs to be investigated properly. All WSNs data transport reliability protocols consider applications working in hostile environments; however, all of them have ignored considering security risks due to malicious attacks. Furthermore, the existing reliability protocols lacks the use of in-node data processing, instead they rely at the sink to regulate the flow of data in order to achieve reliability. This increases the communication which is the highest power consumer in a wireless network.

In addition to retransmissions, information redundancy is another way of achieving data transport reliability which is so far ignored by most of the existing reliability schemes except [10]. Only few of the schemes have used redundancy techniques to achieve reliability in WSNs, where most of them have just performed theoretical analysis [7, 9]. Only [11] have validated the importance of redundancy-based reliability on real test beds, while [10, 12, 13] have validated their schemes through the use of simulations. Only [10] have proposed a combination of retransmission and redundancy, but the authors have not provided the details of which FEC techniques they have used for decoding/encoding and how it is performed. DTSN claims to provide *end-to-end* redundancy on core part of data to be transmitted and retransmission-based reliability on rest of the data. Whereas, *hop-by-hop* redundancy [12] would provide much better reliability as compared to *end-to-end* redundancy by refreshing the data at each hop. We believe that an intelligent combination of retransmissions and redundancy schemes to achieve high level reliability will be the center of future research. This diverts our attention towards achieving data transport reliability in wireless multimedia sensor networks and WSNs powered by energy harvesting [43], which is mostly unexplored in terms of reliability.

## References


[1] M. Maróti, "Directed flood-routing framework for wireless sensor networks," in *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware* (Middleware), Toronto, Canada, pp. 99–114, 18-22 October 2004.

[2] L. Rizzo, "Effective erasure codes for reliable computer communication

protocols," *ACM Computer Communication Review*, vol. 27, no. 02, pp. 24 – 36, 1997.

[3] J. Postel, "Transmission Control Protocol," *IETF RFC 793*, September 1981.

[4] C.-Y. Wan, A. Campbell and L. Krishnamurthy, "Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 04, pp. 862 – 872, 2005.

[5] S.-J. Park, R. Sivakumar, I. Akyildiz and R. Vedantham, "GARUDA: Achieving effective reliability for downstream communication in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 07, no. 02, pp. 214 –230, 2008.

[6] G. Clark, *Error-Correction Coding for Digital Communications*, Plenum Press, 1981.

[7] H. Wen, C. Lin, F. Ren, Y. Yue and X. Huang, "Retransmission or Redundancy: transmission reliability in wireless sensor networks," in *IEEE 4th Internatonal Conference on Mobile Adhoc and Sensor Systems* (MASS), Pisa, Italy, pp. 1 – 7, 08-11 October 2007.

[8] J. C. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications* (SIGCOMM), San Francisco, California, USA, pp. 289–298, 13-17 September 2005.

[9] S. Ali, A. Fakoorian and H. Taheri, "Optimum Reed-Solomon erasure coding in fault tolerant sensor networks," in *Proceedings of the 4th International Symposium on Wireless Communication Systems* (ISWCS), Trondheim, Norway, pp. 6–10, 16-19 October 2007.

[10] B. Marchi, A. Grilo and M. Nunes, "DTSN: Distributed transport for sensor networks," in *Proceedings of the 12th IEEE Symposium on Computers and Communications* (ISCC), Aveiro, Portugal, pp. 165–172, 01-04 July 2007.

[11] S. Kim, R. Fonseca and D. Culler, "Reliable transfer on wireless sensor networks," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks* (IEEE SECON), Santa Clara, California, USA, pp. 449–459, 04-07 October 2004.

[12] M. S. Srouji, Z. Wang and J. Henkel, "RDTS: A Reliable Erasure-Coding Based Data Transfer Scheme for Wireless Sensor Networks," in *Proceedings of the 17th International Conference on Parallel and Distributed Systems* (ICPADS), Tainan, Taiwan, pp. 481–488, 07-09 December 2011.

[13] R. Kumar, A. Paul, U. Ramachandran and D. Kotz, "On improving wireless broadcast reliability of sensor networks using erasure codes," in *Mobile Ad-hoc and Sensor Networks* (MSN), ser. Lecture Notes in Computer Science, J. Cao, I. Stojmenovic, X. Jia and S. K. Das, Eds. Springer Berlin Heidelberg, vol. 4325, pp. 155-170, 2006.

[14] K. Obraczka, K. Viswanath and G. Tsudik, "Flooding for reliable multicast in multi-hop ad hoc networks," *Wireless Networks*, vol. 7, no. 6, pp. 627-634, 2001.

[15] W. Peng and X.-C. Lu "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing* (MobiHoc), Boston, Massachusetts, USA, pp. 129-130, 11 August 2000.

[16] P. Levis, N. Patel, D. Culler and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation* (NSDI), San Francisco, California, pp. 15–28, 29-31 March 2004.

[17] O. B. Akan and I. F. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 05, pp. 1003–1016, 2005.

[18] V. C. Gungor and O. B. Akan "DST: delay sensitive transport in wireless sensor networks," in *Proceedings of the 7th International Symposium on Computer Networks* (ISCN), Istanbul, Turkey, pp. 116-122, 16-18 June 2006.

[19] Y. Iyer, S. Gandham and S. Venkatesan, "STCP: a generic transport layer protocol for wireless sensor networks," in *Proceedings of 14th International Conference on Computer Communications and Networks* (ICCCN), San Diego, California, USA, pp. 449 – 454, 17-19 October 2005.

[20] Y. Zhou, M. Lyu, J. Liu and H. Wang, "PORT: a price-oriented reliable transport protocol for wireless sensor networks," in *16th IEEE International Symposium on Software Reliability Engineering* (ISSRE), Chicago, Illinois, USA, pp. 10 pp. –126, 08-11 November 2005.

[21] N. Tezcan and W. Wang, "ART: an asymmetric and reliable transport mechanism for wireless sensor networks," *International Journal of Sensor Networks*, vol. 02, no. 3-4, pp. 188–200, 2007.

[22] Y. Xue, B. Ramamurthy and Y. Wang, "LTRES: A loss-tolerant reliable event sensing protocol for wireless sensor networks," *Computer Communications*, vol. 32, no. 15, pp. 1666 – 1676, 2009.

[23] H. Park, J. Lee, S. Oh, Y. Yim, S. Kim and K. Nam, "QERP: Quality-based event reliability protocol in wireless sensor networks," in *Proceedings of IEEE Consumer Communications and Networking Conference* (CCNC), Las Vegas, Nevada, USA, pp. 730–734, 09-12 January 2011.

[24] M. A. Mahmood and W. K. G. Seah, "Event Reliability in Wireless Sensor Networks," in *Proceedings of the seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing* (ISSNIP), Adelaide, Australia, pp. 01– 06, 06-09 December 2011.

[25] S. Dalvi, A. Sahoo and A. Deo, "A MAC-Aware Energy Efficient Reliable Transport Protocol for Wireless Sensor Networks," in *Proceedings of IEEE Wireless Communications and Networking Conference* (WCNC), Budapest, Hungary, pp. 1–6, 05-08 April 2009.

[26] H. Zhang, A. Arora, Y.-r. Choi and M. G. Gouda, "RBC: Reliable bursty convergecast in wireless sensor networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing* (MobiHoc), Urbana-Champaign, Illinois, USA, pp. 266–276, 25-28 May 2005.

[27] T. Le, W. Hu, P. Corke and S. Jha, "ERTP: Energy-efficient and reliable transport protocol for data streaming in wireless sensor networks," *Computer Communications*, vol. 32, no. 7-10, pp. 1154 – 1171, 2009.

[28] F. Shaikh, A. Khelil, A. Ali and V. Suri, "Trccit: Tunable reliability with congestion control for information transport in wireless sensor networks," in *5th Annual International ICST Wireless Internet Conference* (WICON), Singapore, pp. 1 – 9, 01-03 March 2010.

[29] H. Zhou, X. Guan and C. Wu, "RTMC: Reliable transport with memory consideration in wireless sensor networks," in *IEEE International Conference on Communications* (ICC), Beijing, China, pp. 2819 –2824, 19-23 May 2008.

[30] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications* (SNPA), Anchorage, Alaska, USA, pp. 102 – 112, 11 May 2003.

[31] C.-Y. Wan, A. Campbell and L. Krishnamurthy, "Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 04, pp. 862 – 872, 2005.

[32] S.-J. Park, R. Sivakumar, I. Akyildiz and R. Vedantham, "GARUDA: Achieving effective reliability for downstream communication in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 07, no. 02, pp. 214 –230, 2008.

[33] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[34] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292-2330, 2008.

[35] T. He, S. Krishnamurthy, J. A.Stankovic, T. F. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui and B. Krogh, "An energy-efficient surveillance system using wireless sensor networks," in *Proceedings of the 2nd international conference on Mobile Systems, Applications, and*

*Services* (MobiSys), Boston, Massachusetts, USA, pp. 270–283, 06-09 June 2004.

[36] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the 2nd international conference on Embedded Networked Sensor Systems* (SenSys), Baltimore, Maryland, pp. 214–226, 03-05 November 2004.

[37] K. A. Bispo, L. H. A. De Freitas, N. S. Rosa and P. R. F. Cunha, "A Reconfiguration Approach Using Ontologies to Save Energy on WSNs," in *Proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (UBICOMM), Sliema, Malta, pp. 182–187, 11-16 October 2009.

[38] A. Silberstein, R. Braynard, G. Filpus, G. Puggioni, A. Gelfand, K. Munagala, J. Yang, B. Kamachari, D. Estrin and S. Wicker, "Data Driven Processing in Sensor Networks," in *Proceedings of the Third Biennial Conference on Innovative Data Systems Research* (CIDR), Asilomar, California, USA, pp. 10–21, 07-10 January 2007.

[39] Y. Chen, H. V. Leong, M. Xu, J. Cao, K. Chan and A. Chan, "In-Network Data Processing for Wireless Sensor Networks," in *Proceedings of the the 7th International Conference on Mobile Data Management* (MDM), Nara, Japan, pp. 26, 10-12 May 2006.

[40] X. Tang and J. Xu "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications* (INFOCOM), Barcelona, Catalunya, Spain, pp. 01–12, 23-29 April 2006.

[41] R. Gonzalez and M. Acosta, "Evaluating the impact of acknowledgment strategies on message delivery rate in wireless sensor networks," in *IEEE Latin-American Conference on Communications* (LATINCOM), Bogota, Colombia, pp. 1 –6, 15-17 September 2010.

[42] M. C. Vuran, . B. Akan and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," in *Computer Networks*, vol. 45, no. 03, pp. 245–259, 2004.

[43] W. K. G. Seah, Z. A. Eu and H.-P. Tan, "Wireless Sensor Networks Powered by Ambient Energy Harvesting (WSN-HEAP) - Survey and Challenges," in *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology* (Wireless VITAE), Aalborg, Denmark, 17-20 May 2009.