A Survey on Reliability Protocols in Wireless Sensor Networks

MOHAMED AMINE KAFI, DTISI, CERIST Research Center, Algiers, Algeria
JALEL BEN OTHMAN, Laboratoire L2TI, Université de Paris 13, Paris, France
NADJIB BADACHE, CERIST Research Center, Algiers, Algeria

Wireless Sensor Network (WSN) applications have become more and more attractive with the miniaturization of circuits and the large variety of sensors. The different application domains, especially critical fields of WSN use, make the reliability of data acquisition and communication a hot research field that must be tackled efficiently. Indeed, the quality of largely used, cheap-cost wireless sensors and their scarce energy supply support these reliability challenges that lead to data loss or corruption. For solving this problem, the conception of a reliability mechanism that detects these shortcomings and recovers to them becomes necessary. In this article, we present a survey on existing reliability protocols conceived especially for WSNs due to their special features. The deep classification and discussion in this study allow for understanding the pros and cons of state-of-the-art works in order to enhance the existing schemes and fill the gaps. We have classified the works according to the required level of reliability, the manner to identify the origins of the lack of reliability, and the control to recover this lack of reliability. Across the discussion along this study, we deduce that the cross-layer design between MAC, routing, and transport layers presents a good concept to efficiently overcome the different reliability holes.

CCS Concepts: • **Networks** → **Sensor networks;**

Additional Key Words and Phrases: Wireless sensor networks (WSNs), reliability, transport layer protocols, redundancy, retransmission

ACM Reference Format:
Mohamed Amine Kafi, Jalel Ben Othman, and Nadjib Badache. 2017. A survey on reliability protocols in wireless sensor networks. ACM Comput. Surv. 50, 2, Article 31 (May 2017), 47 pages.
DOI: http://dx.doi.org/10.1145/3064004

1. INTRODUCTION

A wireless sensor network is a set of hundreds or thousands of tiny devices scattered in order to monitor a specific area and forward the happening events to a specific unit, named the sink or basestation [Sharma and Aseri 2012; Shaikh et al. 2007]. The main advantage of using a WSN, which has become feasible with microcircuitry evolution, is that each sensor is a low-cost device with very limited capacities in terms of processing, memory storage, communication range, and power supply (even with harvesting capacities) but which brings its strength from being used cooperatively with other ones through the construction of a large, self-organized, multihop network to sense accurately critical events happening [Sharma and Aseri 2012; Rahman et al. 2008; Katiyar et al. 2012; Pereira et al. 2007]. Numerous monitoring applications could be concretized using WSNs, from civil ones (e.g., home automation and surveillance, fire

Authors' addresses: M. A. Kafi and N. Badache, 05 Rue des 3 Frères Aissou, Ben Aknoun, Algies, Algeria; emails: kafiamine@gmail.com, badache@cerist.dz; J. Ben Othman, 99 Avenue Jean-Baptiste Clément, 93430 Villetaneuse; email: jalel.ben-othman@univ-paris13.fr.

DOI: http://dx.doi.org/10.1145/3064004

detection, precision irrigation [Ouadjaout et al. 2014], healthcare application [Kafi et al. 2015], industrial processing and control, traffic control [Kafi et al. 2012, 2013], etc.) to military-based applications (e.g., battlefield monitoring, country borders surveillance, enemy tracking, etc.).

Two types of flows characterize the WSN communications: the collected data from the events' sources toward the sink, and the dissemination of control commands, data queries, and nodes' code reprogramming from the sink toward the entire or a part of the network [Ayadi 2011]. These two flows' types require a certain reliability level that differs according to the data packets' types and application criticality. For example, code reprogramming packets do not tolerate losses compared to query-based ones; also, the collected data regarding military information need more reliability than irrigation information data [Shin et al. 2007]. From the node characteristics point of view, reliability could be seen in relation to its energy supply, whose depletion leads to node permanent nonfunctionality unless harvesting capabilities are possible, or to node failure, which could be due to environmental causes such as fire or physical breakage [Silva et al. 2012]. These two previous causes result in permanent node failure. On the other hand, link failure could be a cause of transient packet losses. In general, link failures could be related to the quality of low-cost sensors that characterize most of the market's sensors and that leads to unreliable, low-bandwidth, error-prone, and time-varying wireless mediums, or due to the harsh environment that dominates the use of WSN including noise, interference, packet collisions, signal attenuation, channel fading, and near-ground positioning [Katiyar et al. 2012; Wang et al. 2006; Mahmood et al. 2015].

The challenge of WSNs is to ensure reliable applications, through reliable data forwarding, over the use of unreliable media and nodes [Shaikh et al. 2007]. In order to fulfill this goal of reliability, at least transport and network protocols must provide these functionalities, in addition to the combination of fault tolerance mechanisms. From the network level, usual node failures, leading to path and topology changes and segmented routing, must be treated to diminish related packet loss and thus nodes' energy depletion [Katiyar et al. 2012]. Also, from a transport point of view, congestion must be treated to avoid related losses, and the bandwidth fairness allocation must be guaranteed, especially for faraway nodes [Sharma and Aseri 2012; Wang et al. 2006; Mahmood et al. 2015]. Indeed, sensor nodes could have different communication capacities that lead to packet loss due to buffer overflow congestion when a sender node transmits to its receiver with a higher rate than its reception capacity [Kafi et al. 2014; Ghaffari 2015; Sergiou et al. 2014]. This phenomenon becomes more severe as packets travel toward the sink, which degrades application reliability and performance [Katiyar et al. 2012]. There remain the problems of noise and error-prone links for which recovery-based methods (like retransmission and redundancy) could be the appropriate remedy [Mahmood et al. 2015].

WSN communications approaches differ compared to traditional wired networks due to their density, large node number, and multihop nature, which are justified by the short range, scarce energy supply, computational capacities, and memory storage of sensor nodes. The events' collected data at sources should be transported in a reliable multihop network toward the sink. This needs routing reliability mechanisms to face topology changes due to mobility, link, and node failures [Silva et al. 2012; Mahmood et al. 2015; Woo et al. 2003]. Indeed, single-path routing could present limitations that make the use of multipath strategies a mandatory solution to overcome reliability-related problems [Suganya et al. 2013]. Also, as application specification varies from one to another, QoS demands change too. This leads to design routing specification according to the application requirements. In fact, challenges of time-critical applications (e.g., target tracking) differ from those of habitat monitoring and so on [Marjan et al. 2012].

In this article, reliability mechanisms used in the state of the art of the WSN field are presented and surveyed. In addition, taxonomies are proposed in order to ease decision making at the application conception moment. Also, shortcomings' explicitness makes the proposition of new mechanisms more suitable. The remainder of this survey study is organized as follows. In Section 2, we present previous reliability-related studies. Section 3 highlights terminology of reliability components. In Section 4, metrics to evaluate the reliability of proposed mechanisms are discussed. Different existing reliable protocols in WSN with their classifications are presented in Section 5. We analyze and discuss reliability behaviors in Section 6. Finally, we conclude and give a summarization of the study in Section 7.

2. RELATED WORK

The reliability field is a very important axis in the WSN conception, as this can be seen through the state-of-the-art works [Sharma and Aseri 2012; Shaikh et al. 2007; Rahman et al. 2008; Katiyar et al. 2012; Pereira et al. 2007]. These works cover reliability surveys, reliability modeling, and comparative studies.

2.1. Reliability Surveys

Rathnayaka and Potdar [2013] propose a generic framework for WSN transport protocols in order to allow unified evaluation. They present technical and experimental benchmark attributes to use to evaluate protocols regarding the application scale-based view. Their scheme eases the validation of new proposition protocols by comparing their performance evaluation directly against the existing ones in a same application-based scenario. Mahmood et al. [2015] examine reliability protocols based on retransmissions and redundancy mechanisms and give a deep presentation on different works. In Suganya et al. [2013], multipath routing is surveyed. The authors analyze the performance of various multipath routing protocols for reliability purposes. Zafar [2011] focuses on problems faced by TCP in wireless networks in general and WSN especially, and highlight some extensions of TCP and other new protocols dedicated to WSNs.

2.2. Reliability Modeling

Shaikh et al. [2007] categorize faults during data transport into intolerable faults, which cannot be handled by transport protocols (like sink failure), or tolerable faults, which can be handled by transport protocols (like link or some node failures). They also propose a reliability block model in order to simplify the investigation of the transport problem. Bein et al. [2005] present Markov models for reliability and investigate the replacement of failed sensors. Antônio et al. [2014] propose a model to assess WSN reliability, taking into account the battery level. They base their model on routing algorithms and evaluate the reliability of different regions in the WSN regarding the power consumption parameter in order to make appropriate decisions, like the addition of new nodes. In Silva et al. [2012], a methodology based on an automatic generation of a fault tree is presented with the aim to evaluate the reliability and availability of WSNs in the presence of permanent failures on devices. The proposed scheme allows one to model different topologies, redundancy levels, arbitrary failure occurrences, and devices' importance. The authors highlight many works that are interested in modeling the devices' failure using different modeling methods (Markov chains, Petri nets, FTA, etc.). In Chen and Wang [2012], the reliability problem seen from a security point of view is treated. The authors try to maximize the system lifetime through energy consumption minimization, but without decreasing reliability performance, by identifying the needed code attestation verification frequency and the number of sensor code verifiers in each event attestation. In Zhou [2012], a system model is proposed where the integrated task model takes into account energy consumption and message delay for safety-critical scenarios. So, both energy and time factors are considered in the task-integrated reliability model. Jin et al. [2012] establish a relation between WSN reliability and network parameters (topology and acquisition rate) in order not to exceed the network capacity, which would lead to buffer overflow and congestion, while ensuring a sufficient reliability level. Also, in the case these calculated network parameters could not lead to a satisfactory reliability level, methods like changing topology, decreased sensory data acquisition level, and improving nodes' capacity calculations are proposed. Jaggle et al. [2009] show that through knowledge of link probabilities and single sensor node lifetime distributions, the reliability analysis model of the whole WSN is performed. AboElFotoh et al. [2006] show that the reliability measure for an arbitrary WSN, based on an estimation of the data generation rate and each node's failure probability, is #P-hard. They therefore propose an exponential algorithm and compute the reliability for data-gathering applications.

2.3. Comparative Studies on Reliability

2.3.1. Analytical-Based Comparative Studies. Wen et al. [2007] conduct an analytical comparative study between retransmissions and erasure code (redundancy)-based recovery mechanisms. They also study the packet arrival probability and average energy consumption for the both approaches. They conclude that erasure code was the best one, in terms of energy consumption and reliability, in low- and moderate-loss probability scenarios, whereas retransmission is the more appropriate one when loss probability reaches or exceeds a certain level. In this high-loss case, or with the increasing hop number, erasure code shows a worse performance. Therefore, in these last scenarios, erasure code, which seems energy efficient, sacrifices reliability for energy consumption. Zonouz et al. [2013] treat the k-coverage WSN reliability by doing an analytical evaluation where they compare two single-path routers, namely, shortest-path-distance and shortest-path-hop algorithms. The evaluation was done regarding communication reliability and energy consumption by varying the node density, channel conditions, and monitored areas. The authors' conclusions were that the shortest-path-distance algorithm is more reliable than the shortest-path-hop algorithm but consumes more energy for communication.

2.3.2. Experimental-Based Comparative Studies. Reason and Rabaey [2004] present the results of an experimental study done for a moderate-size, multihop WSN. The study focuses on radio energy consumption and packet reliability. The energy measurements were established for each traffic type and radio state. Kim et al. [2004] perform experimental tests using mica2dot motes in order to show the advantage of using retransmission, erasure codes, and alternative next-hop route, or their combinations, to overcome packet loss. Their results claim the usefulness of each method to face different failure causes. They show, through real tests, the utility of combining, intelligently, the previous methods to reach as high as 99% reliability with a low overhead. Korkmaz and Sarac [2010] show, through their experimental IRIS mote tests, that the combination of link-level retransmission with multihop routing constitutes a good reliability solution for small-length-path-based WSNs. However, with the increasing path-length networks, this solution becomes costly in order to get an acceptable reliability level. The authors try to quantify the consequence of their use in large-scale path-length WSNs and propose the use of hierarchical routing structures in large-scale WSNs and to take link reliability quality as a mandatory factor in the paths' construction. The experimental study done by Zhao and Govindan [2003], using 60 micaz motes, has as its goal the quantification of packet delivery in WSNs. For this reason, the authors carefully control network topology to take a clear view of its reaction on packet delivery. The real tests were done in indoor and outdoor scenarios for the examination of different encoding schemes and a wider variety of performance characteristics.

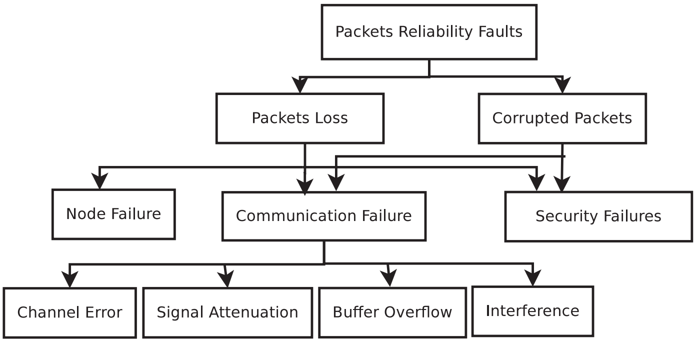

Fig. 1. Packet reliability fault classes.

2.3.3. Simulation-Based Comparative Studies. In Shaikh et al. [2008], a comparative study, through simulation of three reliability protocols in different but similar scenarios, is done to show their efficiency. Interesting results regarding different parameters were presented. The Shin et al. [2007] comparative simulation study goal was to highlight the factors that limit packet delivery reliability in dense WSNs and to propose simple but efficient enhancements to existing protocols' stack mechanisms in order to ameliorate their performance.

In Jukka et al. [2009], the authors' simulations had as a goal the determination of end-to-end reliability, latency, and availability in a low-duty-cycle WSN. The authors study the consequences of Beacon losses on nodes' synchronization, which is essential for good performance. They also study the effects of network load, packet errors, and buffer sizes.

Even with the large number of works in the reliability field conceived for WSNs, there still exists a lack of a deep view that allows the conception of a solid mechanism for different causes of loss and corruption of data. This can be seen through the angle from which each work targets the reliability concept. In this study, we try to highlight reliability requirements from different corners, thus giving the interaction between the proposed mechanisms to allow the conception of a new powerful mechanism to face the lack in reliability.

In the following section, we will present terminologies related to reliability mechanisms in WSNs and the different kinds of reliabilities.

3. RELIABILITY-RELATED TERMINOLOGY

In WSNs, reliability assurance is done by avoiding or recovering faults due to communication, node, or security failures. These faults lead to packet loss or corruption, as well as high latency that also degrades reliability. The communication failures are due, in general, to interference, congestion, collisions, and buffer overflow, whereas node failures lead to momentary or permanent network path failures that alter data forwarding and lead to topology change [Shaikh et al. 2007]. From the security failures side, it can result in reception of corrupted packets that could change application behavior and trigger wrong alerts or reactions. This could be due to the lack of protection for physical nodes (e.g., battlefield, fire, etc.) or to a lack of security communication mechanisms. In Figure 1, the previous packet reliability faults are depicted. In this section, we present different reliability terminologies related to the required reliability level by the application; the direction of reliability regarding the packets' transmission, the packets' loss detection, and notification methods; and the recovery mechanisms.

3.1. Reliability Level

The diversity of WSN applications gave birth to different QoS exigencies. From the packet delivery point of view, different requirements could be planned in terms of loss and delay [Sharma and Aseri 2012]. The importance of these two parameters leads to different kinds of delivery priorities. For example, real-time applications (object tracking, hazard detection, etc.) necessitate delay-sensitive delivery more than the one-packet delivery, and thus, some losses are tolerated. On the other hand, code reprogramming applications are interested more in the delivery of each packet in order to have a correct program version, so tolerating some delay but no packet loss is allowed. In the following, packet loss tolerance is used to classify the resulting QoS requirements:

3.1.1. Packet Reliability. In such application requirements, the delivery of each packet from its source to the concerned destination is a must. This policy could lead to high overhead and energy consumption, but at the same time could be mandatory to the application work. In order to ensure this kind of reliability delivery, careful loss detection and congestion avoidance in addition to recovery mechanisms have to be put in practice [Sharma and Aseri 2012].

3.1.2. Event Reliability. At the event happening, its successful forwarding to the sink is essential to ensure application fidelity. In such cases, some individual packets losses are tolerated as long as the sink receives sufficient information about the concerned event. In fact, the loss nature of WSN nodes' communication pushes ahead to put redundant sensing, from different nodes, in order to overcome the resulting forwarding failures. Compared to the packet reliability paradigm, the event reliability leads to less overhead and energy depletion. But note that this kind of reliability is not always possible, especially in code reprogramming scenarios, or when the event coverage is ensured by a small number of nodes [Sharma and Aseri 2012]. In this kind of reliability, aggregation mechanisms bring a lot of benefit as they allow the combination of sensor readings, therefore reducing the overall overhead [Jones and Atiquzzaman 2007]. Also, the required reliability level in event forwarding operations could be fixed by a delivery probability or threshold method in order to reach the appropriate level [Willig and Karl 2005].

3.1.3. Destination Reliability. In this kind of delivery, both packet or event reliability could be considered. The packets have to reach a certain region rather than the whole network, with a defined reliable level [Rathnayaka and Potdar 2013]. So, in the case of the packet reliability scenario, the reprogramming of the sensors in this region could be an example. On the other hand, the event forwarding of specific monitored regions from a whole large-scale WSN could be considered also.

3.2. Reliability Direction

The aforementioned reliability levels concern different directions and are treated according to the communication pattern. The reliability directions can be classified as follows:

3.2.1. Upstream Reliability. In this communication direction, the packets are sent from the sensor source nodes toward the sink node and concern the forwarding of the collected data [Sharma and Aseri 2012]. This transmission is also known as the convergecast, unicast, or many-to-one communication [Rahman et al. 2008].

3.2.2. Downstream Reliability. This type of communication concerns the packets to be sent from the sink node to the sensor nodes. The sink usually sends control packets, query demands, and reprogramming codes [Sharma and Aseri 2012]. As the sink node sends, generally, the same packets to all the nodes, the transmission is done using

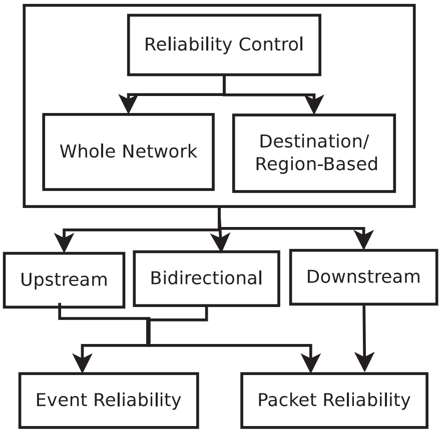

Fig. 2. Different directions and levels of reliability.

broadcasting or multicasting rather than unicasting and is also known as the one-to-many communication [Rahman et al. 2008].

3.2.3. Bidirectional Reliability. Some protocols try to ensure the two previous reliabilities, namely, the upstream and downstream reliabilities. The benefit of using the same protocol for the two directions is to ensure low complexity, control heterogeneity, and decrease the packets' control overhead [Sharma and Aseri 2012].

In Figure 2, a scheme highlighting reliability directions and levels is depicted.

3.3. Loss Detection and Notification

The packet loss detection is the first step of any reliability protocol in order to ensure appropriate recovery. After that, it is followed by notifying the adequate node to initiate the recovery part as a third step. Depending on the mechanism used to detect the reliability failures, a sender-based or a receiver-based detection is envisaged. The overhead resulting from the used mechanism also depends on the loss frequency. The loss notification is needed if a receiver-based mechanism is used for loss detection and a recovery mechanism is envisaged at the sender side. According to the recovery mechanism used, a hop-by-hop detection and eventually notification, or end-to-end one, are provided.

3.3.1. Sender-Based Mechanism. When the sender transmits a packet, it can hear its successful forwarding through its parent toward the grandparent, in the upstream direction, or through its child toward its grandchild (two-hop child), in the downstream direction. This can be considered as an Implicit Acknowledgment (IACK). The nonoverhearing of this forwarding operation during a predefined time period could be used as a loss detection strategy. This timeout policy could also be needed if simple positive acknowledgments (ACK) are used at the receiver. In this situation, the receiver does not detect any loss but simply acknowledges the reception of packets. It is the charge of the sender to detect any loss by the observation of sequence number gaps or timeouts [Wang et al. 2006]. The use of packet sequence numbering is mandatory with any acknowledgment-based mechanism [Rathnayaka and Potdar 2013], as can be seen in the following section. The hearing method is used in a hop-by-hop manner, allows detecting single packet loss, and is named implicit detection. Basic positive ACK could be used in a hop-by-hop or end-to-end manner, allows detecting single or multiple packet loss, and is named explicit detection. With the use of ACK, overhead is due

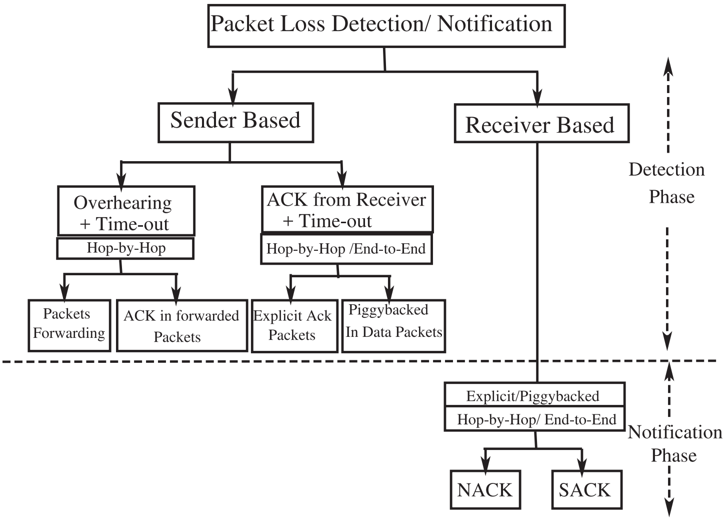

Fig. 3. Reliability detection and notification subclasses.

to the headers used and sequence numbering. On the other hand, with the hearing method, overhead is due to the passive listening and the nonuse of duty cycling. Also, with the hearing method, piggybacking ACK information allows for detecting multiple losses rather than single losses with only forwarding packets. If aggregation methods are used, the simple overhearing could not be applied for loss detection. In this case, a piggybacking of ACK information about the well-received packets, in the header of the aggregated packet, could be heard by the sender. But another problem of implicit hearing is that the sender must wait until listening for the well forwarding, which could take time, as the receiver queue contains other packets.

3.3.2. Receiver-Based Mechanism. In this detection type, the receiver detects the loss by the gaps presented in the packets' sequence number [Rathnayaka and Potdar 2013] or by the timeout mechanism triggered after a nonreceiving period. The receiver informs explicitly the previous-hop forwarder about this loss if a hop-by-hop method is used, or the loss is forwarded in the reverse path until reaching the source sender with end-to-end behaviors. In this case, the acknowledgment method is named negative ACK (NACK), because it mentions not-received packets. The overhead resulting from this mechanism depends on the NACK type used. But decreasing the sending frequency of NACK toward the sender, by compacting different losses in the same NACK, known also as loss window, could lead to long delay notifications. Thus, appropriate conception, taking into account application specifications, is of high importance. Selective ACK could be used at the receiver side also. In this case, the receiver detects the gaps in sequence numbers and responds back to the sender with an ACK packet that contains the last packet received in the order and the list of the other packets received in disorder. If the transmission is done in a bidirectional direction, piggybacking ACK in data information toward the sender will be energy efficient.

In Figure 3, a scheme depicting the different detection and notification subclasses is presented.

3.4. Error Recovery

In order to ensure application fidelity and depending on the required reliability level (packet or event based), a recovery mechanism is essential to replace the lost packets.

This mechanism could be retransmission based or redundancy based [Mahmood et al. 2015]. The retransmission-based mechanism is also named ARQ based (Automatic Repeat Request) if it is done at the MAC level, while redundancy based on coding mechanisms is also named FEC (Forward Error Control) [Carle and Biersack 1997]. On the other hand, and from the place the recovery is done, the recovery could be hop by hop or end to end [Sharma and Aseri 2012].

In retransmission-based recovery, the concerned sender node retransmits the missing packets after the loss happening. In the redundancy-based mechanism, the concerned sender node adds supplementary information packets to the original ones, from the beginning of the transmission, that the receiver node exploits in the loss case in order to reconstitute the lost packets. The redundancy-based approach could be applied by sending multiple copies of the same packet or by using the coding and decoding method. This method of coding and decoding the packets is also named erasure coding. The aim is to encode m fragment packets into $m + k$ ones but that allow reconstituting the m packets if ones from the $m + k$ are lost (at least m fragments received) [Mahmood et al. 2015].

Carle and Biersack [1997] describe the use of a combination between ARQ and FEC mechanisms for IP-based networks in order to overcome the shortcomings of both mechanisms, named hybrid ARQ type 1 and hybrid ARQ type 2. In the type 1 mechanism, the sender starts sending redundant data from the beginning, but if the loss rate is still high and the original packet could not be reconstructed, the receiver asks for retransmission of missing packets in order to recover the original ones. In the type 2 mechanism, the sender starts transmitting redundant data by the first retransmission request.

But the problem that could be confronted in the redundancy-based mechanism is that even with the multiple copies or coding/decoding, losses could be not recovered, which requires the use of retransmission if the reliability must be ensured.

In the hop-by-hop recovery mechanism, every intermediate node is responsible to initiate the recovery procedure, namely, caching and retransmitting the lost packets if the retransmission-based method is used, or sending multiple copies (or even encoding/decoding) if the redundancy method is applied. Hop-by-hop recovery is most suitable for WSNs due to the multihop communication paradigm. The hop-by-hop recovery allows a fast operation and avoids wastage of the previous hops' energy resource due to an error occurring on the latter hops. But at the same time, the hop-by-hop recovery requests more memory in terms of caching packets or coding/decoding operations. On the other hand, if end-to-end recovery is implemented, the recovery mechanism is only initiated at the source and sink nodes, whereas the intermediate nodes only relay the packets [Mahmood et al. 2015]. End-to-end recovery causes higher delay and offers less throughput compared to hop-by-hop recovery. Also, the end-to-end erasure coding demands more redundancy packets than the hop-by-hop method, which leads to more traffic, even though the coding/decoding is performed one time rather than at every hop [Mahmood et al. 2015]. The end-to-end recovery consumes energy as the packet recovery means the loss of the previous hops' efforts. This overhead becomes higher as the network size scales, as does the loss rate. For example, with a link loss rate of 10%, the loss becomes 80% at the sink distance of 15 hops, which is a tolerable hop count for WSNs [Kim et al. 2004].

Figure 4 highlights the different recovery mechanisms. But trying to recover lost packets cannot give good results without taking into account the error causes. For example, if the loss is due to buffer overflow, a congestion control must be applied before recovering the lost packets. If the cause is path failure, finding a backup path or momentarily waiting for path re-establishment has to be considered. If, on the other hand, the loss is due to channel error, there is no benefit to reduce sources' rates before recovering the losses, as this will underuse the channel capacity, lowering the

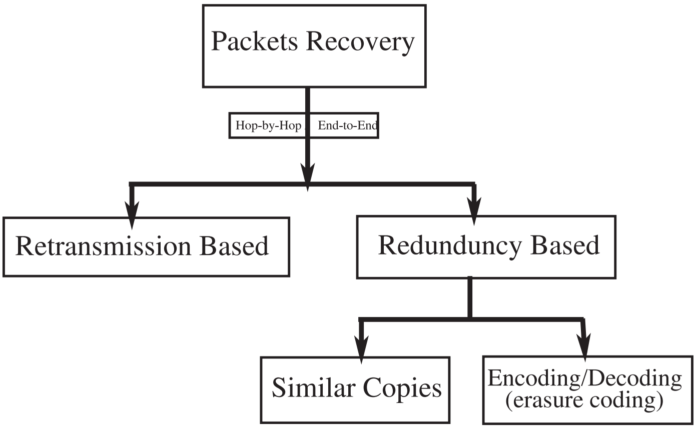

Fig. 4. Recovery mechanism scheme.

throughput [Wang et al. 2006]. As MAC layer retransmissions, which are done in a hop-by-hop manner, recover well from channel error loss but not from buffer overflow loss, and as network layer information clears up the path state, the idea of a cross-layer reliability mechanism at the transport layer, taking into account lower-layer information, reveals a better candidate than an alone-layer solution [Pereira et al. 2007; Willig and Karl 2005].

In the following section, we will describe the metrics used to evaluate the state-of-the-art reliability protocols, which are used also by the protocols described in Section 5.

4. RELIABILITY EVALUATION METRICS

For evaluating any proposed protocol that ensures reliability, many parameters were envisaged in the literature and used in simulation or by experimentation. In the following section, the most-used evaluation parameters are presented.

Event Reliability: This reflects how well the event is reported to the basestation [Rahman et al. 2008]. It is measured as the ratio of received events to the number of generated ones [Shaikh et al. 2008].

Node Reliability: This is the ratio of the number of received packets at the sink for a node to the total number of generated packets by that node [Zhang et al. 2007]. Some protocols use the complementary of this ratio, named *packet loss ratio*. It is calculated as the ratio of lost packets in the network to the number of generated packets by the node [Rahman et al. 2008].

Average Delivery Overhead: This value quantifies the number of packets needed in order to correctly receive the data packet [Wan et al. 2005]. In some protocols, this value concerns only control packets, like the acknowledgment packet (ACK or NACK) [Rahman et al. 2008]. Another variant of overhead, which is complementary to it, is *protocol efficiency*. It is measured as message complexity and calculates the total number of packets to deliver an event successfully to the sink, including retransmissions [Shaikh et al. 2008]. *Redundancy,* which is the number of duplicate packets during a time period, also quantifies the overhead and resource wastage [Rahman et al. 2008]. Some protocols use *the energy loss per node or per the whole network* in order to quantify the delivery overhead [Shaikh et al. 2008]. This energy loss has a direct impact on *network lifetime*, which is also defined as the period during which the network is well functional [Rahman et al. 2008]. In Le et al. [2009], *energy consumption* is used rather than energy loss in order to quantify the required energy for delivering packets to the basestation. The high overhead results in resource wastage like energy, processing,

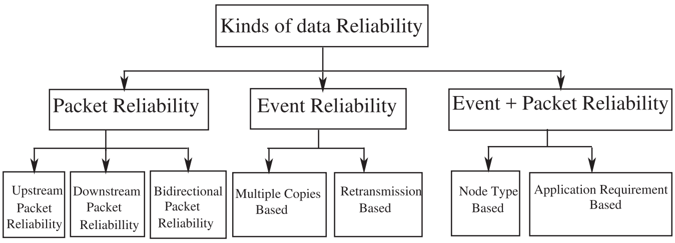
Kinds of data Reliability
Packet Reliability
Event Reliability
Event + Packet Reliability
Upstream Packet Reliability
Downstream Packet Reliabillity
Bidirectional Packet Reliability
Multiple Copies Based
Retransmission Based
Node Type Based
Application Requirement Based

Fig. 5. Data reliability classes.

and transmission collisions. In Kim et al. [2004], *the overhead* is defined as *the cost* per hop per successful delivered packet.

Deadline Miss Ratio: This quantifies the efficiency of the protocol for real-time requirements. It is calculated as the ratio of packets that reach the basestation during a time deadline to the total number of received packets. It is generally expressed as a percentage [Rahman et al. 2008]. In some protocols, the time elapsed from the generation of the event message to its arrival at the sink is used to quantify *the timeliness* of the protocol through *the average latencies* of all generated events [Shaikh et al. 2008]. In Zhang et al. [2007], *event goodput* is defined as the ratio of unique packets received at the basestation to the necessary time period to receive these packets. *The deadline metric* reflects the delays resulting from queuing, retransmissions, and propagation delay [Sharif et al. 2010]. In our previous work [Kafi et al. 2014a], delay, buffer, and other parameters were used in order to quantify the early congestion detection in WSNs.

In order to have a comprehensive comparison and evaluation, the chosen parameters have to be evaluated in scenarios that take into account the network scalability, since the reaction of any protocol differs with the increasing number of used nodes. The network connectivity is also an important parameter to show the efficiency of the evaluated protocol against network changes and especially bit errors [Shaikh et al. 2008].

5. STATE-OF-THE-ART PROTOCOLS

In the literature, many protocols were proposed in order to enhance reliability [Gungor et al. 2008; Paek and Govindan 2007; Sankarasubramaniam et al. 2003; Sundaresan et al. 2005; Dunkels et al. 2004; Zhou et al. 2005; Iyer et al. 2005; Gungor and Akan 2006; Tezcan and Wang 2007; Kim et al. 2007; Kumar et al. 2006; Stann and Heidemann 2003; Deb et al. 2003; Iyer and Kleinrock 2003; Ganesan et al. 2001]. Some of these protocols focus on the delivery of each individual packet. On the other hand, many protocols are interested in the reliability of the event rather than the individual packets. A third class of protocols is interested in the delivery of individual packets or events regarding the kind of packet itself. In the following sections, the previous classification is used to highlight the state-of-the-art works, in addition to a deeper view concerning each class. In Figure 5, a possible classification for these protocols is shown. In addition, we provide for each class a summary table that highlights the behavior of that class.

A. Packet Reliability Protocols. As mentioned previously, in this class of protocols, the aim is to ensure the delivery of each packet. Some protocols are interested in the delivery of upstream packets, which are in general data packets. Another subclass of protocols is interested in the delivery of downstream packets, which are generally control packets, reprogramming packets, or query packets. A third subclass is interested in delivering the packets of the two directions. As the packet delivery insurance is an obligation in these subclasses of protocols, a retransmission-based recovery mechanism

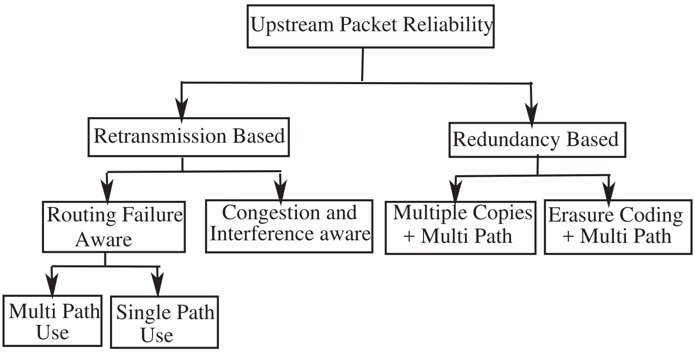
Upstream Packet Reliability
Retransmission Based
Redundancy Based
Routing Failure Aware
Congestion and Interference aware
Multiple Copies + Multi Path
Erasure Coding + Multi Path
Multi Path Use
Single Path Use

Fig. 6. Upstream packet reliability classes.

is the best suited in order to offer reception guarantee, rather than a redundancy-based recovery mechanism, which could fail if the loss reaches or exceeds a given threshold. The recovery could be done in a hop-by-hop or end-to-end manner. But with the hop-by-hop method, the retransmission request could jump to the previous hop if the direct neighbor does not hold this packet. In the following sections, we will describe in more detail the protocols belonging to these subclasses.

A.1. Upstream Packet Reliability Protocols. In this subclass of protocols, data packets that originate from source nodes have to travel toward the sink in a reliable manner that requires a recovery mechanism in case of loss (retransmission based or redundancy based). In addition to the recovery measure, the protocol could be aware of the loss cause resulting into routing-failure aware protocols and congestion-aware protocols. This awareness helps to avoid usefulness recovery in case of persistence of the loss cause. In the following sections, a possible classification is presented, while in Figure 6 a summary scheme is depicted.

A.1.1. Retransmission-Based Protocols. In this subclass of protocols, the recovery of lost packets is done through their retransmission, which could be done in a hop-by-hop or end-to-end manner. Also, as mentioned previously, the protocol could be aware of the loss cause in order to allow for appropriate control to avoid next losses or usefulness retransmission that leads to energy waste.

A.1.1.1. Routing-Failure-Aware Protocols. In this subclass of protocols, losses caused by routing failure are treated in order to choose another path that allows receiving the sent data. In this case, an alternative path could be used after the break of the first one resulting in single-path class protocols, or a priori other constructed/used path(s) could be employed. The choice of new links could be done according to the link quality enhancement rather than total break of the old ones.

A.1.1.1.a) Single-Path-Based Protocols. As mentioned previously, in this category, a single path is used per time, and as the routing failure is happening, another path is employed.

RMST (Reliable Multi-Segment Transport) [Stann and Heidemann 2003] is designed as a transport layer above the Directed Diffusion routing protocol [Intanagonwiwat et al. 2000]. It uses Selective NACK to allow retransmission, with cache at intermediate nodes. RMST enables the use of MAC layer retransmissions in order to decrease the losses. The cross-layer design of RMST makes it aware of route changes using Directed Diffusion behavior. RMST proposes the MAC layer retransmissions with the end-to-end communication or hop-by-hop retransmissions without MAC-ARQ. In the case of hop-by-hop recovery, the intermediate node receiving the NACK packet tries to recover from its own cache. In case of missing packets, it forwards the NACK to the previous-hop neighbor. But RMST does not consider congestion control while trying to recover missing packets, which could affect the performance of retransmissions in many cases. Even RMST works above directed diffusion, but it is not clear how it reacts to route failure happening during the retransmission operations. RMST also lacks details about the use of applications that send-small size messages that stay in single packets.

WISDEN [Xu et al. 2004]: This is designed for continuously collecting vibration information of structures to the basestation using WSN. WISDEN functionalities contain three algorithms: reliable transport, time synchronization, and data compression. Sensor nodes construct a tree-based network toward the basestation that could change during the application lifetime according to link quality and transmit the collected data in a fixed-rate manner. In order to ensure data transmission reliability, WISDEN uses both hop-by-hop and end-to-end retransmissions based on implicit NACKs. The end-to-end recovery part is justified by authors to the topology change, which loses the relation between the parent and child retransmission request, and by the small cache size in case of high loss rate. But as stated by the authors themselves, WISDEN lacks a congestion control mechanism that is in charge of changing the rate adequately to the network state. Even WISDEN is tested in a real scenario, but it is not compared with other reliable transport protocols designed for WSNs. In addition, changing the parent node when this last one asks for retransmissions could be a waste of resources.

CTP (Collection Tree Protocol) [Gnawali et al. 2013] is a collection routing protocol that forwards packets toward the sink in a reliable and efficient manner. CTP provides three techniques in order to ensure its high functioning: an efficient and quick link estimation that allows detecting dynamics in link quality by exploiting physical, link, and network layer information. CTP discovers topology change and failure and avoids loop formation through beacons and data packets. The frequency of beacon control messages changes according to the network dynamics, being so low in stable topology. CTP uses a retransmission policy to enhance reliability.

R3E (Reliable Reactive Routing Enhancement) [Niu et al. 2014] is a reactive routing protocol that aims at increasing path reliability (packet delivery) through remedying links' unreliability. R3E uses a backoff scheme for choosing the opportunistic forwarding nodes and links in the guide path based on the packet reception ratio (PRR) of the links that construct the path. R3E performs well without the use of location information, which leads to additional cost.

Reliable RPL [Ancillotti et al. 2014]: With the emergence of IOT (Internet of Things) applications, LLNs (Low-power Lossy Networks) are becoming more and more successful. IETF groups have proposed many initiatives in different communication layer stacks in order to answer LLN needs. Through the following sections, we will present some of the proposed standardization works by IETF (or community researchers' enhancements), each one in the class that belongs. In this section, we present an enhancement of RPL (Routing Protocol for Low-Power and Lossy Networks). Ancillotti et al. [2014] propose an enhancement for link estimation in order to choose the parent nodes for the RPL [Winter et al. 2012] protocol. The parent is chosen between neighbors according to link estimation methods to put the reliable links in routing tables rather than those based on hop count. The used metrics explore cross-layer information to enhance reliability, and the links between neighbors are evaluated in a probabilistic and adaptive manner to evaluate even the not already chosen nodes with short probing and passive link monitoring methods.

A.1.1.1.b) Multipath-Based Protocols. In this subclass, in addition to retransmissions in loss cases, multiple paths could be used if the losses are due to routing failure.

Rate-Lifetime Tradeoff [Zhu et al. 2008]: In this work, the authors invoke the tradeoff problem between the network lifetime, which leads to transmission minimization to preserve energy, and application performance, which leads to increasing the transmission rate to have more accuracy. The authors formulate this tradeoff as an optimization problem. The reliability is ensured in the proposed study by ensuring hop-by-hop retransmissions, and multipath routing is used in order to enhance the performance. The fairness problem is also treated in order to ensure an homogeneous view of the whole network at the moment of rate attribution. But the authors do not invoke in their scheme any congestion control, which could be a serious cause of data loss.

Improving Lifetime and Reliability in Routing Real Time [Sanaz Naziri and Hasanpoor 2011]: In this work, the authors propose a routing algorithm in order to ensure reliability and decrease energy consumption, for increasing network lifetime. The authors try to balance nodes' energy consumption by using a genetic algorithm in which previous saved routes are used to enhance reliability, in a manner that gives a weight regarding these parameters, namely, energy and reliability. The chromosome in this algorithm is a route from the source toward the sink. The reliability is defined by the probability of successful packet transmission over a path. When data is lost, the buffer is used to store it, and if the actual route fails, other routes are used. But there is a lack of details in the loss detection and notification phases. Also, there is no congestion control in the case of forwarding overhead.

POLY [Qureshi et al. 2011] is a routing protocol that constructs polygons in order to enhance reliability of the formed network. POLY is based on modeling the network as a connected dominating set (CDS) to save network energy by deploying a number of nodes while the others are in sleep state. The deployed nodes named as active ones are those forming a polygon that enhances at the same time the topology reliability by having multiple paths toward the sink. POLY also proposes the topology maintenance by taking into account energy depletion of the active nodes in order to enhance the lifetime of the network. In order to form the polygon-based topology, POLY uses three types of messages: hello messages, finish discovery messages, and topology construction messages. The nodes that do not find their ID in the topology construction messages know that they should stay in sleep state for saving energy.

In Table I, a summary of protocols belonging to the "upstream packet reliability based on retransmissions (routing failure aware)" subclass is highlighted.

A.1.1.2. Congestion/ Interference-Aware Protocols. In this subclass, the protocols are aware of any losses due to congestion or interference. In fact, the congestion could be avoided or controlled after it happens. The fact of being aware about the congestion cause (excessive rate or link interference) could lead to avoidance of lack of reliability, by the use of scheduled communications or by the calculation of the appropriate sending rates. In the case that congestion has already happened, controlling the new rate before triggering the retransmission is important to avoid future packets loss too.

FLUSH [Kim et al. 2007] is a reliable bulk transport protocol for multihop wireless networks. The aim of FLUSH is to ensure packet reliability delivery for applications requiring large data. The sink schedules sequentially the sources to transmit their data for the purpose of avoiding interpath interference leading to congestion and packet loss. For each source, FLUSH tries to maximize the path bandwidth utilization while avoiding intrapath interference and congestion. The first step is to construct a path toward the sink after the sink request for the concerned source. The second step is to maximize the transfer along this path, while ensuring reliability as much as possible

Table I. Summary of Upstream Packet Reliability Based on Retransmission Protocols (Routing Failure Aware)

Protocol	Traffic Type	Loss Detection and Notification	Loss Detection Node	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
RMST [Stann and Heidemann 2003]	Continuous	SNACK: H-by-H	Receiver	No	Simulation NS2	Delivery ratio, delivery overhead	Alone
POLY [Qureshi et al. 2011]	Periodic, event based	-	-	No	Simulation (Atarraya), modeling	Energy overhead, residual energy, network connectivity.	CDS-Rule K, Energy-efficient CDS (EECDS), A3
CTP [Gnawali et al. 2013]	Periodic, event based	ACK: H-by-H	Sender: H-by-H	No	Experimentation	End-to-end delivery ratio, duty cycle, efficiency, robustness	MultiHopLQI
R3E [Niu et al. 2014]	-	ACK: H-by-H	Sender: H-by-H	No	Simulation NS2	Packet delivery ratio, delivery latency, data transmission cost	AODV-ETX, REPF, GOR
Reliable-RPL [Ancillotti et al. 2014]	Periodic, event-based, query	-	-	No	Emulation COOJA, experimentation	Average packet delivery rate, packet loss ratio, end-to-end delay, energy consumption	RPL variants

through a fixed number of hop-by-hop link layer retransmission attempts. In the third step, the remaining lost packets are retransmitted, end to end, in order to check the integrity of the whole data. FLUSH uses end-to-end selective NACK to inform the source about the losses, allowing three packet holes of information. The rate control is done in the network in a manner that avoids congestion. For ensuring bandwidth maximum utilization, FLUSH tries to adapt to link interference dynamics so that each node sends in a rate that does not exceed the one of its successor, and only when being sure that its successor is free from interference. These two rules avoid losses due to buffer overflow. The drawback of FLUSH is its restriction to only one flow through the whole network, while multiple flows could coexist, especially if the shared part of the path is short. Also, any loss due to broken paths is not detected as FLUSH retransmits without taking into account loss cause.

RCRT (Rate-Controlled Reliable Transport) [Paek and Govindan 2007]: In this protocol, upstream packet reliability is ensured for loss-intolerant applications. The whole control is localized at the sink node, which demands end-to-end retransmissions through NACK packets in loss cases, and triggers a rate decrease if these losses take a long time to be recovered. RCRT interprets these losses as a congestion indication rather than link errors, which leads generally to only a few losses. Each source communication starts by the establishment of a connection with the sink in which the desired rate is mentioned. RCRT uses an AIMD rate scheme with a time-dependent multiplicative decrease, based on loss rate. The new rate is piggybacked in NACK or sent in a separate packet. RCRT gives different rates to sources according to their queries and application requirements. But the end-to-end used mechanism does not scale well with large WSNs, leading to high energy consumption, as the loss detection is made only at the sink. Also, sensors must keep packets until having sink feedback, which could lead to buffer overflow. In addition, the route-failure losses are not detected, which in reality needs to not trigger retransmission until the new route is established.

RBC (Reliable Bursty Converge Cast Protocol) [Zhang et al. 2007]: The aim of RBC is to ensure reliability for real-time upstream bursty packet forwarding toward the sink, where many packets are generated in a few moments at event detection. It is interested mainly in packet retransmission, by trying to schedule hop-by-hop node retransmissions in order to decrease channel contention. RBC gives more priority to nodes having more packets and packets having fewer retransmissions. Therefore, new packets are sent immediately to enable continuous packet forwarding. It uses the implicit block ACK and NACK behavior so that the new ACK or NACK contains information about the previous ones in order to overcome the ACK or NACK loss phenomena. It also allows out-of-order packet forwarding. RBC handles different timers for transmission triggering according to queue lengths. To avoid congestion, nodes piggyback their free queue size in packets, and the sender detecting this number below a threshold stops sending for a certain period. But implicit listening leads to overhearing that does not allow duty cycling use, which causes more energy consumption. Not overhearing packets forwarding could be due also to path failure, which is not handled and could lead to unnecessary retransmissions. Further, the fairness is not well ensured.

CTCP (Collaborative Transport Control Protocol) [Giancoli et al. 2008]: The aim of this protocol is to ensure packet reliability and avoid losses caused by congestion. It proposes two types of hop-by-hop reliability using ACK. In the first one, an ordinary single ACK is used each time the packet reaches its one-hop forwarder. In the second type, a double ACK mechanism is used in order to ensure more reliability. But the authors do not explain the overhead resulting from this double ACK use.

RTMC (Reliable Transport with Memory Consideration) [Zhou et al. 2008]: The goal of RTMC is to provide hop-by-hop retransmission and congestion control. It is inspired from the pipeflow concept, where the source has enough information and adapts to basestation and relay nodes' capacities. RTMC uses many control packets in order to monitor the communication. Each node has a small memory to store some packets in order to allow retransmissions. The authors do not give enough details on the congestion control mechanism.

CRRT (Congestion-Aware and Rate-Controlled Reliable Transport) [Alam and Hong 2009]: Its aim is to ensure upstream packet reliability through a double retransmission mechanism. The first one is through MAC reservation-based hop-by-hop retransmissions in which the sender asks the receiver for a retransmission, through special bits in the following packets, when it does not receive the ACK. The reservation mechanism tries to avoid losses caused by collisions between packet transmissions and retransmissions, and to decrease congestion in the case the receiver does not have enough buffer space. This part of reliability assurance is reduced a lot for the second end-to-end mechanism, where the sink asks for retransmissions through NACK packets. CRRT controls congestion through a hop-by-hop AIAD (additive increase additive decrease) rate control mechanism after hop-by-hop congestion notification. But it is the role of the sink to assign the source rate in order to offer fairness between nodes. Unfortunately, CRRT does not treat losses due to routing failure. In addition, its congestion control does not avoid collisions, and it lacks a scheduling mechanism that could bring a lot of benefit in such applications. In our previous work [Kafi et al. 2014b], a scheduling-based congestion control mechanism was proposed.

ERCTP (End-to-End Reliable and Congestion Aware Transport Protocol) [Sharif et al. 2010]: As its abbreviation notes, ERCTP ensures both congestion control and packet reliability. It allows assigning different rates to sources according to their packets' priority. The sink detects congestion due to intermediate nodes' buffer state in addition to end-to-end packet delay, and attributes rates (increase or decrease) regarding the actual congestion state. To offer reliability, ERCTP stores packets in intermediate nodes and the sink triggers retransmissions using NACK packets. In order to free intermediate node buffers, periodic ACK packets are sent back, or a time-based mechanism to delete packets could be used. But ERCTP lacks details on how to choose

Table II. Summary of Upstream Packet Reliability Based on Retransmission Protocols (Congestion/Interference Aware)

Protocol	Traffic Type	Loss Detection and Notification	Loss Detection Node	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
FLUSH [Kim et al. 2007]	Continuous	MAC: H-by-H, SNACK: E-to-E	Receiver	Yes	Experimentation	Overall throughput, transfer phase throughput	Different fixed sending rate
RCRT [Paek and Govindan 2007]	Continuous	NACK: E-to-E	Sink	Yes	Experimentation	Goodput, rate, packet reception	IFRC
RBC [Zhang et al. 2007]	Event based	IACK, INACK (by block): H-by-H	Sender (ACK), Receiver (NACK)	Yes	Experimentation	Event reliability, packet delivery delay, Event goodput, node reliability	Synchronous explicit ACK (SEA), Stop-and-wait implicit ACK (SWIA)
RTMC [Zhou et al. 2008]	Continuous	ACK: H-by-H	Receiver	Yes	Experimentation + Simulation	Transport time, packets transmitted	Synchronous explicit ACK (SEA
CRRT [Alam and Hong 2009]	Periodic, event based, query	ACK: H-by-H + NACK: E-to-E	Sender: H-by-H + Sink: E-to-E	Yes	Simulation NS2	Delivery ratio, energy efficiency, average end-to-end delay, end-to-end retransmissions	CSMA/CA
TASA [Palattella et al. 2013]	Periodic	ACK: H-by-H	Sender: H-by-H	Yes	Simulation (python)	Average single-hop delay, maximum achievable throughput, average network duty cycle, average power consumption	Alone
OTF [Palattella et al. 2016]	Periodic, event-based	ACK: H-by-H	Sender: H-by-H	Yes	Simulation (6TiSCH simulator), experimentation	End-to-end latency, end-to-end reliability	Alone

these storage nodes, as well as why it doesn't use hop-by-hop recovery as the packets are available in these intermediate storage nodes, rather than waiting for the sink to discover this loss. In addition, the hop-by-hop congestion control could be more helpful than the centralized sink-based one. The route failures are not invoked, which could cause losses also. ERCTP is not compared to any transport protocol conceived for WSNs and is only compared to TCP-based versions conceived for wireless networks, which have differences in WSN behaviors.

TASA (Traffic Aware Scheduling Algorithm) [Palattella et al. 2012, 2013] is a centralized schedule scheme based on TSCH (IETF proposition) behavior. The aim of TASA is to construct a tree-based schedule, at the sink level, that takes into account nodes' traffic load while ensuring as high a throughput as resources allow and using the advantage of frequency diversity. TASA uses matching and coloring in graph theory methods in order to achieve the mentioned goals. The result of the schedule is a number of slots that construct a frame where in each slot a number of nodes send or receive packets without any interference.

OTF (On-the-Fly Scheduling) [Palattella et al. 2016] presents a distributed schedule that ensures reliability through avoiding interferences by the use of slots. OTF adapts the slot number according to the network need of the nodes regarding the traffic load. The added or removed resources are sent to the sublayer 6Top (IETF proposition) for schedule adaptation.

In Table II, a summary of protocols belonging to the "upstream packet reliability based on retransmissions (congestion/interference aware)" subclass is highlighted.

A.1.2. Redundancy-Based Protocols. In this subclass of protocols, the recovery in loss case is ensured through the information already transmitted. This could be done through the transmission of multiple copies of the same information or through adding additional information to the original one. In this second case, this data is fragmented into several packets that could be received partially rather than in totality, but allowing its reconstruction at the receiver side, which is in charge of detecting the losses. In the following subsections, these protocols are highlighted.

A.1.2.a) Multiple Copies Based + Multipath Use Protocols. As explained previously, in this subclass, the protocols try to send in advance multiple copies of the same data on different paths regarding the known loss probability, in order to be sure of the correct reception of the data at the receiver side.

MVSA (Multipath Virtual Sink Architecture) [Seah and Tan 2006]: The authors propose a scheme where multiple virtual sinks are used in order to decrease contention appearing around the physical sink. It is specially conceived for harsh environments. Multiple copies of the same packet are used to ensure reliability rather than retransmission-based reliability. The protocol used multiple disjoint paths in parallel to increase reliability toward different sinks that are related through high-speed links. The number of paths depends on the channel quality and application requirements. This multitude of sinks and paths is for decreasing the contention that could result from the convergence of the paths toward the physical sink. A packet is successfully received if at least one virtual sink receives the packet.

A.1.2.b) Erasure Coding + Multipath Use Protocols. In WSN communications, loss of little bits in a packet is frequent, which is expensive to overcome through whole packet retransmissions. The aim of FEC is to allow these little losses and try to reconstruct the packet through the received bits. This is done by localizing the exact bit loss positions and reconstructing the original packet. Erasure codes are used for link and transport layer recovery using this redundancy-based policy. The use of redundancy information is beneficial due to the constrained memory of WSN nodes, as the added information for redundancy avoids caching data that wait for ACK and probably retransmissions. The most used code in the literature is Reed-Solomon [Reed and Solomon 1960], which uses a matrix for coding and decoding, named the Vandermonde matrix. The coding operation results in a an S vector, S=V*C, where V is the Vandermonde matrix and C is the original data vector. At the reception side, the original data vector C is constructed through the received vector R as that R=V*C. In the Vandermonde matrix, any set of rows is linearly independent [Srouji et al. 2011].

SPREAD (Secure Protocol for REliable dAta Delivery) [Lou et al. 2004]: The aim of SPREAD is to ensure security through splitting the original message into many shares using a secret sharing scheme and sending them toward the sink using multiple independent paths. The scheme is resistant to security attacks in the sense that even if many packets are compromised, the message could be reconstructed with the reception of at least a certain threshold packet at the destination. This scheme is also applicable for simple loss packets not related to security attacks but rather due to collisions and link loss. The share allocation is formulated as an optimization problem in order to minimize message compromise probability, while ensuring reliability through allowing the loss of certain packets. Redundancy is used to enhance the reliability but without affecting the security goal.

N-to-1 Multipath Routing Protocol [Lou 2005], H-SPREAD [Lou and Kwon 2006]: In the N-to-1 Multipath Routing Protocol [Lou 2005], the authors try to ensure transmission reliability through the construction of multiple node-disjoint paths toward the sink in order to be used for data transmission. The route construction is done by flooding in two stages. In the first one, the sink broadcasts a route-update message to construct a spanning tree, and nodes discover multiple paths to the sink. The goal of the second stage is to discover more disjoint paths. These disjoint paths are used by the nodes in order to split their traffic. The intermediate nodes use a per-hop packet salvaging technique to recover quickly from any link failures. This technique aims to choose another route at each hop at the moment of link failure. The authors claim the security of the protocol by the fact that the source splits the information into multiple shares that travel through multiple paths. So, the original message could not be reconstructed at any compromised intermediate node. On the other hand, the original message could be reconstructed at the basestation with the reception of any T packets from the N sent ones, allowing losses due to the lack of reliability (link and node failures) and node compromise (lack of security). In H-SPREAD (Hybrid-SPREAD) [Lou and Kwon 2006], the authors present an analytical framework for reliability and security analysis ensured by their scheme.

EQSR (Energy-Efficient and QoS-Based Multipath Routing Protocol) [Ben-Othman and Yahya 2010]: In this work, the authors try to ensure reliability, energy efficiency, and delay requirements using multiple paths. EQSR differentiates between real-time and non-real-time packets in order to give more priority to real-time packets through a queuing model. For reliability insurance, EQSR uses redundancy information in order to recover lost packets from different causes (interference, signal loss, node failure). The packets are transformed through an XOR-based FEC mechanism in order to allow their reassembly at the basestation, even if some losses will be present. EQSR fulfills its goals through different steps. In the first one, a broadcast message is sent in order to estimate, at each node, the cost of transmission through different neighbors. In the second step of EQSR, route discovery starting from the basestation is launched through the source nodes so that each node selects its preferred next-hop neighbor. This neighbor choice is based on different parameters, namely, residual energy, available buffer, and Signal-to-Noise Ratio (SNR). In addition to the primary path construction, additional paths are constructed to allow the selection of the set to be used in relation to the successful transmission probability through splitting the traffic between node-disjoint paths. In the same way, EQSR chooses the best paths with reduced delay propagation and dedicates them for real-time packets.

RDTS (Reliable Erasure-Coding Based Data Transfer Scheme) [Srouji et al. 2011]: The authors use hop-by-hop erasure coding in order to ensure reliability requirements. RDTS uses the partial coding mechanism in order to decrease computational overhead, by enabling the construction of original packets without decoding if these latter are correctly received. The number of redundant packets at each hop is computed according to the link quality. In a full coding process, at each hop, the node decodes the received packets to construct the original ones and after that performs the coding operation for the next-hop node. Systematic codes permit reconstructing the original packets without decoding, as the original fragments are contained in the received ones. The partial coding mechanism allows sending to the next-hop node a sufficient number of fragments, without reconstructing original ones. As the receiver receives m fragments, the next ones will be deleted to decrease delay.

In Table III, a summary of redundancy-based upstream packet reliability protocols is provided.

A.2. Downstream Packet Reliability Protocols. In this subclass of protocols, the packets could be control packets (like the rate to be used by source nodes), reprogramming packets (in order to change nodes codes), or query packets (in order to trigger event

Table III. Summary of Upstream Packet Reliability Based on Redundancy Protocols

Protocol	Traffic Type	Loss Recovery	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
MVSA [Seah and Tan 2006]	Continuous	Multiple copies: E-to-E	No	Simulation	Latency, transmission reliability, energy consumption	Single path (SP)
SPREAD [Lou et al. 2004]	Continuous	Redundancy: E-to-E	No	Simulation	Latency, transmission reliability, energy consumption	Single path (SP)
N-to-1 Multipath Routing [Lou 2005]	Continuous	Redundancy: E-to-E	No	Simulation	Reliability, security	Alone
H-SPREAD [Lou and Kwon 2006]	Continuous	Redundancy: E-to-E	No	Simulation	Latency, transmission reliability, energy consumption	Single path (SP)
EQSR [Ben-Othman and Yahya 2010]	Continuous	Redundancy: E-to-E	No	Simulation NS2	Energy consumption, delivery ratio, average delay	MCMP
RDTS [Srouji et al. 2011]	Continuous	Redundancy: H-by-H	No	Simulation NS2	Packet success rate, communication load overhead, energy consumption, network lifetime, load balancing	EEEC, no reliable transfer scheme

sending). These kinds of packets require a high level of reliability, because their loss could affect the application running and lead to its dysfunction. In order to ensure this high reliability exigency, retransmission-based recovery is better suited by the protocols rather than redundancy-based recovery to ensure individual packet reliability, rather than the roughly data reception where the information could be recovered generally through reading from different sensors. For these reasons, many mechanisms that accompany the sending process are provided. For example, the checksum verification of the whole packets at the end of the code transmission is recommended to find packet gaps. Also, the high number of retransmissions of each packet, different from the reduced ones in data packets, is used for reliability purposes. The slow transmission operation in order to avoid congestion from happening, in addition to the hop-by-hop recovery used mode, is also useful. Caching packets even after successful forwarding could also help for future loss recovery. All of these mechanisms are to be taken into account by the application developers to guarantee the applications' good functioning. In the following section, downstream packet reliability protocols are presented.

PSFQ (Pump Slowly, Fetch Quickly) [Wan et al. 2002, 2005]: The aim of this protocol is to ensure packet reliability in the downstream direction. This goal is mandatory, especially for reprogramming nodes and important control messages, and could also be in critical application queries. Every node has in its charge to redisseminate the packets toward its downstream neighbors in a slow manner, which will consequently avoid congestion. Every neighbor receiving these packets verifies their sequence numbers in order to detect any hole. In this gap presence, the receiver will directly send back a NACK packet containing the hole sequence number and momentarily stop any further forwarding for the purpose to halt the NACK that will certainly come from this forwarding. The sender that detects this NACK packet will quickly retransmit it in order to permit the normal dissemination operation. Every sender caches the forwarded packets, in general for no charge, as it needs them itself. In case this sender node no longer has this required packet, it will forward again through the reverse path the recovery demand, on the condition to listen many times to the same NACK to be sure that other neighbors do not respond if they have that packet. The normal pump operation, by the detector node, will resume after reception of the lost packet.

Finally, a report step to summarize the transport operation is envisaged. PSFQ takes into account the special loss cases like the first loss packet and the dissemination stop before reaching the envisaged file size. It proposes therefore combined timeout and NACK mechanisms to overcome these cases. PSFQ envisages a mechanism to reduce the broadcast overhead by avoiding the rebroadcast of a packet already heard many times, as it has a high chance to be already sent to the neighbors. As PSFQ does not offer any congestion control, it could not be used in the upstream direction, which needs the use of another protocol to ensure upstream packets' reliability and transport. This will pose the cohabitation problem between the two protocols, that the upstream is more used than the downstream communications. The most annoying problem is that the loss cause is not known, which could lead to sending different NACKs toward the sender without having the retransmitted packet, especially in the case of route failure and congestion overhead scenarios.

GARUDA [Park and Sivakumar 2003; Park et al. 2004, 2008]: This is designed to ensure downstream reliability when transporting message control, like queries and reprogramming code. For ensuring this goal, GARUDA is especially interested in the delivery of the first packet, which could be critical in single-packet-query scenarios. Therefore, it can benefit from the use of NACK packets for triggering retransmissions, as the NACK method cannot detect single packet loss. To this aim, GARUDA uses a pulse method, named Wait-for-First-Packet (WFP) pulse. The goal and characteristic of this impulse is to reach the nodes without interfering with any data transmission. The forwarding method of this pulse makes it reach the whole concerned nodes, which start themselves impulsing. The sink starts just after the transmission of the first packet. Each node receiving this packet stops pulsing and sends the packet again. For ensuring recovery optimality, GARUDA divides the network into core and no-core nodes. The second step in GARUDA functioning is the construction of core nodes, which is a virtual minimum dominating set (MDS) of the whole network, and which will have on its charge to ensure the reliability of the whole network. While the first packet is reliably forwarded, each node selects itself as a core node if it has not heard from an already elected core node, and being multiple three-hop-counts from the sink. The recovery of missed packets starts in two phases. In the first one, only the core nodes establish recovery. After that, the no-core nodes recover their missing packets from the core ones. Contrary to PSFQ [Wan et al. 2005], GARUDA allows the out-of-sequence packet forwarding, by informing the core node receiver of the missing packets in order not to send back any NACK until being sure of the packets' requested availability at the upstream core node. This is done by sending the A-map (Availability-map) to the next core node, which contains the list of missing packets. The recovery of core nodes is done at the same time of ordinary packet forwarding. The recovery of no-core nodes starts after the hearing of the A-map from the core node, indicating the reception of all packets. But GARUDA does not treat problems related to the originality of losses, namely, congestion and path failures. Also, the WFP pulse mechanism is energy consuming and the overhead of GARUDA mechanisms becomes high as the network size scales. The cohabitation of GARUDA with any upstream reliability mechanism is not studied.

FBcast [Kumar et al. 2006]: The authors propose a broadcast mechanism to use for node reprogramming and data dissemination in general. They use an erasure code redundancy mechanism in order to ensure reliability. Fbcast is evaluated against a probabilistic broadcast mechanism, and the authors claim the efficiency of Fbcast concerning links' overhead, collisions, and reliability. Each node that has to rebroadcast a packet (in a probabilistic way) encodes it to allow its decoding at the reception side. The rebroadcasting probability parameter is tuned in order to decrease collisions in

Table IV. Summary of Downstream Packet Reliability Protocols

Protocol	Traffic Type	Loss Detection and Notification	Loss Detection Node	Loss Recovery	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
PSFQ [Wan et al. 2005]	Continuous	NACK: H-by-H	Receiver	Retransmission: H-by-H	No	Simulation NS2 +testbed	Average delivery ratio, average latency, average delivery overhead	Scalable Reliable Multicast (SRM)
GARUDA [Park et al. 2008]	Continuous	NACK: H-by-H, two hops	Receiver	Retransmission: H-by-H	No	Simulation NS2	Latency, number of data sent, retransmission overhead, energy consumption	ACK scheme, in/out-of-sequence forwarding schemes+NACK
FBcast [Kumar et al. 2006]	Continuous	-	Receiver	Redundancy: H-by-H	No	Simulation Tossim	Reliability, transmission overhead, latency	Probabilistic broadcast

high-density networks. The authors use the Fountain codes class, which has the advantage of needing any k packets from the m sent, in order to reconstruct the original message. Fbcast ensures confidentiality by the fact that the encoding and decoding operations need a random seed and generator, which are shared between nodes. The authors propose two variants of Fbcast, where in the first one nodes are supposed to be one hop from the source. In the second variant, repeater nodes exist in the multi-hop network and are chosen based on a heuristic for their placement. But as stated in the beginning of this class, the use of redundancy coding could not ensure packet reliability in the case that the loss rate reaches a high threshold that does not allow the reconstruction of the original data.

In the Table IV, a summary of these aforementioned protocols is highlighted.

A.3. Bidirectional Packet Reliability Protocols. In this subclass of protocols, the aim is to ensure the delivery of each packet in two directions, namely, the upstream and downstream directions.

ATP (AdHoc Transport Protocol) [Sundaresan et al. 2005]: This uses the feedback taken from intermediate nodes for different purposes related to congestion control and avoidance. Namely, it uses initial rate feedback for startup rate estimation, and uses progressive rate feedback for congestion detection, congestion avoidance, and congestion control. The third feedback is used for path failure notification. The intermediate nodes calculate the feedback values and piggyback them in the forwarded packets toward the receiver, which at the reception moment responds to the sender by giving these feedback values. The packets' reliability is ensured through end-to-end SACK packets sent by the receiver. ATP uses 20 SACK blocks in its reliability feedback. The priority at the sender side is given to the packets to be retransmitted before the new data packets. The main drawback of ATP is its end-to-end mechanism, which leads to a high delay as the network scales. This delay concerns the rate control and the path change that lead to packet loss and energy consumption.

DTC (Distributed TCP Cache) [Dunkels et al. 2004]: The authors try to resolve the problems of TCP and make it usable in WSN networks in order to benefit from the IP communication with other networks and infrastructures. For that, many mechanisms were proposed to overcome TCP problems in WSN applications. A spatial IP assignment was proposed, which is based on the node location, so that nodes of the same IP subnet do not need to transmit the full IP address in the packets' header. In this situation, a header compression is used and shared between these nodes. The second mechanism is Distributed TCP Caching, which consists of cache packets in the network to allow local retransmissions, rather than invoking end-to-end packet retransmissions. DTC uses

Table V. Summary of Bidirectional Packet Reliability Protocols

Protocol	Traffic Type	Loss Detection and Notification	Loss Detection Node	Loss Recovery	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
TCP-DTC [Dunkels et al. 2004]	Continuous	SACK: H-by-H	Receiver	Retransmission: H-by-H	Yes	Simulation Omnet++	Number of packets sent	Alone
ATP [Sundaresan et al. 2005]	Continuous	SACK: E-to-E	Sink	Retransmission: E-to-E	Yes	Simulation NS2	Throughput, fairness	TCP default, TCP ELFN
TSS [Braun et al. 2007]	Continuous	IACK: H-by-H	Receiver	Retransmission: H-by-H	Yes	Simulation Omnet++	Throughput, number of packet transmissions	TCP

timeouts and SACK to ensure packet loss detection based on RTT (round-trip time) values. The authors combine the use of UDP for packets that do not need reliability and TCP for those needing reliability. But the well-known congestion control problem related to TCP in wireless environments is still present in DTC. Also, the spatial IP addressing does not work in mobility scenarios. DTC does not detect route failure loss problems either.

TSS (TCP Support for Sensor Nodes) [Braun et al. 2007]: This is an extension to the DTC [Dunkels et al. 2004] caching mechanism for allowing hop-by-hop retransmission in loss cases. TSS could be seen as an intermediate layer between TCP and the network layer. It differs from DTC as it does not delete packets in cache after a timeout but only after acknowledgment. TSS does not forward a new packet until hearing that the previous one is correctly transmitted by its forwarding node, which plays a double role of congestion control and implicit acknowledgment for a reliability mechanism. In the case of non-ACK reception (not overhearing forwarded packet), a time-out mechanism is used carefully for triggering retransmission. But the assumption that each buffer holds only one packet is not logical, as a node could have many children that send it their packets that must be held by this parent. Also, TSS could suffer from overhearing IACK energy consumption, especially if the forwarder has many children. Even though the congestion control mechanism used by TSS avoids buffer overflow losses, it is so restrictive as previous nodes are not allowed to transmit until the next ones receive their ACKs.

In Table V, a summary of the previous subclass of protocols is presented.

B. Event Reliability Protocols. In this class of protocols, only the upstream direction is concerned because the downstream kinds of packets do not allow losses. The event reliability is interested in the delivery of the event rather than each packet. This could be ensured using a redundancy-based (similar copies) mechanism only, or enhanced with a retransmission mechanism. The aim of event-based reliability protocols is to preserve the energy resources by accepting just the sufficient information. Also, delay-sensitive applications could be the cause for an event-based choice in order to reduce the network load that could result from the individual packet reliability. In the following sections, a deeper view of the upstream event-based protocols will be presented, where the protocols are classified regarding their awareness of congestion and/or routing failure. In addition, multipath or single-path at each time is used for these classifications. Also, the delay bounds and energy consumption minimization parameters are used in our classifications. In Figure 7, a possible classification of event-based protocols is highlighted.

B.1. Redundancy-Based Upstream Event Reliability Protocols. As explained previously, in this subclass of protocols, the upstream event reliability is ensured through sending multiple copies of the same information by the same node through a given

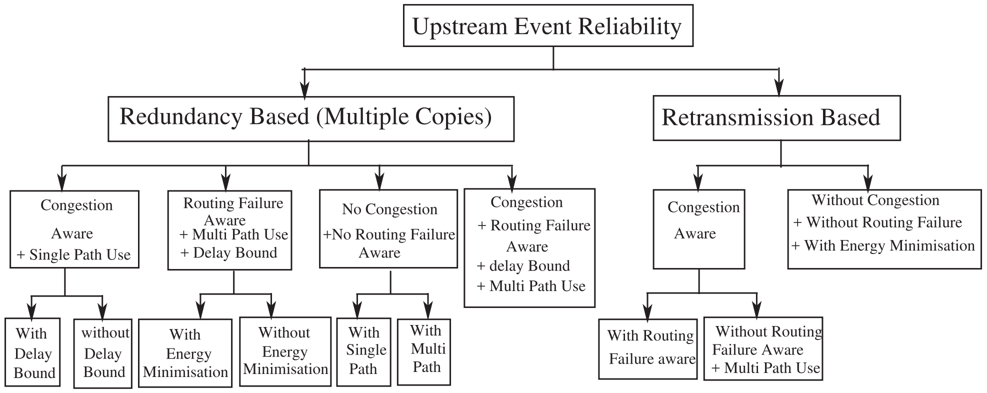
Upstream Event Reliability
Redundancy Based (Multiple Copies)
Retransmission Based
Congestion Aware + Single Path Use
Routing Failure Aware + Multi Path Use + Delay Bound
No Congestion +No Routing Failure Aware
Congestion + Routing Failure Aware + delay Bound + Multi Path Use
Congestion Aware
Without Congestion + Without Routing Failure + With Energy Minimisation
With Delay Bound
without Delay Bound
With Energy Minimisation
Without Energy Minimisation
With Single Path
With Multi Path
With Routing Failure aware
Without Routing Failure Aware + Multi Path Use

Fig. 7. Upstream event reliability classes.

rate, or by a number of nodes in order to sense adequately a happening event. The classification of this class of protocols is done regarding the awareness of the loss cause, namely, congestion and routing failure causes.

B.1.1. Congestion-Aware Protocols. In this subclass, the described protocols are aware of congestion happening and try to react through the rate decrease. We have stated that they all are based on a single-path use and are not aware of any routing failure happening.

B.1.1.a) Without Delay Warranty. In this subclass of protocols, the delay is not taken into account and no priority between nodes packets is performed.

ESRT (Event to Sink Reliable Transport Protocol) [Sankarasubramaniam et al. 2003]: The goal of ESRT is to ensure reliable event detection by allowing some losses, in order to decrease the energy consumption due to packet reliability and the congestion that could be caused for that purpose. The sink defines the minimum number of packets to be received at a defined epoch in order to consider that the event is correctly detected, in relation to application requirements. In ESRT, sensors change their sending rate using the sink feedback regarding the reliability level or congestion detection. Every node sets a CN congestion notification bit in the packets as soon as its buffer reaches a threshold. The sink periodically computes a new reporting rate for all sources based on a reliability measurement, the received CN, and the old reporting rate. It broadcasts this new rate with a high-powered amplification. ESRT running includes five states: No Congestion Low Reliability (NCLR), No Congestion High Reliability (NCHR), Congestion High Reliability (CHR), Congestion Low Reliability (CLR), and Optimal Operating Region (OOR). In NCLR, the reporting rate is increased to reach an acceptable reliability. In both NCHR and CHR, the reporting rate is decreased in order to save energy as the event is correctly captured and to avoid the congestion in the CHR case. In CLR, the reporting rate decrease is sharper than the two previous states because the lack of reliability is due to the congestion losses. In OOR, the reporting rate is unchanged for the next decision interval, as the reliability is already ensured and the congestion is avoided. Among the drawbacks of ESRT is that treating differently characterized regions (event priority, node density) in the same way will decrease the system throughput. Moreover, ESRT does not give attention to interference causes. In addition, ESRT regulates the rate for the next epochs but does not recover any packets even if the reliability is below the required level. This is worse because the required reliability level could take a long time to be reached through the used regulation method.

Also, as the decision is taken at the sink, the end-to-end negative delay properties are present. The ESRT regulation method is well suited for continuous applications, as it is conceived in this view, where rate regulation has a meaning for the next epochs. But this method is worse suited for bursty event-based detection application, where the event spontaneously happens in random locations [Shaikh et al. 2008]. The path losses are not invoked in the ESRT strategy, as only congestion-based losses are treated, which could decrease application reliability in some path failure cases.

E2SRT (Enhanced Event-to-Sink Reliable Transport) [Kumar et al. 2009]: This can be considered as an improvement to the well-known ESRT [Sankarasubramaniam et al. 2003] protocol. E2SRT treats the case when the desired reliability level by the sink could not be reached by the network capacity, resulting to an "overdemanding" case. This situation leads to a fluctuation in the rate allocated to the nodes by the sink between "congestion-low-reliability" and "no-congestion-low-reliability" regions, without reaching the optimal point. E2SRT discovers this case by its algorithm and converges to the maximum operating region (MOR) rather than the nonreachable optimum operating region (OOR). In the case where the network capacity allows reaching the desired reliability level, E2SRT has similar functioning to ESRT. But E2SRT lacks a retransmission scheme, where the packets are lost even with the applied strategy, which could have negative consequences on application fidelity.

LTRES (Loss-Tolerant Reliable Event Sensing Protocol) [Xue et al. 2009]: This is designed for event-based reliability applications, where individual sensor losses are permitted. The sink determines the event-sensing frequency regarding the application requirements. LTRES handles congestion control by using the packet loss threshold as a congestion indication in order to reduce control overhead and regulate source rates. It uses also a distributed steady-state throughput estimation approach to ensure fair rate control with maximum bandwidth utilization and fast convergence time toward the path, when the application requirements could not be fulfilled by network conditions. LTRES named the event-area sensors as Enodes, which work in two stages. In the first one, they try to fulfill application requirements through the sink frequency. In the second stage, Enodes provide best-effort service using steady-state throughput estimation. But LTRES does not ensure any packet recovery in case of lack of reliability. Also, any packet loss due to route failure is treated.

B.1.1.b) With Delay Warranty. The aim of this subclass is to give more priority to packets that must arrive on time for ensuring application requirements, because its nondelivery at times could lead to dysfunctional problems.

DST (Delay Sensitive Transport) [Gungor and Akan 2006]: This is an extension of ESRT and targets critical delay event applications, where the sink must receive at the time the event is happening. The event delay is the time between the event detection and the sink notification. In DST, the event reliability is treated as in ESRT, with the notion of delay-sensitive application as a new requirement. DST uses a Time Critical Event First (TCEF) scheduling with prioritized MAC to ensure delay bounds. It measures the elapsed time to update the remaining time to the deadline at each node and piggybacks it in event packets. With decreasing values, the packets get higher priority. DST detects congestion using average node packet delay and buffer level. Average node delay measures the contention around the node, which varies depending on the used rate and channel load. A congested node, having delay or buffer values higher than a threshold, informs the sink using the notification (CN) bit in packet headers. Using a reliability indicator and current network condition, the sink adjusts sensors' reporting frequency, as in ESRT [Sankarasubramaniam et al. 2003]. Neither DST nor ESRT tries to avoid collision-based congestion; they just decrease the source rates. No details for deadline attribution and TCEF scheduling have been given. The same remarks as ESRT are still available in DST, concerning the lack of recovery in the case that the reliability is under the suitable level. Also, apart from mentioning that DST works with routing protocols, DST does not take into account lack of reliability due to path failure, which does not necessitate any frequency increasing, but rather necessitates stopping sending until a new path finding or swapping to a pre-established backup path.

B.1.2. Routing Failure Aware Protocols. In this subclass of protocols, broken paths are detected by the protocols in order to avoid any recovery that will be useless. In addition, we have stated that in this subclass, multipath is used for ensuring the event reliability. Also, delay sensitivity is taken into account by the surveyed protocols. On the other hand, energy minimization is envisaged by ECMP but not by MCMP.

MCMP (MultiConstrained QoS Multipath Routing Protocol) [Huang and Fang 2008]: In this protocol, the authors are interested in two QoS challenges, which are delay and reliability. For that, a multipath strategy is used in order to answer application needs. For the delay constraint, the protocol avoids paths with a delay higher than tolerated by the application. For reliability, many paths are combined and used in parallel (multiple copies) for increasing the envisaged reliability. In MCMP, QoS is ensured probabilistically through formulating the end-to-end QoS problem. It uses probabilistic modeling and linear programming to solve the routing challenge, which is based on a hop-by-hop manner. Also, the MCMP routing algorithm takes into account energy by choosing the minimum set of paths when offering the required reliability. Each node chooses its next forwarder based on link quality.

ECMP (Energy Constrained Multipath Routing Protocol) [Bagula and Mazandu 2008]: In this work, the authors extend MCMP by taking into account energy minimization at the moment of path choice, in addition to the QoS requirements taken by MCMP, namely, delay and reliability. The authors claim resolving energy minimization as an optimization problem with reliability, delay, and geo-spatial path selection constraints. The authors transform the path-based model into a link-based model using probabilistic approximations, in order to affine energy consumption minimization. This is done by choosing the next-hop node related to energy minimization rather than randomly. So, ECMP could choose longer paths if these ones lead to minimum energy while also satisfying also the other QoS constraints.

B.1.3. Congestion + Routing-Failure-Aware Protocols. In this subclass, the protocols take into account the congestion control in addition to being aware of routing failure happening. They use multiple paths in order to recover from any broken path. We have stated that they take a delay warranty in the transmissions also.

MMSPEED (Multi-Path Multi-SPEED Protocol) [Felemban et al. 2006]: In this protocol, the authors provide a mechanism to deliver packets while ensuring QoS service differentiation levels for timeliness and reliability requirements. MMSPEED proposes different packet speed options to be applied for different packet types in order to reach deadline requirements. For reliability differentiation requirements, MMSPEED proposes probabilistic multipath forwarding in order to choose the number of parallel paths (redundancy) regarding the application requirements. Each node chooses the number of neighbors to use (which represents also the number of copies) according to the link error and geographic distance from the destination. MMSPEED takes local decisions and uses the compensate method when the packet travels progressively to correct errors resulting from its local view. The goal of this local view is to guarantee scalability, network dynamics, and urgent data transmission. MMSPEED uses longer paths in order to balance traffic and ensure congestion avoidance. In addition, it proposes MAC changes in order to support prioritized access for high-speed traffic and to support reliable multicast delivery. The authors themselves claim that MMSPEED is energy consuming in order to ensure its functionalities.

TREND (Timely, Reliable, Energy-efficient and Dynamic WSN Protocol) [Marco et al. 2010] is a cross-layer protocol that aims to reduce energy consumption while ensuring an acceptable level of reliability and latency. The protocol is conceived for WSN-based industrial control applications. An optimization problem is formulated to answer the energy consumption minimization. TREND adapts dynamically to controllers' requirements in terms of reliability. It is based on duty cycle to enhance energy consumption through the use of TDMA/CSMA-based communications. TREND proposes a routing algorithm composed of static and dynamic parts, while the static one concerns intercluster communications that use TDMA-based MAC, and the dynamic one concerns nodes in the same cluster that send packets to the next hop in a random manner. TREND uses aggregation in order to minimize energy and traffic load.

B.1.4. No Congestion + No Routing Failure Awareness Protocols. In this category, the protocols do not treat the loss causes that could be due to congestion happening or path failures.

B.1.4.a) With Single-Path Use. In this subclass, a single path is used at a time in addition to not being aware of the loss cause.

QoS Control Protocol [Iyer and Kleinrock 2003]: The authors define the QoS as the resolution of sensing in a WSN. This is done by enabling a number of sensors to send information about the happening event, regarding the reliability requirements of the application. So, many losses are tolerated based on the redundancy of the sensed event in a field by the sensors. The sink broadcasts the desired redundancy level as a sending probability regarding the previously received packets. But this value changes during the network lifetime, as the sensors' energy is exhausted. The authors use a model, named the Gure game, where each node is a player and votes positively or negatively. The whole vote resulting at the sink evolves the node enabling probability. But no explicit congestion control is presented, which could cause packet loss and decrease the application fidelity. Also, no explicit recovery mechanism is envisaged.

B.1.4.b) With Multipath Use. The protocols of this subclass use multipath in their communication, even though they do not take into account routing failure losses and congestion-based ones.

REINFORM (REliable INformation FORwarding Using Multiple paths) [Deb et al. 2003]: Its aim is to send multiple copies of the same packet toward the sink using different random routes. The packet duplication could be done in an end-to-end or hop-by-hop manner. The number of duplicated packets is in relation to the link error probability, the hop count of the node from the sink, and application event reliability requirements. The duplicator node decides the forwarder node and the duplicate number of copies. The forwarding choice is done in a random mechanism to allow energy depletion and network load balancing toward multiple disjoint-path forwarding. But REINFORM does not take any congestion control mechanism into account, which could be a loss cause. Also, even with multiple copies of the same packet, there is no guarantee that the packet will reliably reach the sink, which could have negative consequences on application.

DTRP (Directed Transmission Routing Protocol) [Nassr et al. 2007]: This protocol uses proactive multipath in order to offer reliability and scalability for WSNs. The protocol allows one to parameterize the reliability level, as a probability, through the

Table VI. Summary of Upstream Event Reliability Protocols Based on Multiple Copies

Protocol	Traffic Type	Control Node	Copies Generation	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
ESRT [Sankarasubramaniam et al. 2003]	Continuous	Sink	E-to-E	Yes	Simulation (NS2) +analytically	Normalized reliability, average power consumption	Alone
E2SRT [Kumar et al. 2009]	Continuous	Sink	E-to-E	Yes	Simulation +analytically	Throughput, latency, loss rate, normalized reliability	ESRT
LTRES [Xue et al. 2009]	Continuous	Sink	E-to-E	Yes	Simulation GloMoSim	Convergence time, average packet loss rate, and bandwidth utilization	ESRT
DST [Gungor and Akan 2006]	Continuous	Sink	E-to-E	Yes	Simulation NS	Convergence time, energy consumption	ESRT
MCMP [Huang and Fang 2008]	Continuous	Sink	E-to-E	No	Simulation PARSEC	On-time packet delivery ratio, packet delivery ratio, expiration ratio, average packet delay.	Single-path routing, braided multipath routing
ECMP [Bagula and Mazandu 2008]	Continuous	Receiver	H-by-H	No	Simulation	Average energy consumption, delivery ratio, average data delivery delay	Single path (SP), MCMP, LDPR (Link-Disjoint Paths Routing)
QoS Control [Iyer and Kleinrock 2003]	Query based	Sink	E-to-E	No	Simulation	Packets received	Alone
REINFORM [Deb et al. 2003]	Event driven	-	H-by-H / E-to-E	No	Simulation	Attained reliability, overhead	Flooding, single-path forwarding
DTRP [Nassr et al. 2007]	Continuous	Receiver	H-by-H	No	Experimentation	Packet delivery ratio (PDR), total network load	Gossiping, MINTRoute
MMSPEED [Felemban et al. 2006]	Continuous	Receiver	H-by-H	No	Simulation J-SIM	Average delay, Reaching probability, overhead	SPEED
TREND [Marco et al. 2010]	Continuous	-	-	No	Experimentation	Reliability, latency, duty cycle, load balancing	SERAN, IEEE 802.15.4

redundancy ensured at the source node regarding the application requirements. Each concerned node takes a flood decision according to the required reliability probability.

In Table VI, a summary of protocols belonging to upstream event reliability based on multiple copies is presented.

B.2. Redundancy + Retransmission-Based Upstream Event Reliability Protocols. In this subclass of protocols, the event reliability is ensured not only by the multiple copies sent from the beginning but also through a number of retransmissions in case of loss happening. As used in the previous redundancy-based event-based protocol classification, in this subclass of protocols, the classification is also based on the awareness of loss, which is loss due to congestion or routing failure.

B.2.1. No Congestion + No Routing Failure Awareness Protocols. As its nomination states, in this subclass of protocols, the loss cause is not treated. But on the other hand, we have stated that the protocols belonging to this subclass try to decrease energy consumption.

AReIT [Shaikh et al. 2009]: Its aim is to offer adaptive reliability according to the application requirement and network conditions. For that purpose, AreIT uses spatial and temporal redundancy. Spatial redundancy means multiple copies of the same data handled by different sensors, while temporal redundancy means the retransmission of the same data by the same sensors in case of loss. Offering this adaptive reliability allows for the saving of energy if the information maps with the application need and no more reliability is required. The fact of being aware of the application need allows one to adapt to the network condition perturbations also. AreIT is based on the RBC [Zhang et al. 2007] protocol in order to offer flexible reliability rather than the packet reliability handled by RBC. AreIT changes the number of sending nodes and the number of copies according to the network conditions and application requirements. It uses implicit ACK in order to reduce the overhead that could result from the explicit ACK use. It also uses hop-by-hop caching. BEP (bit error probability) is employed to know about the links' qualities before deciding the number of senders and the number of retransmissions.

ERTP (Energy-Efficient and Reliable Transport Protocol) [Le et al. 2009]: This is specially designed for data streaming along WSNs. ERTP's goal is to ensure energy consumption minimization and reliability requirements, and it does not focus on latency issues. The reliability concerns the whole event and not the individual packets. Its threshold is defined according to the application need and loss rate. This reliability level is transformed to a retransmission threshold in a hop-by-hop manner, where each node hears its forwarder resulting on implicit ACK. ERTP implements a timeout mechanism to trigger efficient retransmissions, rather than a fixed timeout retransmission that could lead to inefficient retransmissions. The calculation of the retransmission number in an adequate manner leads to a high level of reliability and low energy consumption regarding application requirements. But ERTP supposes low rate transmissions and does not present any congestion control mechanism in order to avoid congestion-based losses, which could lead to energy wastage in some cases. Also, hearing neighbor transmissions could be cost inefficient as it does not allow duty cycling.

ERP (Event Reliability Protocol) [Mahmood and Seah 2011]: The authors focus on upstream event reliability and try to minimize network overhead and energy consumption by deleting redundant information packets based on spatiotemporal correlation. ERP uses hop-by-hop implicit acknowledgment in order to detect losses and ensure a region-based selective retransmission mechanism. This retransmission is done via controlling that any other packet from the same region was not present in the node queue, in order to avoid unnecessary data packet retransmission. Doing this control in a hop-by-hop manner avoids overhead of sink end-to-end control, which could lead to congestion and energy consumption. But as with any implicit acknowledgment mechanism, ERP could suffer from long hearing periods in order to detect eventual losses, which could be energy consuming too. Despite that, ERP minimizes congestion by the elimination of these retransmissions but does not present any explicit congestion control mechanism. Also, the redundancy control mechanism presented at the sender part could be used at the forwarder part in order to avoid retransmissions when the packet is forwarded from nodes other than the sender. This is in order to exploit the overhearing of the forwarder that is already used by the authors to ensure IACK.

B.2.2. Congestion-Aware Protocols. The protocols that we have stated in this subclass are aware of congestion happening and try to control it through rate decreasing. In addition, the protocol could be aware of the routing failure or not.

B.2.2.a) Routing-Failure-Aware Protocols. In this subclass, the protocols take into account the causes of loss in order to avoid the next retransmissions by resolving this loss cause.

PORT (Price-Oriented Reliable Transport Protocol) [Zhou et al. 2005] is interested in the added information that a node can bring to that already taken by the sink, in order to avoid usefulness transmissions leading to overhead while ensuring application fidelity. PORT attributes a price to each node, which concerns the total number of transmission attempts from the node to the sink until a successful reception. This is in order to avoid links with high communication overhead and leading to high energy consumption. This price increases with the congestion happening. After the reception of this information from the nodes, namely, transmission attempts and congestion level, the sink assigns to each node a reporting rate that takes into account its price and its importance to enhance application fidelity, while trying to decrease congestion and energy consumption as much as possible. This is different from ESRT [Sankarasubramaniam et al. 2003], which gives to all sources the same rate. In PORT, nodes choose dynamically their forwarding nodes based on congestion level and loss rate for the purpose of selecting the appropriate one that decreases this phenomenon. Route failures are taken into account by the nodes at the forwarding moment in order to avoid broken paths. The detector node makes the price value infinite to let other nodes avoid this wrong path. But PORT does not give enough detail about the retransmission attempt system in case of packet loss. Also, sending end-to-end control information to every node could be difficult to achieve in multihop networks. In addition, dynamic maintenance of a list of neighbors, at every node, with continuous updating of loss rates has a significant memory footprint and communication overhead.

B.2.2.b) No Routing Failure Awareness Protocols. In this category, multiple paths are used to fulfill event reliability, even not being aware of the routing failure loss cause.

TRCCIT (Tunable Reliability with Congestion Control for Information Transport) [Shaikh et al. 2010]: In this protocol, in addition to the basic idea of AreIT [Shaikh et al. 2009], the hybrid acknowledgment (HACK) mechanism is used in order to ensure the desired reliability level. A hop-by-hop control is established to permit the sender overhearing the retransmission of its packet by the forwarder, which constitutes the IACK reliability part. In the case where the receiver decides to suppress the received packet, as the desired reliability level is already reached, it transmits a simple ACK to the sender for the purpose of avoiding its packet retransmission after the timeout expiration. TRCCIT tries to control congestion by using multiple path forwarding at the moment of congestion detection. But using multiple paths is not always possible, and therefore not sufficient as a congestion control mechanism.

In Table VII, a summary of the retransmission-based upstream event reliability protocols is highlighted.

C. Event + Packet Reliability Protocols. In this subclass of protocols, event and/or packet reliability is ensured regarding the application requirement, data kinds, and/or node types. We have stated that this subclass is further divided into an upstream protocol class and bidirectional protocol class. In the following sections, a deeper view on protocols is performed.

C.1. Upstream Protocols. The aim of this subclass is to ensure the event and/or packet reliability in the upstream direction. We can find protocols that ensure this reliability regarding the node types, like RT2, or application requirements, like STCP, RDFA, and DTSN. In addition, we can further classify the protocols regarding their awareness of the loss cause, which could be congestion and/or route failure happening that require adequate control before triggering any recovery. In Figure 8, a possible classification of this subclass of protocols is depicted.

Table VII. Summary of Upstream Event Reliability Protocols Based on Multiple Copies and Retransmissions

Protocol	Traffic Type	Loss Detection and Notification	Loss Detection Node	Loss Recovery	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
PORT [Zhou et al. 2005]	Continuous	H-by-H resource control + E-to-E traffic control	Receiver	Retransmission	Yes	Simulation NS2	Energy consumption	Directed diffusion + ESRT
ERTP [Le et al. 2009]	Continuous	IACK: H-by-H	Sender	Retransmission	No	Simulation NS2 + experimentation tinyos	Delivery ratio, energy consumption, average packet delay	Simulation: Jacobson algorithm, RTO=100*RTT, RTO=10*RTT. Experimentation: Surge
AReIT [Shaikh et al. 2009]	Continuous	Hybrid ACK (IACK or ACK): H-by-H	Sender	Probabilistic adaptive retransmissions	Yes	Simulation Tossim	Information transport reliability, timeliness, efficiency	TRCCIT (with no congestion control), RBC
TRCCIT [Shaikh et al. 2010]	Continuous	Hybrid ACK (IACK or ACK): H-by-H	Sender	Probabilistic adaptive retransmissions	Yes	Simulation Tossim	Information transport reliability, timeliness, efficiency	TRCCIT (with no congestion control), modified MMSPEED, RBC
ERP [Mahmood and Seah 2011]	Continuous	IACK: H-by-H	Sender	Conditional retransmission	No	Simulation GloMoSim	Number of event detections reported, number of events reported with duplicates, average number of transmissions per node	Stop-and-wait with implicit acknowledgment (SWIA) from DFRF, no-retransmissions

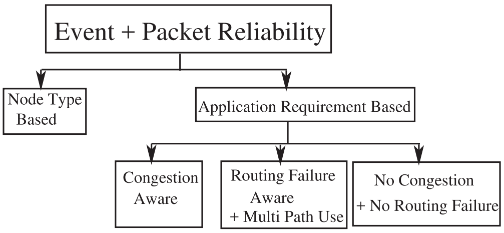
Event + Packet Reliability
Node Type Based
Application Requirement Based
Congestion Aware
Routing Failure Aware + Multi Path Use
No Congestion + No Routing Failure

Fig. 8. Upstream event + packet reliability classes.

C.1.a) Node-Type-Based Upstream Event/Packet Reliability Protocols. In this subclass, the choice of event or packet upstream reliability is established regarding the node type, which indicates the necessity of such decisions. We have surveyed in this subclass the RT2 protocol, which ensures at the same time congestion and routing failure awareness, in addition to delay boundaries of critical data packets.

RT2 (Real Time and Reliable Transport) [Gungor et al. 2008]: It treats the problem of event reliability in wireless sensor and actuator networks (WSANs). In WSANs, the sensors perform the environment sensing task, while the actuators make decisions and perform actions on the environment. RT2 could be seen as an extension of the DST [Gungor and Akan 2006] protocol, which is itself an extension of ESRT [Sankarasubramaniam et al. 2003]. The aim of ESRT is to ensure event reliability while avoiding congestion and unnecessary energy consumption. DST came to introduce the delay-sensitive paradigm to enhance ESRT functionality while keeping the event reliability one. RT2 divides the WSN network into sensor and actuator parts. The sensor part of the network is characterized by its dense deployment nature, which justifies the authors' choice of using event-based reliability to forward the packets toward the actuator nodes. From a resemblance point of view, the actuator node has the equivalence of sink in DST and ESRT protocols in its subnetwork responsibility. The reliability assurance in this subnetwork part is exactly the same as explained in the DST protocol. So, RT2 ensures both reliability and delay delivery, taking into account the avoidance of congestion, which is the cause of these losses, and lack of reliability. The actuator node sends periodically to the sensor nodes the frequency to use in order to get the delay reliability level while avoiding congestion. When the data comes to the actuator node, the event reliability paradigm is no longer available, and ordinary packet reliability is used in order to ensure collaboration and appropriate actions among actuators. So, the need for the SACK (selective ACK) mechanism to recover packet losses between the actuator nodes is mandatory. In addition to the recovery part, the congestion control in this part of actuator-actuator communication is treated through the rate adjustment mechanism similar to that used in the sensor-actuator part, but without leading to any packet loss. Also, RT2 works with any routing protocol and uses cross-layer information. RT2 is aware of route failure and applies momentary stop sending until the re-establishment of the new route, rather than wrong congestion control, by the use of probe packets to ensure the re-establishment of a new route. Despite that, RT2 shows different advantages but the same drawbacks of recovery present in DST and ESRT. Namely, it lacks a recovery mechanism in the case of insufficient event reliability in the previous epochs, waiting for the frequency equilibrium in the next epochs. As the communication in the actuator-actuator part is done in an ad hoc manner, the authors do not give details on sink communication, and unfairness between actuators could result. If the actuator part is large, RT2 will suffer from end-to-end recovery also.

DRINA (Data Routing for In-Network Aggregation) [Villas et al. 2013] is a routing protocol for in-network data aggregation. DRINA ensures high reliability by handling routing failure through changing the next-hop nodes and retransmitting the lost packets that are not acknowledged after a timeout period. DRINA minimizes messages exchange while ensuring tree routing. DRINA is a cluster-based aggregation protocol where the nodes detecting the event send the packets to the coordinator node that aggregates the reading before transmitting the resulting packet to the next-hop node toward the sink using the possible short-path construction.

RARE (Robust Asynchronous Resource Estimation) [Zhao et al. 2013] is a MAC layer framework that ensures reliable communications in a hybrid network of a densely large number of transmit-only and standard nodes by constructing a single-hop cluster architecture that minimizes energy consumption by the duty cycling approach. Transmissions of standard nodes are scheduled to avoid collisions. Each transmit-only node transmits for a defined number of times to ensure a given reliability level, whereas standard nodes retransmit in the case of not receiving ACK from the sink. Between RARE drawbacks is the transmit-only node number of transmissions that need to be defined before the deployment. Also, as the authors claimed, clock drifting should be handled to enhance the transmit-only nodes' performance. Multihop architecture should be constructed to scale the use.

C.1.b) Application-Requirement-Based Upstream Event/Packet Reliability Protocols. In this subclass, the protocols allow the choice of event or packet reliability regarding the application requirement. In addition, we can further classify the protocols according to their awareness of congestion or routing failure happening. For example, STCP is aware of losses due to congestion losses and triggers the rate reduction at its happening.

RDFA is aware of routing failure and uses multipath in order to overcome path failures. On the other hand, DTSN is not aware of any of the previous losses and triggers recovery without any control.

STCP (Sensor Transmission Control Protocol) [Iyer et al. 2005]: It tries to ensure a flexible reliability depending on the application requirement, namely, allowing packet reliability or event reliability. The application types taken into account by STCP could be continuous or event based. If the continuous application is envisaged, STCP offers NACK-based retransmissions at the sink node in order to ensure reliability. In this case, it uses the notion of window size to trigger the NACK if the packets are not received in a timeout period with a threshold value. This is done in order to allow packets loss recovery according to application requirements. If event-based application is targeted, where the event is generally in one packet, ACK-based retransmission is used at the source side. STCP ensures congestion control in the case of continuous application, to avoid resulting losses, by observing the buffer state and piggybacking the information in data packets toward the sink. On the other end, the sink will inform the source by piggybacking again in NACK or ACK packets. In the two cases, STCP suffers from its end-to-end mechanism. Also, STCP suggests the clock synchronization between sensor nodes, which could result in a high overhead, especially with the large-size network. STCP is not aware of the loss cause, which could lead to waste of retransmitted packets in case of path failure rather than congestion.

DTSN (Distributed Transport for Sensor Networks) [Marchi and Nunes 2007]: The aim of this protocol is to offer total and differentiated upstream reliability for WSNs. In full reliability mode, all intermediate nodes cache packets in order to allow retransmissions that are triggered by the source. This last sources sends an EAR (Explicit Acknowledgment Request) toward the sink after sending a block of packets. The sink answers by an ACK if all packets are correctly received, or by a NACK containing gaps in sequence numbers in a loss case. This NACK when traversing the reverse path could be recovered in its totality or partially by caching nodes, which change the NACK content according to their contributions. In the differentiated reliability mode, only the core part of the information is transmitted with a total reliability, needing so to be buffered at the source until receiving its correct transmission. On the other hand, the additional data part will be beneficial for more details in the application (e.g., more resolution in an image) but will not be transmitted with the same reliability requirements in order to decrease energy resource consumption. The authors propose the use of FEC for the core part in this reliability mode. But the fact that the NACK is sent by the sink node in a total reliability mechanism constitutes an overhead as the packet retransmissions could be asked by the intermediate nodes, without waiting for the sink request. The use of the FEC mechanism could also be used in the total reliability mechanism, and not restricted to a differentiated mechanism, as authors claim. Also, DTSN does not offer a congestion control mechanism in order to stop losses due to congestion. A lack of details in the use of the FEC mechanism is perceived in the study, as both hop-by-hop and end-to-end FEC mechanisms are present in literature. Also, a deeper comparison with reliability protocols for WSNs is missing.

RDFA (Reliable Data Forwarding Algorithm) [Jukka et al. 2009]: This protocol tries to ensure packet reliability even with high packet error rates (PERs). The authors use unlimited retransmissions at each hop using the ARQ policy. The protocol combines these retransmissions with a multipath routing strategy in order to overcome link failures by alternative available links, if needed. The protocol ensures event-based reliability by allowing packet drop after transmission failure while packet priority has a low value, which is an indication of its tolerance. But the protocol lacks a congestion control scheme in order to choose the appropriate rate that avoids congestion-based losses. Also, it is not compared with protocols designed for WSNs.

In Table VIII, a summary of upstream event + packet reliability protocols is depicted.

C.2. Bidirectional Protocols. In this subclass of protocols, the reliability is ensured in the two directions. In the downstream direction, the packet reliability is required, at least for a category of nodes. On the other hand, event reliability for upstream direction can be sufficient, even if packet reliability is also required sometimes for a category of nodes.

ART (Asymmetric and Reliable Transport) [Tezcan and W. Wang 2007]: This is interested in upstream event reliability and downstream query packet reliability together with congestion control. ART divides the network into E-nodes (essential) and N-nodes (nonessential) based on their energy supply, with E-nodes having the higher energy. A topology toward the sink is formed with these E-nodes, while N-nodes have to be connected to the near E-nodes. The E-nodes take on their charge the reliable forwarding of event messages toward the sink, and the query forwarding downstream toward the N-nodes by recovering lost messages as necessary. In case of query loss, the E-nodes send back NACK messages to the sink in order to trigger retransmissions in the sequence number holes state. ACK requests are sent to the sink after sending the event with the timeout retransmission strategy. The congestion control mechanism of ART is tuned by the E-nodes, as they send a congestion alarm (CA) to the one-hop N-nodes in order to stop their sending at the ACK loss from the sink. If this ACK loss persists, the E-node retransmits the CA message with a higher hop count in order to mitigate the congestion. In the case of ACK received from the sink, the transmission of N-nodes is resumed by sending them a congestion safe (CS) message. But considering ACK loss as a congestion indication could be wrong due to channel errors and path failure loss causes. ART does not invoke either the event or the query reliability in the N-nodes part, which could have importance in some cases where the event detection is not performed sufficiently. Further, the fairness aspect between E-nodes has been completely ignored.

HERO [Cañete et al. 2012] is a hierarchical routing protocol that constructs cluster-based topologies in order to be used for bidirectional transmission from the nodes to the cluster head through multihop and vice versa. HERO is fault tolerant and reliable through using the ACK mechanism with retransmissions. HERO also uses special nodes named clue nodes to estimate nodes' positions so that the cluster head sends the packets. The user could choose a defined reliability level, which will be transformed to a number of transmission attempts. It checks continuously new paths in order to be fault tolerant. HERO uses three operations repeatedly: (1) discovering/maintenance operation, which leads to (2) joining the network in order to perform (3) the routing operation. The number of retransmissions in HERO depends on the desired reliability and is also based on the links' quality. In the discovery phase, nodes choose the shortest paths toward the cluster head. After that, the joining phase consists of asking permission to join the cluster head.

In Table IX, the ART and HERO summary that belong to the bidirectional event + packet reliability class is highlighted.

6. DISCUSSION AND ANALYSIS

In this section, we will discuss different points related to reliability, starting with its definition, end-to-end or hop-by-hop control, different caching policies, acknowledgment of reception, recovery strategies, reliability consequence on delay, tradeoff between energy consumption and reliability level, effects of low security on system reliability,

Table VIII. Summary of Event+Packet Upstream Reliability Protocols

Protocol	Traffic Type	Reliability Level	Loss Detection and Notification	Loss Detection Node	Loss Recovery	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
STCP [Iyer et al. 2005]	Continuous/ event	Packet/event	EACK/NACK: E-to-E	Sink	Retransmission (E-to-E)	Yes	Simulation Tossim	Packet latency, energy spent	Alone
DTSN [B. Marchi and Nunes 2007]	Continuous	Packet/event	ACK, NACK: E-to-E	Sink	Retransmission, FEC (Erasure code)	No	Simulation OMNET++	Throughput, energy consumption	Caching strategies (100%, 50%)
RT2 [Gungor et al. 2008]	Continuous	Actuators: Packet/sensors: Event	Actuators: SACK (E-to-E)	Actuators: receiver	E-to-E: actuators (retransmission)/ sensors (multiple copies)	Yes	Simulation (NS2)	Convergence time, average energy consumption, aggregate throughput, average packet delay	ESRT, TCP-NewReno, TCP-ELFN, ATP
RDFA[Jukka et al. 2009]	Continuous	Packet/event	ACK: H-by-H	Source	Retransmission	No	Simulation	Energy efficiency, transmission attempts, goodput, reception interval, end-to-end latency	Multipath routing, traditional transport protocol
DRINA [Villas et al. 2013]	Periodic, event based	Aggregator node: Packet/other nodes: Event	ACK: H-by-H	Sender: H-by-H	Retransmission	No	Simulation (SinalGo)	Packet delivery rate, control overhead, efficiency, routing tree cost, loss of raw data, loss of aggregated data, transmissions number	InFRA, SPT
RARE [Zhao et al. 2013]	Periodic	Standard node: Packet/ transmit node: Event	ACK: one hop from sink	Sender	Retransmission	Yes	Simulation (Qualnet), analysis	Data delivery probability, QoS differentiation, system capacity, energy consumption, reliability	QoMoR

Table IX. Summary of Event+Packet Bidirectional Reliability Protocols

Protocol	Traffic Type	Loss Detection and Notification	Loss Detection Node	Loss Recovery	Congestion Control	Evaluation Type	Evaluation Parameters	Compared With
ART [Tezcan and W. Wang 2007]	Continuous query, event driven	Upstream: ACK, downstream: NACK	Up: Sink, down: E-nodes	Retransmission: E-to-E	Yes	Simulation NS2	Residual energy, network lifetime, E-to-E delay, loss ratio	Message delivery, no congestion control
HERO [Cañete et al. 2012]	Periodic, event based	ACK: H-by-H	Sender: H-by-H	Retransmission	No	COOJA simulator on Contiki	Energy efficiency, packet delivery ratio	IQ/MuMHR

importance of routing layer for enhancing reliability, and finally necessity of cross-layer design to ease reliability consideration.

Since reliability insurance gives rise to additional cost and overhead, ensuring a certain reliability level translates to a tradeoff between each of the previous points and reliability that must be taken into account. For that, in the following sections, we will show how the tradeoff vision could be ensured to minimize the cost and overhead while adopting an acceptable level of the aforementioned parameters.

6.1. Reliability Definition

Generally speaking, reliability of a system describes if the system works properly in order to reach its specifications [Silva et al. 2012]. For the sensing part of the system, it is considered reliable if the sensor nodes detect correctly the phenomenon. This concerns the number of sensor nodes that must be sufficient and redundant, especially due to their cheap cost, which could induce wrong reading values [Willig and Karl 2005]. For transport and even routing layers, reliability insurance covers the reception of the sent packets, but at the same time it concerns the reception of noncorrupted packets. From the reception point of view, we can talk about packets or event-based reliability. The first one concerns the delivery of each individual packet toward the destination, while the second one concerns the delivery of a given number of packets that permits the event interpretation, tolerating some packet loss. In case of losses, retransmission or redundancy-based recovery allows ensuring reliability. For ensuring the delivery of noncorrupted packets, physical security of wireless sensor nodes or software security through encoding mechanisms allows the reception of correct packets.

In WSNs, dataflows travel in two directions, leading to three reliability directions. Upstream reliability concerns the delivery of data from sensor nodes to the sink node. Downstream reliability concerns the delivery of control packets, queries from sink to sensor nodes, and reprogramming nodes, which is mostly a multicast/broadcast transfer. Bidirectional reliability provides both upstream (sensor-to-sink) and downstream (sink-to-sensor) delivery mechanisms [Sharma and Aseri 2012]. In general, downstream reliability that concerns control packets requires a high reliability level since its loss could seriously affect the system execution [Jones and Atiquzzaman 2007].

If a WSN covers different applications with different reliability requirements, namely, some needing packet reliability recovering and other ones with only event reliability exigencies, the chosen transport control protocol must answer this requirement [Wang et al. 2005]. For this reason, a configurable recovery mechanism is needed to support packet-level and application-level reliability, and to be helpful for energy conservation regarding the targeted application [Sharma and Aseri 2012; Rathnayaka and Potdar 2013].

In the following sections, we will discuss many points related to reliability insurance starting with the end-to-end or hop-by-hop insurance.

6.2. End to End Versus Hop by Hop

End-to-end reliability means that only the source and the final destination participate in the detection of losses and therefore the recovery mechanism. In hop-by-hop reliability, every intermediate node in the multihop path can detect losses and start the recovery mechanism. Due to the high level of loss seen generally in WSN applications, hop-by-hop reliability mechanisms are preferred over end-to-end ones. The hop-by-hop control avoids the delay and energy wastage due to losses that trigger recovering of previous hops, especially if it is caused by congestion [Rahman et al. 2008], but at the same time leads to control overhead at intermediate nodes [Jones and Atiquzzaman 2007]. On the other hand, hop-by-hop control could fail if losses are caused by topology change or by the intermediate nodes' death. So, a tradeoff between end-to-end and hop-by-hop control can be a good alternative [Rahman et al. 2008].

6.3. Caching Policies

As invoked in the previous section, a tradeoff between end-to-end and hop-by-hop control could be a good choice. This poses the question of where to cache the packets in order to ensure the recovery process. Choosing the loss node point, or a closer node to it, as the recovery one is also a good idea. It concerns the bottleneck link if these losses are due to congestion, or the node enabling the lossy or failing link in the case of losses due to topology change or bad link quality. If the recovery node is not the same as the loss one, the number of hops between the two nodes is called the recovery distance, which gives an estimation about the involved cost in terms of energy [Wang et al. 2006]. Another question that arises is how long to cache these packets in case of losses. For hop-by-hop control, this duration is simple to compute and covers the sum of service time and one-hop propagation delay. On the other hand, the end-to-end caching period is more complicated to be computed and have not to be less than a round trip time (RTT). In the two cases of control, the source and final destination have to cache the packets for reliability purposes. The final one does it for the purpose of information integrity, while the source one caches the packets in order to allow the recovery if the intermediate caching points fail from recovering [Rahman et al. 2008].

A caching mechanism that takes into account packets' origin and priority will be a good choice. In this mechanism, control or reprogramming code packets will get high priority, and caching them in all nodes or in most nodes will be beneficial to recover from losses or corruption.

6.4. Acknowledgment

For ensuring data reception reliability, an acknowledgment must be sent by the receiver toward the sender or to the intermediate corresponding storage node. Positive acknowledgment could be used as an explicit packet (EACK), or implicitly (IACK) in order to inform the sender. On the other hand, and to avoid overhead that could be caused by the positive ACK, negative acknowledgment (NACK) could be sent each time a packet is expected but not received. Another form of acknowledgment packet is selective ACK (SACK), where the receiver sends a summary of packets received correctly and incorrectly (out of sequence). In general, hop-by-hop ACKs are performed at the MAC layer. But with low bit error links, it is energy saving not to use them. In contrast, with high loss rate links, the end-to-end loss recovery has to be used in addition to MAC layer hop-by-hop ones [Willig and Karl 2005]. If the application requires the reception of one packet at each event or query, the NACK message could not be used alone [Park and Sivakumar 2003]. According to application importance and link loss rate, a tradeoff mechanism of using acknowledgment frequency could be used in order to decrease the end-to-end delay that retransmissions involve on the application performance.

6.5. How to Ensure Reliability (Retransmission/Redundancy)

In order to recover from any loss, recovery mechanisms are needed for repairing these losses. This reparation could be done through retransmitting the lost packets, which could be done at the MAC layer and is called Automatic Repeat Request (ARQ). The other method consists of adding some redundant information to the original ones to allow reconstructing the original data by receiving a fraction of the sending packets; this is called Forward Error Control (FEC) and does not need any acknowledgment mechanism. The amount of redundancy information has a relation to the loss rate [Mahmood et al. 2015]. Combining retransmission-based and redundant-based control could be a good choice in order to gain from the two strategies. A first combination consists of sending redundant information by the first retransmission request. A second combination consists of sending redundant information from the beginning and retransmitting lost packets if these redundant packets are still lost due to the high loss level [Mahmood et al. 2015; Carle and Biersack 1997; Kim et al. 2004]. Some redundant-based strategies consist of sending multiple copies of the same data rather than sending additional packets to the original ones. This strategy could be not energy efficient due to the excess of sending. So, retransmission of lost packets at the loss happening could be more energy efficient [Wang et al. 2006]. Generally speaking, a redundant-based mechanism presents low performance when the loss rate increases, because the recovery requires a certain number of packets to allow the reconstruction of original packets. Adding more redundant packets for permitting the recovery could have bad results, such as creating congestion and consuming energy. Choosing dynamically the level of redundancy in relation to network characteristics is a more efficient strategy [Mahmood et al. 2015].

6.6. Reliability Versus Timeliness (Delay)

In practice, applications could have different requirements in terms of delivery reliability and delivery latency, also named application responsiveness. We can find different degrees of these QoS requirements regarding the aim of the application. For example, military applications (like battlefield surveillance) need high reliability in minimal latency, while home automation applications generally do not need high reliability and could tolerate some losses [Shaikh et al. 2007]. On the other hand, some scientific applications could give more importance to reliability by gathering all sensed values and could tolerate a high latency forwarding [Rathnayaka and Potdar 2013; Katiyar et al. 2012]. The conception of a reliability mechanism that guaranties a tradeoff between the required reliability level and data freshness is mandatory in order to answer the application goals.

6.7. Reliability Versus Energy

Ensuring high reliability is generally contradictory with minimizing energy consumption. This is due to the fact that high reliability requires recovery mechanisms that consume more energy [Antônio et al. 2014]. The use of retransmissions consumes due to the sending of new packets, while redundancy mechanisms also consume power energy due to the multiple copies of the same packet or introducing more data to the original one. On the other hand, if reliability requirements could be fulfilled through avoiding congestion, this will ensure more reliability by prohibiting buffer overflow and help to save the sensors' energy [Jones and Atiquzzaman 2007]. On the other side, channel errors affect both the reliability and energy depletion [Wang et al. 2006].

Choosing an appropriate recovery mechanism like the use of recovery point control (end to end or hop by hop) also affects the energy consumption [Willig and Karl 2005; Dunkels et al. 2004]. So, a good design of deployment could help to minimize energy consumption while ensuring a high reliability level. This could be done by avoiding excessive redundancy in the same deployment area, as well as contention due to simultaneous transmissions at the same neighbor [Katiyar et al. 2012]. From the application point of view, a tradeoff between reliability level and energy consumption could be mandatory in order to extend the application lifetime [Rathnayaka and Potdar 2013].

6.8. Reliability and Security

Ensuring reliable information has a strong relation with the physical location of the sensors, which could be in harsh environments and exposed to malicious individuals like that in battlefield deployment, or to software attacks in order to change data content and delay application goals. In addition, the wireless links used in WSN applications promote their compromise. Broadly speaking, as the domain of use becomes more critical, ensuring security becomes a must. On the other hand, the limited power and processing capabilities avoid the use of standard security mechanisms [Katiyar et al. 2012]. Also, using hop-by-hop recovery and control rather than end-to-end mechanisms leads to many vulnerability point entries, as the end-to-end encryption will not be possible [Pereira et al. 2007].

As TSCH scheduling is used more and more with LLN applications for IOT, securing mechanisms have to be handled in order to choose as short a slot size as possible while offering security in the link layer. This can be done through hardware accelerators and software. Enabling short slots needs more attention and development [Sciancalepore et al. 2017]. In addition, standardization of security protocols in the field of IOT seems necessary in order to offer more interoperability of devices that characterizes the IOT use [Keoh et al. 2014]. IETF groups propose many standardization initiatives for IOT compliance. Using existed mechanisms like IPsec for the IOT applications also leads researchers to propose enhancements so that they can be used with LLN applications [Raza et al. 2011, 2014].

For physical sensors' security in critical applications, it would be interesting to develop a mechanism that gets rid of the sensor that is manipulated by the enemies since it is touched to avoid its snooping. On the other hand, for software security, it will be interesting to use a coding mechanism according to the packet priority to avoid coding low-priority packets. Also, it is possible to choose different coding schemes concerning their capacity and memory cost according to data or control packets. In brief, a tradeoff coding mechanism based on priorities constitutes a good choice.

6.9. Effects of Routing and Multipath on Reliability

As explained in the previous sections, reliability insurance is generally contradictory with resource and consumption minimization. From a routing layer point of view, this is applicable since single-path routing consumes fewer resources than multipath routing but offers less reliability and throughput, especially in case of route failure or packet loss. Using multiple paths or having a backup path helps to gain in delay and throughput in the case of failures in the primary path, since the discovery time of the new path is not needed compared to single-path routing [Kim et al. 2004]. The use of multipath routing is possible with WSN applications due to their dense deployment and their short-range coverage. Multipath routing could be used concurrently to balance the channel overhead and contention, or one path at a time. In case of using concurrent paths, different copies of the same packets could be forwarded in order to reach a high reliability level, or the erasure coding technique could be used by adding additional information to the original ones. After that, the packets are distributed toward different paths [Willig and Karl 2005]. The reception of a portion of these packets allows the reconstruction of the original information. Path disjointedness is used to evaluate the consistency of multipath methods, as paths are disjoint as the system is consistent against failures. Node disjointedness is more consistent than link disjointedness, since having two paths with no common intermediate nodes allows system functionality even with one node failure, contrary to link disjointedness that could not face node failure [Marjan et al. 2012]. Another parameter that is important to guarantee a high reliability level in the routing layer is the cost function used to choose a routing node, which could be its hop count or its link quality, for example. Choosing a link due to its acceptable link quality also ensures an acceptable reliability level, as this link will lead to minimum retransmissions. This metric is called Minimum Transmission (MT) [Zhao and Govindan 2003; Woo et al. 2003]. There exist several other cost functionalities like delay, network lifetime, link capacity, and path throughput [Marjan et al. 2012]. With large-scale WSN applications, it is mandatory to balance the network structure by the minimization of hop counts, in addition to choosing good links between nodes [Korkmaz and Sarac 2010].

6.10. Mobility Handling Effects on Reliability

In many WSN-based applications, mobile nodes are part (or the only part) of the network. In some applications, the mobile nodes are the basis of the application itself, and their strong mobility (like vehicular networks) could lead to a lack of reliability due to frequent topology breakage. Therefore, special features should be taken in order to accomplish the application requirements. On the other hand, in some applications, introducing mobile nodes could enhance reliability through data collection based on these mobile nodes (sinks or cluster heads), especially with static WSN topologies where the only mobile nodes are those collecting data. Handling the topology containing mobile nodes becomes more difficult and needs more memory resources and deep conception, especially with strong mobility. For example, establishing trajectory plans for gathering clusters' data, through mobile sinks or special nodes, needs minimizing energy consumption at the same time of maximizing network coverage. These plans could follow random, controlled, or predetermined trajectories [Silva et al. 2014].

From the layers stack view, handling mobility at the MAC level, adding even more complexity, brings more robustness to that of the network layer. In fact, handling duty cycling, latency, and mobility, with a minimum of signalization, ensures more reliability and application throughput. With the IOT applications enlargement, taking the special features of LLN networks while proposing mobility solutions leads also to more robustness and saves precious nodes' resources [Bouaziz and Rachedi 2016]. Proposing mobility solutions based on IP compliance and being interoperable with 6loWPAN behaviors could facilitate the scalability use [Ghaleb et al. 2016]. In all cases, reducing the lack of reliability that manifests through packet loss or latency needs to be treated while conceiving mobility-based network and MAC layer solutions.

6.11. Cross-Layer Design Necessity

Even the independence of different layers' conception is seen as a positive point, but their information exchange brings a lot of benefit for WSN conception. This cross-layer design will help fulfill reliability requirements, which is done in general at the transport layer. In order to provide a good reliability level for the system and save the precious energy supply, it is important to know the original cause of packet loss or corruption before triggering any recovery method.

In fact, resource control is the basis for reliability enhancements. This could be done through adding more resources in order to reach a given reliability level, activating

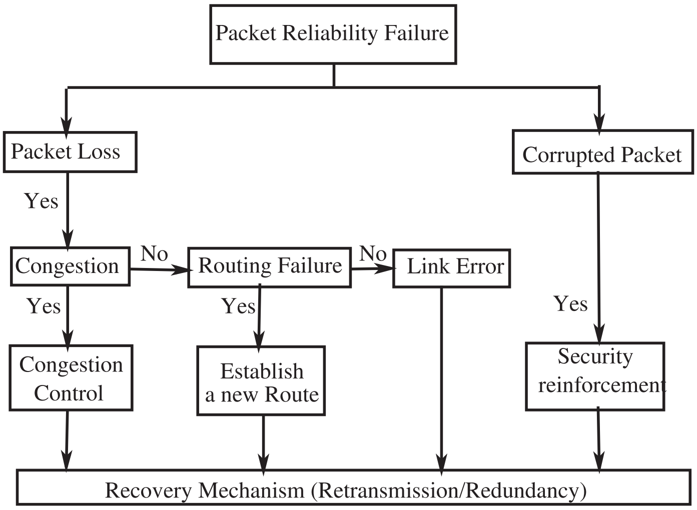

Fig. 9. Recovery strategies depending on packet loss causes.

some sleeping nodes that activate new paths, or increasing the reading of some existing nodes if the lack of reliability is due to insufficient values. On the other hand, if the lack of reliability is due to excessive values that manifest in the form of contention or congestion, halting some low-priority reading could be the right solution. If the packet loss is due to congestion and therefore buffer overflow, it is a must to control this congestion through rate reduction or other control and delaying any retransmission-based recovery, since it will be lost until the congestion level is reduced [Wang et al. 2006; Mahmood et al. 2015]. The MAC layer could also prioritize the concerned links in order to drain the packets to send, in addition to eventual rate reduction. On the other hand, if the packet loss is due to route failure rather than congestion, the routing layer could inform the transport layer about this happening in order to avoid any rate reduction that will not be necessary. In addition, prohibiting immediate recovery until finding a new route or using a backup route is necessary, since any recovery will waste the energy before a new path will be constructed. If the loss is due to link contention and interference, scheduling the communication will be the main solution unless losses will aggravate the situation [Kafi et al. 2014b; Kafi et al. 2016]. Also, if the loss is due to channel error, no rate reduction is needed, just simply recovering the packet in order to keep high throughput and link utilization. Otherwise, changing the physical location of the node or changing the transmission power could be useful too [Rathnayaka and Potdar 2013; Rahman et al. 2008; Katiyar et al. 2012; Pereira et al. 2007; Gungor and Akan 2006]. Finally, if the packets are corrupted due to security failures, reinforcing the physical nodes' security or their messages' encryption will be the appropriate solution. The scheme in Figure 9 summarizes the previous discussion.

7. CONCLUSION

WSN applications know more and more success and use, either in civil or military applications. Producing and communicating the sensed data toward the basestation reliably are very important; otherwise, the application goals could not be reached or could lead to wrong results and decisions. In this study, we have surveyed different protocols dealing with reliability conception. This field is essential to offer the good functioning of applications, avoid loss of important data, and waste scarce energy. The deep classification and analysis of these protocols have shown that reliability concerns different layers rather than a single one, starting from the MAC, to routing, to arriving, to transport layer. We have categorized the existing works according to the required reliability level, the direction of the transmitted packets, and the cause of the lack of reliability. In addition, the methods used for recovering the lost packets were discussed. Our analyses reveal that none of the proposed protocols detects and/or overcomes all the kinds of lack of reliability. For that purpose, we have discussed their shortcomings in order to allow thinking about their enhancement to face lack of reliability. Cross-layer design of new reliability protocols represents a good choice in order to recover all the possible reliability failures. We believe that this support constitutes a good reference for the conception of new reliability mechanisms.

REFERENCES

H. M. F. AboElFotoh, E. S. Elmallah, and H. S. Hassanein. 2006. On the reliability of wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC'06)*, Vol. 8. 3455–3460.

M. M. Alam and C. S. Hong. 2009. CRRT: Congestion-aware and rate-controlled reliable transport in wireless sensor networks. *Ieice Transactions* 92-B, 1 (2009), 184–199.

E. Ancillotti, R. Bruno, and M. Conti. 2014. Reliable data delivery with the IETF routing protocol for low-power and lossy networks. *IEEE Transactions on Industrial Informatics* 10, 3 (Aug. 2014), 1864–1877.

D. Antônio, R. Nelson, and M. Paulo. 2014. Reliability of wireless sensor networks. *Sensors* 14 (2014), 15760–15785.

A. Ayadi. 2011. Energy-efficient and reliable transport protocols for wireless sensor networks: State-of-art. *Wireless Sensor Network* 3 (2011), 106–113.

A. Grilo, B. Marchi, and M. Nunes. 2007. DTSN: Distributed transport for sensor networks. In *Proceedings of the 12th IEEE Symposium on Computers and Communications (ISCC'07)*. 01–04.

A. B. Bagula and K. G. Mazandu. 2008. Energy constrained multipath routing in wireless sensor networks. In *Ubiquitous Intelligence and Computing*. Lecture Notes in Computer Science, Vol. 5061. Springer, Berlin, 453–467.

D. Bein, V. Jolly, B. Kumar, and S. Latifi. 2005. Reliability modeling in wireless sensor networks. *International Journal of Information Technology* 11, 2 (2005), 1–9.

J. Ben-Othman and B. Yahya. 2010. Energy efficient and QoS based routing protocol for wireless sensor networks. *Journal of Parallel and Distributed Computing* 70, 8 (2010), 849–857.

M. Bouaziz and A. Rachedi. 2016. A survey on mobility management protocols in wireless sensor networks based on 6LoWPAN technology. *Computer Communications* 74 (2016), 3–15.

T. Braun, T. Voigt, and A. Dunkels. 2007. TCP support for sensor networks. In *4th Annual Conference on Wireless on Demand Network Systems and Services (WONS'07)*. 162–169.

E. Cañete, M. Díaz, L. Llopis, and B. Rubio. 2012. HERO: A hierarchical, efficient and reliable routing protocol for wireless sensor and actor networks. *Computer Communications* 35, 11 (2012), 1392–1409.

G. Carle and E. W. Biersack. 1997. Survey of error recovery techniques for IP-based audio-visual multicast applications. *IEEE Network* 11, 6 (Nov. 1997), 24–36.

I.-R. Chen and Y. Wang. 2012. Reliability analysis of wireless sensor networks with distributed code attestation. *IEEE Communications Letters* 16, 10 (Oct. 2012), 1640–1643.

B. Deb, S. Bhatnagar, and B. Nath. 2003. ReInForM: Reliable information forwarding using multiple paths in sensor networks. In *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)*. 406–415.

A. Dunkels, J. Alonso, and T. Voight. 2004. Making TCP/IP viable for wireless sensor networks. In *Proceedings of European Workshop on Wireless Sensor Networks (EWSN'04)*. 1–4.

E. Felemban, C.-G. Lee, and E. Ekici. 2006. MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and. timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing* 5, 6 (June 2006), 738–754.

D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. 2001. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mobile Computing Communications Review.* 5, 4 (Oct. 2001), 11–25.

A. Ghaffari. 2015. Congestion control mechanisms in wireless sensor networks: A survey. *Journal of Network and Computer Applications* 52 (2015), 101–115.

S. M. Ghaleb, S. Subramaniam, Z. A. Zukarnain, and A. Muhammed. 2016. Mobility management for IoT: A survey. *EURASIP Journal on Wireless Communications and Networking* 2016, 1 (2016), 165.

E. Giancoli, F. Jabour, and A. Pedroza. 2008. CTCP: Reliable transport control protocol for sensor networks. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'08)*. 493–498.

O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis. 2013. CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks. *ACM Transactions on Sensor Networks* 10, 1 (Dec. 2013), 16:1–16:49.

V. C. Gungor and O. B. Akan. 2006. DST: Delay sensitive transport in wireless sensor networks. In *Proceedings of the 7th IEEE International Symposium on Computer Networks*. 116–122.

V. C. Gungor, Ö. B. Akan, and I. F. Akyildiz. 2008. A real-time and reliable transport (RT)2 protocol for wireless sensor and actor networks. *IEEE/ACM Transactions on Networking* 16, 2 (April 2008), 359–370.

X. Huang and Y. Fang. 2008. Multiconstrained QoS multipath routing in wireless sensor networks. *Wireless Networks* 14, 4 (Aug. 2008), 465–478.

C. Intanagonwiwat, R. Govindan, and D. Estrin. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)*. 56–67.

R. Iyer and L. Kleinrock. 2003. QoS control for sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC'03)*, Vol. 1. 517–521.

Y. G. Iyer, S. Gandham, and S. Venkatesan. 2005. STCP: A generic transport layer protocol for wireless sensor networks. In *Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN'05)*. 449–454.

Z. Jin, T. Liang, X. Hongsheng, and Z. Zhenghuan. 2012. Reliability analysis of wireless sensor networks using Markovian model. *Journal of Applied Mathematics* 2012 (2012).

J. Jones and M. Atiquzzaman. 2007. Transport protocols for wireless sensor networks: State-of-the-art and future directions. *International Journal of Distributed Sensor Networks* 3 (2007), 119–133.

S. Jukka, D. Hamalainen Timo, and H. Marko. 2009. Availability and end-to-end reliability in low duty cycle multihopwireless sensor networks. *Sensors* 9 (2009), 2088–2116.

C. Jaggle, J. Neidig, T. Grosch, and F. Dressler. 2009. Introduction to model-based reliability evaluation of wireless sensor networks. In *Proceedings of the 2nd IFAC Workshop on Dependable Control of Discrete Systems*.

M. A. Kafi, J. B. Othman, M. Bagaa, and N. Badache. 2015. CCS_WHMS: A congestion control scheme for wearable health management system. *Journal of Medical Systems* 39, (2015).

M. A. Kafi, J. B. Othman, A. Ouadjaout, M. Bagaa, and N. Badache. 2016. REFIACC: Reliable, efficient, fair and interference-aware congestion control protocol for wireless sensor networks. *Computer Communications* 101 (2016), 1–11.

M. A. Kafi, Y. Challal, D. Djenouri, A. Bouabdallah, L. Khelladi, and N. Badache. 2012. A study of wireless sensor network architectures and projects for traffic light monitoring. *Procedia Computer Science* 10 (2012), 543–552.

M. A. Kafi, Y. Challal, D. Djenouri, M. Doudou, A. Bouabdallah, and N. Badache. 2013. A study of wireless sensor networks for urban traffic monitoring: Applications and architectures. *Procedia Computer Science* 19 (2013), 617–626. The *Proceedings of the 4th International Conference on Ambient Systems, Networks and Technologies (ANT'13).*

M. A. Kafi, D. Djenouri, J. Ben-Othman, and N. Badache. 2014. Congestion control protocols in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials* 16, 3 (2014), 1369–1390.

M. A. Kafi, D. Djenouri, J. B. Othman, A. Ouadjaout, and N. Badache. 2014a. Congestion detection strategies in wireless sensor networks: A comparative study with testbed experiments. *Procedia Computer Science* 37 (2014), 168–175.

M. A. Kafi, D. Djenouri, J. B. Othman, A. Ouadjaout, M. Bagaa, N. Lasla, and N. Badache. 2014b. Interference-aware congestion control protocol for wireless sensor networks. *Procedia Computer Science* 37 (2014), 181–188.

M. Katiyar, H. P. Sinha, and D. Gupta. 2012. On reliability modeling in wireless sensor networks-A review. *IJCSI International Journal of Computer Science Issues* 9, 6 (Nov. 2012), 134–146.

S. L. Keoh, S. S. Kumar, and H. Tschofenig. 2014. Securing the internet of things: A standardization perspective. *IEEE Internet of Things Journal* 1, 3 (June 2014), 265–275.

S. Kim, R. Fonseca, and D. Culler. 2004. Reliable transfer on wireless sensor networks. In *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*. 449–459.

S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. 2007. Flush: A reliable bulk transport protocol for multihop wireless networks. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys'07)*. 351–365.

T. Korkmaz and K. Sarac. 2010. Characterizing link and path reliability in large-scale wireless sensor networks. In *Proceedings of the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'10)*. 217–224.

R. Kumar, A. Paul, U. Ramachandran, and D. Kotz. 2006. On improving wireless broadcast reliability of sensor networks using erasure codes. In *Mobile Ad-hoc and Sensor Networks*. Lecture Notes in Computer Science, Vol. 4325. Springer, Berlin, 155–170.

S. Kumar, Z. Feng, F. Hu, and Y. Xiao. 2009. E2SRT: Enhanced event-to-sink reliable transport for wireless sensor networks. *Wireless Communications and Mobile Computing* 9, 10 (2009), 1301–1311.

T. Le, W. Hu, P. Corke, and S. Jha. 2009. ERTP: Energy-efficient and reliable transport protocol for data streaming in wireless sensor networks. *Computer Communications* 32, 7–10 (2009), 1154–1171.

W. Lou. 2005. An efficient N-to-1 multipath routing protocol in wireless sensor networks. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*.

W. Lou and Y. Kwon. 2006. H-SPREAD: A hybrid multipath scheme for secure and reliable data collection in wireless sensor networks. *IEEE Transactions on Vehicular Technology* 55, 4 (July 2006), 1320–1330.

W. Lou, W. Liu, and Y. Fang. 2004. SPREAD: Enhancing data confidentiality in mobile ad hoc networks. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, Vol. 4. 2404–2413.

M. A. Mahmood and W. K. G. Seah. 2011. Event reliability in wireless sensor networks. In *7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'11)*. 377–382.

M. A. Mahmood, W. K. G. Seah, and I. Welch. 2015. Reliability in wireless sensor networks: A survey and challenges ahead. *Computer Networks* 79, (2015), 166–187.

P. Di Marco, P. Park, C. Fischione, and K. H. Johansson. 2010. TREnD: A timely, reliable, energy-efficient and dynamic WSN protocol for control applications. In *Proceedings of the 2010 IEEE International Conference on Communications*. 1–6.

R. Marjan, D. Behnam, A. B. Kamalrulnizam, and L. Malrey. 2012. Multipath routing in wireless sensor networks: Survey and research challenges. *Sensors* 12 (2012), 650–685.

M. S. Nassr, J. Jun, S. J. Eidenbenz, A. A. Hansson, and A. M. Mielke. 2007. Scalable and reliable sensor network routing: Performance study from field deployment. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07)*. 670–678.

J. Niu, L. Cheng, Y. Gu, L. Shu, and S. K. Das. 2014. R3E: Reliable reactive routing enhancement for wireless sensor networks. *IEEE Transactions on Industrial Informatics* 10, 1 (Feb. 2014), 784–794.

A. Ouadjaout, N. Lasla, M. Bagaa, M. Doudou, C. Zizoua, M. A. Kafi, A. Derhab, D. Djenouri, and N. Badache. 2014. DZ50: Energy-efficient wireless sensor mote platform for low data rate applications. *Procedia Computer Science* 37 (2014), 189–195. *The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN'14)*.

J. Paek and R. Govindan. 2007. RCRT: Rate-controlled reliable transport for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys'07)*.

M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia. 2012. Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks. In *Proceedings of the 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'12)*. 327–332.

M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel. 2013. On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH. *IEEE Sensors Journal* 13, 10 (Oct. 2013), 3655–3666.

M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. A. Grieco, and T. Engel. 2016. On-the-fly bandwidth reservation for 6TiSCH wireless industrial networks. *IEEE Sensors Journal* 16, 2 (Jan. 2016), 550–560.

S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz. 2004. A scalable approach for reliable downstream data delivery in wireless sensor networks. In *Proceedings of the 5th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'04)*. 78–89.

S.-J. Park and R. Sivakumar. 2003. Sink-to-sensors reliability in sensor networks. *SIGMOBILE Mobile Computing Communications Review* 7, 3 (July 2003), 27–28.

S.-J. Park, R. Sivakumar, I. F. Akyildiz, and R. Vedantham. 2008. GARUDA: Achieving effective reliability for downstream communication in wireless sensor networks. *IEEE Transactions on Mobile Computing* 7, 2 (Feb. 2008), 214–230.

P. R. Pereira, A. Grilo, F. Rocha, M. S. Nunes, A. Casaca, C. Chaudet, P. Almstrom, and M. Johansson. 2007. End to end reliability in wireless sensor networks: Survey and research challenges. In *Proceedings of the EuroFGI Workshop on IP QoS and Traffic Control*.

H. K. Qureshi, S. Rizvi, M. Saleem, S. A. Khayam, V. Rakocevic, and M. Rajarajan. 2011. Poly: A reliable and energy efficient topology control protocol for wireless sensor networks. *Computer Communications* 34, 10 (2011), 1235–1242.

Md. A. Rahman, A. El Saddik, and W. Gueaieb. 2008. Wireless sensor network transport layer: State of the art. *Sensors, Lecture Notes Electrical Engineering, Springer, Heidelberg, Germany* 21 (2008), 221–245.

A. J. D. Rathnayaka and V. M. Potdar. 2013. Wireless sensor network transport protocol: A critical review. *Journal of Network and Computer Applications* 36, 1 (2013), 134–146.

S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig. 2011. Securing communication in 6LoWPAN with compressed IPsec. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS'11)*. 1–8.

S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt. 2014. Secure communication for the internet of things—A comparison of link-layer security and IPsec for 6LoWPAN. *Security and Communication Networks* 7, 12 (2014), 2654–2668.

J. M. Reason and J. M. Rabaey. 2004. A study of energy consumption and reliability in a multi-hop sensor network. *SIGMOBILE Mobile Computing Communications Review* 8, 1 (Jan. 2004), 84–97.

I. S. Reed and G. Solomon. 1960. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics* 8, 2 (June 1960), 300–304.

M. H. S. Naziri and S. Hasanpoor. 2011. Improving lifetime and reliability in routing real-time wireless sensor networks based on hybrid algorithm. *Australian Journal of Basic and Applied Sciences,* 5, 9 (2011), 1105–1109.

Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz. 2003. ESRT: Event-to-sink reliable transport in wireless sensor networks. In *Proceedings of 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03)*. 177–188.

S. Sciancalepore, M. Vučinić, G. Piro, G. Boggia, and T. Watteyne. 2017. Link-layer security in TSCH networks: Effect on slot duration. *Transactions on Emerging Telecommunications Technologies* 28, 1 (2017).

W. K. G. Seah and H. P. Tan. 2006. Multipath virtual sink architecture for wireless sensor networks in harsh environments. In *Proceedings of the 1st International Conference on Integrated Internet Ad Hoc and Sensor Networks (InterSense'06)*.

C. Sergiou, P. Antoniou, and V. Vassiliou. 2014. A comprehensive survey of congestion control protocols in wireless sensor networks. *IEEE Communications Surveys Tutorials* 16, 4 (2014), 1839–1859.

F. K. Shaikh, A. Khelil, A. Ali, and N. Suri. 2010. TRCCIT: Tunable reliability with congestion control for information transport in wireless sensor networks. In *Proceedings of the 5th Annual ICST Wireless Internet Conference (WICON'10)*. 1–9.

F. K. Shaikh, A. Khelil, and N. Suri. 2007. On modeling the reliability of data transport in wireless sensor networks. In *15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing (PDP'07)*. 395–402.

F. K. Shaikh, A. Khelil, and N. Suri. 2008. A comparative study of data transport protocols in wireless sensor networks. In *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'08)*. 1–9.

F. K. Shaikh, A. Khelil, and N. Suri. 2009. AReIT: Adaptable reliable information transport for service availability in wireless sensor networks. In *Proceedings of International Conference on Wireless Networks (ICWN'09)*. 75–81.

A. Sharif, V. M. Potdar, and A. J. D. Rathnayaka. 2010. ERCTP: End-to-end reliable and congestion aware transport layer protocol for heterogeneous WSN. *Scientific International Journal for Parallel and Distributed Computing* 11, 4 (2010), 359–371.

B. Sharma and T. C. Aseri. 2012. A comparative analysis of reliable and congestion-aware transport layer protocols for wireless sensor networks. *International Scholarly Research Network (ISRN) Sensor Networks* (2012).

J. Shin, U. Ramachandran, and M. Ammar. 2007. On improving the reliability of packet delivery in dense wireless sensor networks. In *Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN'07)*. 718–723.

I. Silva, L. A. Guedes, P. Portugal, and F. Vasques. 2012. Reliability and availability evaluation of wireless sensor networks for industrial applications. *Sensors* 12, 1 (2012), 806–838.

R. Silva, J. Sa Silva, and F. Boavida. 2014. Mobility in wireless sensor networks – Survey and proposal. *Computer Communications* 52 (2014), 1–20.

M. S. Srouji, Z. Wang, and J. Henkel. 2011. RDTS: A reliable erasure-coding based data transfer scheme for wireless sensor networks. In *Proceedings of the IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS'11)*. 481–488.

F. Stann and J. Heidemann. 2003. RMST: Reliable data transport in sensor networks. In *Proceedings of the 1st International Workshop on Sensor Net Protocols and Applications*. 102–112.

S. Suganya, P. Prabaharan, and L. Malathi. 2013. A survey on multipath routing protocols for reliable data transmission in wireless sensor networks. *International Journal of Research in Computer and Communication Technology* 2, 11 (Nov. 2013), 1218–1221.

K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar. 2005. ATP: A reliable transport protocol for Ad Hoc networks. *IEEE Transactions on Mobile Computing* 4, 6 (2005), 588–603.

N. Tezcan and W. Wang. 2007. ART: An asymmetric and reliable transport mechanism for wireless sensor networks. *International Journal of Sensor Networks,* 2, 3–4 (2007), 188–200.

L. A. Villas, A. Boukerche, H. S. Ramos, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro. 2013. DRINA: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks. *IEEE Transactions on Computing* 62, 4 (April 2013), 676–689.

C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. 2002. PSFQ: A reliable transport protocol for wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*. 1–11.

C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. 2005. Pump-slowly, fetch-quickly (PSFQ): A reliable transport protocol for sensor networks. *IEEE Journal on Selected Areas in Communications* 23, 4 (2005), 862–872.

C. Wang, K. Sohraby, Y. Hu, B. Li, and W. Tang. 2005. Issues of transport control protocols for wireless sensor networks. In *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS'05)*. 422–426.

C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu. 2006. A survey of transport protocols for wireless sensor networks. *IEEE Network* 20, 3 (May 2006), 34–40.

H. Wen, C. Lin, F. Ren, Y. Yue, and X. Huang. 2007. Retransmission or redundancy: Transmission reliability in wireless sensor networks. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'07)*. 1–7.

A. Willig and H. Karl. 2005. Data transport reliability in wireless sensor networks – A survey of issues and solutions. *Praxis der Informationsverarbeitung und Kommunikation* 28, 2 (December 2005), 86–92.

T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander. 2012. RPL: IPv6 routing protocol for low-power and lossy networks. *RFC 6550, IETF, March* (2012).

A. Woo, T. Tong, and D. Culler. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*. 14–27.

N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. 2004. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. 13–24.

Y. Xue, B. Ramamurthy, and Y. Wang. 2009. LTRES: A loss-tolerant reliable event sensing protocol for wireless sensor networks. *Computer Communications* 32, 15 (2009), 1666–1676.

S. Zafar. 2011. A survey of transport layer protocols for wireless sensor networks. *International Journal of Computer Applications* 33, 1 (Nov. 2011), 44–50.

H. Zhang, A. Arora, Y-r. Choi, and M. G. Gouda. 2007. RBC: Reliable bursty convergecast in wireless sensor networks. *Computer Communications* 30, 13 (2007), 2560–2576.

J. Zhao and R. Govindan. 2003. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*. 1–13.

J. Zhao, C. Qiao, R. S. Sudhaakar, and S. Yoon. 2013. Improve efficiency and reliability in single-hop WSNs with transmit-only nodes. *IEEE Transactions on Parallel and Distributed Systems* 24, 3 (March 2013), 520–534.

H. Zhou, X. Guan, and C. Wu. 2008. Reliable transport with memory consideration in wireless sensor networks. In *IEEE International Conference on Communications (ICC'08)*. 2819–2824.

Q. Zhou. 2012. Integrated reliability modelling for wireless sensor networks, wireless sensor networks - technology and applications. In *Tech*, M. Matin (Ed.). DOI:10.5772/48429. Available from https://www.intechopen.com/books/wireless-sensor-networks-technology-and-applications/integrated-reliability-modelling-for-wireless-sensor-networks.

Y. Zhou, M. R. Lyu, J. Liu, and H. Wang. 2005. PORT: A price-oriented reliable transport protocol for wireless sensor network. In *Proceedings of 16th IEEE International Symposium on Software Reliability Engineering*. 117–126.

J. Zhu, K.-L. Hung, B. Bensaou, and F. Nait-Abdesselam. 2008. Rate-lifetime tradeoff for reliable communication in wireless sensor networks. *Computer Networks* 52, 1 (2008), 25–43.

A. E. Zonouz, L. Xing, V. M. Vokkarane, and Y. Sun. 2013. Application communication reliability of wireless sensor networks supporting K-coverage. In *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'13)*. 430–435.

Received October 2016; revised March 2017; accepted March 2017