A Comprehensive Survey of Congestion Control Protocols in Wireless Sensor Networks

Charalambos Sergiou, *Member, IEEE*, Pavlos Antoniou, and Vasos Vassiliou, *Member, IEEE*

***Abstract*—Congestion control and reliable data delivery are two primary functions of the transport layer in wired and wireless networks. Wireless sensor networks (WSNs) are a special category of wireless ad hoc networks with unique characteristics and important limitations. Limitations concern their resources, such as energy, memory, and computational power, as well as their applications. Due to these limitations and characteristics, the Transmission Control Protocol (TCP), the legacy protocol that implements congestion control and reliable transmission in the Internet, cannot apply to WSNs in its traditional form. To deal with this unavailability of a standard solution, many efforts are taking place in this area. In this paper, we review, classify, and compare algorithms, protocols, and mechanisms that deal directly with congestion control and avoidance in WSNs.**

***Index Terms*—Wireless sensor networks (WSNs), congestion control, congestion avoidance, reliable transmission.**

I. Introduction

WIRELESS SENSOR NETWORKs (WSNs) are wireless networks consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, and pollutants, among others, at different locations [1], [2]. Applications using sensors are being increased. A wide range of them is now deployed in civilian areas like habitat observation [3], [4], health monitoring [5], object tracking [6], [7] etc. Intense study has been carried out concerning many aspects of WSNs especially in the physical layer [8], [9], MAC layer [10]–[12], and network layer [13]–[16]. Lately, the problem of congestion control and avoidance has also attracted a lot of attention. Many research efforts exist in literature that justify the need of congestion control in WSNs. Papers like [17] argue on this issue and provide numerical results, while a number of other documents like [18] and [19] analyze and provide specific solutions on this problem.

In recent years, there have been a small number of survey studies either focusing directly on congestion control approaches for wireless sensor networks [20]–[22], or dealing with congestion control as part of transport protocols [23], [24]. The three survey studies focusing on congestion control approaches are quite limited in content, covering only a very small subset of papers which, in some cases ([20] and [22]), are outdated. Furthermore, the three aforementioned surveys do not provide a critical evaluation of any of the presented approaches, avoiding to address and discuss the strengths and weaknesses of each approach. However, it is worth noting that [22], provides a basic classification of the presented approaches based on the flow direction, the loss recovery control and the congestion notification. The other two survey papers describing the basic design criteria and challenges of transport protocols for WSNs, provide guidelines towards controlling (avoiding or mitigating) congestion in WSNs. Also, the papers emphasize on quality of service and reliability. In [24], a quite limited number of existing congestion control approaches are mentioned. On the other hand, [23] covers a larger set of congestion control approaches, while providing differentiation based on congestion detection, congestion notification, and congestion mitigation mechanisms. However, the last two papers are considered outdated since a considerable number of congestion control approaches have proposed over the last few years. Recent efforts like [25] focus on congestion control techniques for constrained environments, while [26] reviews a limited number of congestion control protocols.

The contribution of this paper is as follows. Initially, this paper aims at providing a comprehensive survey of a significant number of congestion control approaches proposed for WSNs. In particular, the paper addresses and discusses the characteristics as well as the strengths and weaknesses of each approach. Furthermore, the paper provides a wide range of classifications among the different congestion control approaches based on the way: a) congestion is detected, b) congestion is mitigated, c) congestion notification is performed, and d) congestion can be avoided. Finally, this work discusses and attempts to provide specific directives to the readers for the design and development of new congestion control algorithms.

The paper is structured as follows: Section II provides a case study that motivates congestion control, while Section III describes the problem of congestion in WSNs, providing information on how and where congestion occurs. Section IV provides insights on the topics of congestion avoidance, congestion mitigation, and reliable transmission and classifies the twenty-eight examined algorithms based on the control scheme employed. Section V presents an additional classification, based on the mechanisms used for detection, notification, mitigation, and avoidance. Section VI identifies common performance evaluation metrics used in the literature, while Section VII provides

Manuscript received January 19, 2013; revised July 6, 2013 and December 26, 2013; accepted March 29, 2014. Date of publication April 24, 2014; date of current version November 18, 2014.

The authors are with the Department of Computer Science, University of Cyprus, Nicosia 1678, Cyprus (e-mail: sergiou@cs.ucy.ac.cy; paul.antoniou@cs.ucy.ac.cy; vasosv@cs.ucy.ac.cy).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/COMST.2014.2320071

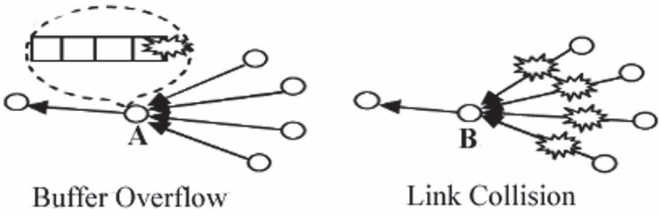

Fig. 1. Typical congestion appearance scenarios in WSNs.

a short review of each of the examined algorithms, highlighting each mechanism's important characteristics. Finally, Section VIII presents a discussion for this survey, while Sections IX and X provide a view of the future directions and the concluding remarks respectively.

II. Motivation

In this section we provide a case study with which we motivate the problem of congestion control in WSNs and then we explain how reliability and fidelity along with other application metrics, like data freshness and availability, are improved because of congestion control.

In particular, in Fig. 1, we illustrate the most representative examples of congestion occurrence in WSNs: *buffer overflow* and *link collision*. Based on the topology and the placement of nodes in WSNs, both types of congestion may occur.

Buffer overflow occurs when a node receives data with a higher data rate than it can transmit (node A in Fig. 1). In this situation it is easy to understand that packet drops will occur. Such an event affects negatively the application, since the throughput is restricted to the maximum data rate of node A. Moreover, as a result of these packet drops, node A and all nodes transmitting to it, waste their power without any benefit to the application, while it is possible to finally exhaust their power and a routing hole to appear at that part of the network. Energy and routing holes can severely affect the ability of the network to perform and can easily reduce its lifetime and overall availability. To overcome this situation a congestion control algorithm can either reduce the data rate of the transmitting nodes in order to cope with the data rate on node A, thus preventing wasting energy due to dropped packets, or it can reroute excess packets through alternative paths, which in turn leads to increased throughput at the sink, often assisting the application have a better idea of the monitored event (fidelity) and a higher reliability to the application due to lower packet loss. In either case, the application will be significantly benefited, either with continuous and on-time delivery of data, or with higher throughput, or by avoiding routing holes that decrease network lifetime.

On the other hand, when link collisions occur, node B in Fig. 1, receives a limited number of packets, even though its neighbor nodes transmit with full data rate. In this case, the sink receives a limited number of packets and the reliability of the application may be affected. In this case a congestion control algorithm that focuses on the MAC layer can be employed to help coordinate the access to the medium in order to avoid collisions. Thus, the throughput of the sink increases, while no energy is wasted due to dropped packets.

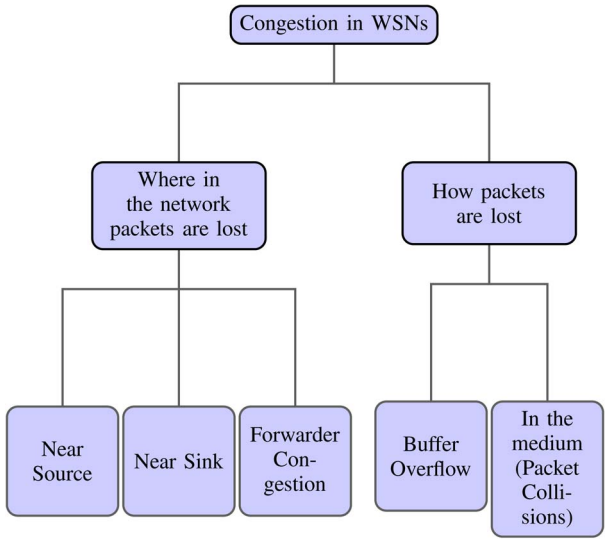

Fig. 2. Congestion in WSNs.

In the next section we provide a more detailed explanation of the causes of congestion in WSNs and extract more detailed conclusions on the network-level metrics affected.

III. Congestion in WSNs

A node in a wireless sensor network (WSN) is a small embedded computing device that interfaces with sensors/actuators and communicates using short-range wireless transmitters. Such nodes act autonomously but cooperatively to form a logical network in which data packets are routed hop-by-hop towards management nodes, typically called sinks or base stations. A WSN comprises a potentially large set of nodes that may be distributed over a wide geographical area, indoor or outdoor. Wireless sensor networks enable numerous sensing and monitoring services in areas of vital importance such as efficient industry production, safety and security at home, and in traffic and environmental monitoring. Traffic patterns in sensor networks can be derived from the physical processes that they sense. Sensor networks typically operate under light load and suddenly become active in response to a detected or monitored event. Depending on the application, this can result in the generation of large, sudden, and correlated impulses of data that must be delivered to a small number of sinks without significantly disrupting the performance (i.e., fidelity) of the sensing application. This high generation rate of data packets is usually uncontrolled and often leads to congestion. In this state, collisions occur in the medium or in case of existence of an effective MAC protocol, the node buffers overflow [11], [16], resulting in random drops of data packets and increased delay. Dropped packets are a major handicap for these networks since they result in severe energy consumption [27]. In the case that no countermeasures are taken, the power of congested nodes can be exhausted leading to the creation of routing "holes" in the network.

Congestion in WSNs can be classified in two major categories concerning how packets are lost and where in the network congestion is taking place [28] (Fig. 2).

A. How Packets Are Lost

1) Packet Collisions in the Medium: In a particular area, many nodes within range of one another attempt to transmit simultaneously, resulting in losses due to interference and thereby reducing throughput of all nodes in the area. We note that explicit local synchronization among neighboring nodes can reduce this type of loss, but cannot eliminate it completely because non-neighboring nodes can still interfere with the transmission.

2) Packet Drops Due to Buffer Overflow: Within a particular node, the queue, or buffer, used to hold packets to be transmitted, overflows. This is the conventional definition of congestion, widely used in wired networks. In this case, nodes receive packets with a higher rate that they can transmit.

B. Where Packets Are Lost

1) Hotspot Near Source—Source Congestion: Densely deployed sensors generating data packets during a critical event will create hotspots very close to the sources (e.g., within one or two hops). In this case, localized, fast time-scale mechanisms capable of providing backpressure messages from the points of congestion back to the sources would be effective for immediate traffic control until the congestion is alleviated by other means. Also local de-synchronization of sources and resource provisioning techniques (resource control) would be effective too.

2) Hotspot Near the Sink-Sink Congestion: Even sparsely deployed sensors that generate data at low data rates can create hotspots in the sensor field, but likely farther from the sources, near the sink. Fast time-scale resolution of localized hotspots using a combination of localized back-pressure and packet dropping techniques would be more effective in this case. Source nodes may not be involved in the backpressure because of the transient nature of the problem in this situation. Also an effective way of alleviating sink congestion is to deploy multiple sinks that are uniformly scattered across the sensor field and, therefore, balance the traffic between these sinks.

3) Forwarder Congestion: A sensor network will have more than one flow (sink-source pair), and these flows will intersect with one another. The area around the intersection will likely become a hot spot. In a tree-like communication paradigm, every intermediate node in the tree can suffer from forwarder congestion. Compared to the other congestion locations, forwarder congestion is far more challenging, because it is very difficult to predict the intersection points due to the network dynamics. In this case, even sparsely deployed sensors generating data will create both transient and persistent hotspots distributed throughout the sensor field. A combination of fast time scale actions to resolve localized transient hotspots, and closed loop rate regulation of all sources that contribute toward creating persistent hot spots seems to be effective. Resource control techniques could be used when traffic control methods cannot meet the application's requirements.

IV. Control Schemes

Generally, algorithms that deal with congestion in WSNs can be initially classified in three major categories. These are: Congestion Control, Congestion Avoidance, and Reliable Data Transmission (Fig. 3). Although there are not clear and explicit boundaries between these three categories, we attempt a first classification of algorithms based on this set. In this work congestion mitigation algorithms are considered the algorithms that take reactive actions when congestion arises in the network and their target is to control it. These algorithms normally involve MAC and network layer operations, and in some cases they also use transport layer actions.

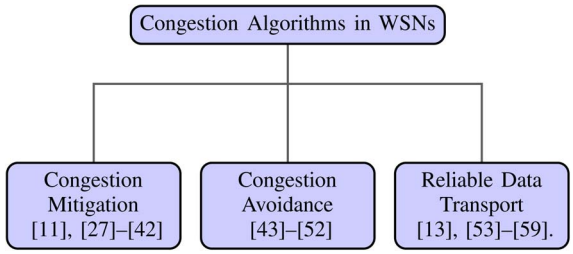

Fig. 3. Initial classification of congestion algorithms in WSNs.

Congestion avoidance algorithms are considered as the algorithms that take actions in order to prevent congestion from happening. These algorithms normally involve MAC and network layer operations.

On the other hand, reliable data transmission algorithms are the algorithms that, besides their effort to control congestion in a network, also attempt to recover all or part of the lost information. These algorithms normally apply when all information is critical for the application and usually involve transport layer mechanisms.

Generally, the presence of congestion means that the load is (temporarily) greater than the network resources can handle in such way that resources become depleted. In such a case the following control schemes may be used: control the load (traffic control), increase the resources (resource control), or employ MAC layer enhancements. MAC layer enhancements could help more in the direction of interference-based congestion (packets collision in the medium). If the packet generation rate is sufficiently small, simultaneous transmission of packets becomes independent of the rate. Rather, it depends on the time at which each node generates the packet. A good way to reduce this type of congestion is to perform phase shifting an observation made by authors in [11]. Small amounts of phase shifting can be performed by introducing slight jitters at the data-link layer. In [11] the application layer itself also introduces phase shifts. While jittering at the data-link layer aims to cause small transmission variations between neighboring nodes, we think, that phase shifting at a higher layer can be achieved on a larger time scale. To handle buffer based congestion (packet drops due to buffer overflow) one may employ the other two methods, a) traffic control or b) resource control as these would help in emptying the buffers of intermediates sensor nodes. It is possible to have more than one types of congestion occurring at the same time.

V. Classification of Algorithms

In this section we further classify the three major categories of congestion algorithms in WSNs as presented in Fig. 3 using additional attributes.

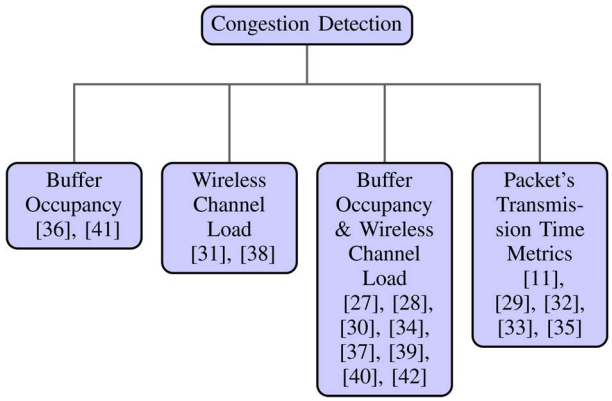

Fig. 4. **Congestion mitigation algorithms**: Detection mechanisms.

A. Congestion Mitigation Algorithms

Algorithms that deal with congestion mitigation can be classified on the way they detect congestion, the way they notify the other nodes for this incident, as well as the way they face congestion (counteractions mechanisms).

1) Congestion Detection: Currently, there are four ways that algorithms use to detect congestion (Fig. 4).

These are the following:

- **Buffer occupancy**: The algorithms that use this way for detecting congestion assume that there is an effective MAC protocol, able to avoid packet collisions in the medium, or they assume that more than one nodes that are not in the range of each other, transmit packets with a small time shift to a receiving node. In this case, congestion is measured through the increment of queue length in nodes. Algorithms that employ this method like [36] and [41], act proactively by inferring congestion when the buffer occupancy exceeds a certain percentage.
- **Wireless Channel Load**: Algorithms that employ this method for congestion detection, only measure the packet load in the medium and take actions when the time frame for the transmission of a single packets, exceeds some predefined thresholds.
- **Buffer Occupancy and Wireless Channel Load**: With this method, congestion is detected either at the medium or in the buffer. In this category we also insert algorithms that form clusters and measure congestion through traffic intensity.
- **Packet Transmission Time Metrics**: Algorithms that employ this method to detect congestion use packet service time and packet inter-arrival time (or a combination of them) to detect congestion. Specifically, algorithms like [33] and [35] count the packet service time and packet interarrival time and if it is beyond a limit they infer that congestion is imminent.

Judging the above mentioned methods we can safely state that each one presents advantages and disadvantages. Buffer occupancy is a very simple method, which can be easily implemented while it does not require much resources from nodes. The disadvantage of this method, relies on its dependence on the MAC protocol. If the MAC protocol is not efficient it is possible for collisions to exist in the medium (if more than one nodes transmit concurrently packets to specific nodes) and for the buffers to receive a limited number of packets. In this case the network is not possible to detect the hotspots and inevitably problems will be created.

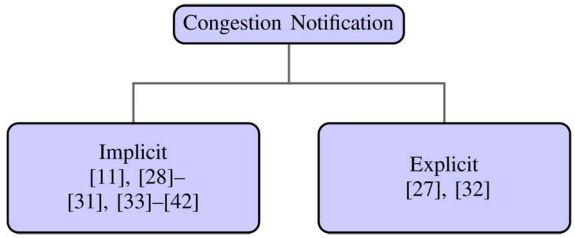

Fig. 5. **Congestion mitigation algorithms**: Notification methods.

By adjusting the wireless channel load we can efficiently tackle the problem of medium collisions but we cannot react if buffers are fully occupied and start dropping packets.

Packet transmission time metrics, although it is considered the most efficient method it is heavily depended on the application in use. This means that it is possible to have packet drops due to other reasons e.g., environment or physical causes and the network to detect and trigger congestion control actions as a result.

Buffer occupancy and wireless channel load is, according to our opinion, the most efficient way for congestion detection. It captures congestion either in the wireless medium or in the node buffers. It can be easily implemented and it is, relatively, low power consuming. It is the method that it has been adopted by the majority of congestion mitigation protocols.

2) Congestion Notification: A second classification can be the method used by the algorithms in order to notify the rest of the network about congestion events. This can be either explicit or implicit (Fig. 5).

- **Explicit**: Using explicit congestion notification, additional control packets are broadcast by congested nodes to the rest of the nodes in order to inform them about their congestion state. This method has been used by the first congestion algorithms like [27] and [32]. Since then, it has been proven that transmitting extra control packets when congestion has occurred adds significant load to the already congested environment. Therefore, explicit congestion signaling has not been adopted by subsequent congestion control protocols.
- **Implicit**: Implicit congestion notification is the method that has been employed by the vast majority of subsequent congestion control protocols. Specifically, congestion information is propagated to the rest of the network by overhearing the data packets that are being transmitted. If congestion is detected, the information is piggybacked in a data packet header or in some cases in ACK packets. This technique avoids the unnecessary injection of extra packets to the already heavily loaded part of the network.

In some cases [11], [30], [34], [53], nodes use binary feedback for congestion notification, i.e. a single bit is added to each packet indicating whether there is congestion or not, in order to notify source nodes to decrease or decrease traffic rate respectively. In other approaches [27], [32], nodes use more

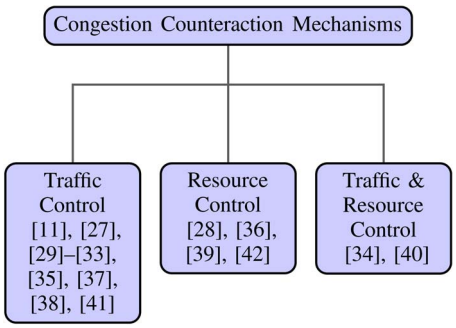

Fig. 6. **Congestion mitigation algorithms**: Counteraction mechanisms.

sophisticated congestion notification approaches, i.e., rather than just generating a binary feedback, generate a linear feedback. In this case, nodes can proactively monitor network statistics (channel load [27], queue length [27], [32]). This information is propagated in the network and source nodes can increase or decrease its traffic rate depending on how far the reported channel load or the reported buffer length is the from a specified threshold.

A representative example of a congestion control algorithm that employs linear feedback for congestion notification is "Interference-Aware Fair Rate Control" (IFRC) [60] algorithm. This algorithm measures congestion levels through an exponentially weighted moving average of the instantaneous queue length and if this average exceeds a certain upper threshold, the node is said to be congested. Then the node halves its current data rate and then starts additively increasing it. The average queue length is updated whenever a packet is inserted into the queue.

3) Congestion Counteraction Mechanisms: Finally, a classification can be performed regarding the reaction of algorithms in their effort to mitigate the effects of congestion. Congestion is mitigated either by rate reduction (*traffic control*) or by the creation of alternative paths from the source(s) to the sink(s) for forwarding the excess data packets (*resource control*) (Fig. 6).

- **Traffic Control**: *Traffic control* mechanisms are concerned with measures taken in order for a network to operate at an acceptable performance level, when resource demands are near, or exceed, the capacity of network. Traffic control can be seen as a means of taking traffic reducing steps, such as reducing the amount of packets injected into the network, to alleviate congestion. Traffic control may follow a window-based, or a rate-based approach.

  In the windows-based approach, a sender probes for the available network bandwidth by slowly increasing a congestion window (used to control how much data is outstanding in the network); when congestion is detected (indicated by the loss of one or more packets), the protocol reduces the congestion window greatly. The rapid reduction of the window size in response to congestion is essential to avoid network collapse. One of the most popular window-based approaches for adjusting sending rates is the Additive Increase Multiplicative Decrease (AIMD) policy which involves binary feedback control messages. A number of congestion control approaches in wireless sensor network like [11], [27], [32], and [34], are based on the AIMD policy.

  The advantage of the AIMD policy is that such a protocol is agnostic to the underlying link layer, requiring no prior knowledge of the available capacity. However, the AIMD policy is shown to provide unsatisfactory performance in wireless environments where high packet loss rates are often attributed to the time-varying conditions of the wireless channel, e.g., interference, multi-path fading, etc. Therefore, the resulting saw-tooth rate behavior may violate the QoS requirements (e.g., fidelity of the reported events).

  Rate-based approaches attempt to estimate the available network bandwidth explicitly. This can be achieved, for example, by using a throughput formula, or empirically derived directives. In [41], a deterministic population balance equation inspired from biological systems is used as a throughput formula to adjust the sending rate of sensor nodes. In [61], a fuzzy logic based approach was proposed to combine a set of equations and rules to evaluate the traffic rates at source nodes. Other efforts employ linear approaches. In particular, in [62] each node calculates the source rate based on the aggregate price of capacity and energy along its path to the sink, and constantly updates the data rate both based on the capacity and energy of the passing flows as well as from a feedback from the sink. Furthermore, rate adjustments can be performed on the basis of empirically derived regions of operation [53].

  One of the advantages of traffic control method is that the burden of congestion alleviation lies, in most of the cases, to only one node, the source node. Also, when traffic reduction is applied, congestion can be alleviated relatively quickly since the load in the network decreases. On the other hand, traffic control is not efficient for event-based networks, where the network becomes active when sensor nodes are triggered by an event. In this case traffic reduction can jeopardize the network's mission since all data packets carry valuable information about the event. In addition, traffic control can not be efficiently used during transient congestion phenomena caused by aperiodic and short term packet bursts due to the slowness of traffic control mechanisms (e.g., rate reduction) to react.
- **Resource Control**: To eliminate the disadvantages of the traffic control method, an alternative method, called *resource control* has been proposed [28], [36], [39], [42]. In this method, when the network is congested (either in the medium or in the buffers), data packets follow alternative paths, which are not congested, in order to be forwarded to the sink. This method has the advantage that traffic control is avoided and all data packets have a greater potential to reach the sink. On the other hand, special care needs to be taken in order to meet the performance requirements like packet travel time, avoidance of loops etc.
- **Traffic and Resource Control**: Some algorithms employ both methods in their effort to face congestion. This way is actually a hybrid method that attempts to trade on the advantages of both methods.

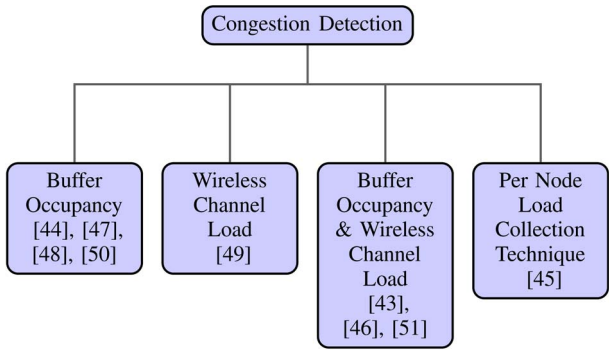

Fig. 7. **Congestion avoidance algorithms**: Detection mechanisms.

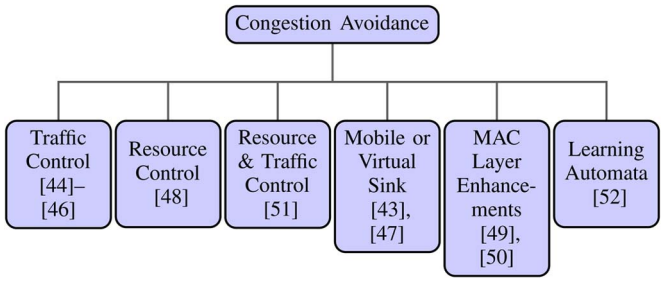

Fig. 8. **Congestion avoidance algorithms**: Avoidance mechanisms.

Choosing the best counteraction method for congestion is not a trivial effort. According to our opinion the choice is application-dependent. The traffic control method better applies to transient congestion situations or in applications where reducing the rate with which sources are injecting data in the network is acceptable. The Resource control method better applies in applications where all data need to be transferred to the sink and in cases where a more sustained congestion situation is expected. A vital requirement for the successful operation of the resource control method is the existence of dense and redundant placement of nodes. Such placements can provide the required alternative paths in order to avoid the congested hotspots. Choosing both methods in a single scheme is an idea that has been adopted in [40] and seems to be promising.

4) Further Attributes: Congestion Control algorithms can also be classified in other parameters like the following:

- Traffic Direction, if it is upstream or downstream.
- Transport of packets, if it is Hop-by-Hop or End-to-End.
- Whether it supports fairness among the nodes.
- Whether it supports multiple classes (e.g., high priority or low priority packets).
- Whether it states clearly or it is evident from any experimentation results that it conserves energy.

B. Congestion Avoidance

Congestion avoidance algorithms can be classified on the way they detect that congestion is going to happen and on the mechanism they use to avoid congestion.

1) Congestion Detection: Similar to congestion mitigation algorithms, congestion avoidance algorithms employ respective methods in order to detect congestion, with the difference that they act in a preventive way instead of a reactive way. These are buffer occupancy, wireless channel load, or both of them as well as the "per node load collection technique" as it appeared in [45] (Fig. 7).

2) Congestion Avoidance Mechanisms: Congestion can be avoided using similar techniques as with congestion mitigation algorithms (traffic, resource control or both) with the difference that also in this case congestion avoidance algorithms act in a preventive way instead of a reactive way. Other techniques have also been also been introduced. All methods are presented in Fig. 8.

- **Mobile or Virtual Sink**: In these cases the sink is placed near the regions that show signs of high load. In the case of mobile sink [47] the network is split into clusters and there, an in-network storage model is introduced. Cluster heads collect all load from their nodes and transmit them to the mobile sink when it passes near the cluster. On the other hand, [43], introduced the concept of mini-sinks where some nodes with longer communication range, tunnel the traffic event from regions that are going to become congested.
- **MAC Layer Enhancements**: MAC layer enhancements are used in the MAC layer in order to avoid collisions in the medium. Implementations span from priorities in heavy loaded nodes [29], [50] to transmission phase shifting [11].
- **Learning automata**: In this case code capable of taking intelligent actions (called automata) is developed at each of the network's nodes that are capable of controlling the rate of flow of data at the intermediate nodes based on probabilistically how many packets are likely to get dropped if a particular flow rate is maintained.

Concerning congestion avoidance mechanisms we have the same comments as for congestion mitigation algorithms. Concerning the rest of the methods we believe that mobile or virtual sink is a promising idea but more effort is needed in order to be implemented in real scenarios. MAC layer enhancement is a very efficient and effective method for collision avoidance. Finally concerning learning automata we can not be so fair in our judgement since we have seen it in just one paper [52].

C. Reliable Data Transport

Reliable data transport can be considered as a different category of transport layer protocol that focus in the reliable data transmission of information packets. But since they also provide congestion control they can be also considered as part of this work.

Reliable data transport protocols can be divided based on three basic attributes. These are traffic direction (Fig. 9), if they provide end-to-end or hop-by-hop reliability (Fig. 10), as well as on which parameter the reliability focuses on (Fig. 11).

In a Table I a synoptic classification for congestion control algorithms is provided, in Table II a synoptic classification for congestion avoidance algorithms is provided, while Table III refers to the classification of reliable data transport protocols.

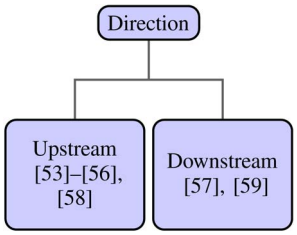

Fig. 9. **Reliable data transport protocols**: Direction.

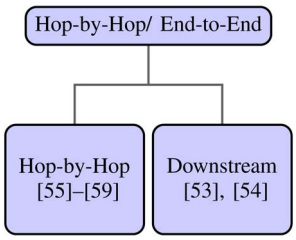

Fig. 10. **Reliable data transport protocols**: Range.

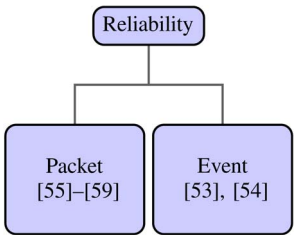

Fig. 11. **Reliable data transport protocols**: Reliability.

VI. Common Congestion Control Metrics

In this section we present the most common metrics that the majority of algorithms use in order to evaluate the performance of their efforts.

- **Packet Delivery Ratio**: This metric is used in order to measure the efficiency of the algorithm concerning the delivery of packets to the sink. Packet drops are normally measured as the percentage of the total packets that are received by sinks divided by the number of packets that are produced by sources. The more closer to 100% the result is, the more efficient the algorithm.
- **Throughput**: Throughput is defined as the number of packets per unit time that is received by the sink. The higher the value of throughput is, the more efficient is the algorithm.
- **End-to-End Delay**: This metric is used in order to measure the time that is required for a packet to reach the sink. This metric is an indication of the efficiency of the algorithm, to quickly mitigate or avoid congestion. The shortest the time is, the better the algorithm's performance is, since delay normally occurs in congested hotspots due to retransmissions or due to long routes (in the cases where a resource control algorithm is employed).
- **Hop-by-hop Delay**: Hop-by-hop delay is also a metric that measures the efficiency of the algorithm in terms of congestion and overhead, since when congestion is avoided, high queueing delays are also avoided.
- **Network Lifetime**: This metric reflects the long-term energy efficiency of the network. If the power of the nodes is exhausted uniformly then this value increases. This metric is usually high in resource control algorithms.
- **Average Node Energy Consumption**: This metric indicates the energy consumption of nodes. The value of this metric should be kept low in order to indicate an energy-efficient congestion control algorithm.

VII. Short Review of Algorithms

There are several algorithms that have been proposed in the literature, which attempt to resolve the congestion problem in WSNs. In this section we present and review a representative number of them. The algorithms are classified into congestion control, congestion avoidance, and reliable data transmission algorithms. In this effort we try to match each algorithm to the category that fits it better, since many algorithms fit in more than one categories. Algorithms are sorted within each category, based on the year of publication. Through this classification we attempt to present the evolution of this research field through time.

A. Congestion Control Algorithms

The first actual effort for controlling the traffic in WSNs is the "Adaptive Rate Control" [11] algorithm. Although this algorithm does not refer directly to congestion control, it can be considered as so, since it controls the network's data rate in order to guarantee fairness and subsequently to avoid overload situations.

ARC: Woo *et al.* proposed the Adaptive Rate Control (ARC) scheme in 2001 [11]. ARC does not involve any congestion detection or notification mechanisms. ARC uses an AIMD-like traffic control scheme to mitigate congestion, which works as follows: an intermediate node increases its sending rate by a constant a if it overhears a successful packet forwarding by its parent node. Otherwise, the intermediate node multiplies its sending rate by a factor b, where $0 < b < 1$. ARC maintains two independent sets of a and b, for source traffic and transit traffic respectively, in order to guarantee fairness. ARC has been evaluated on a 11-node star topology, a 11-node tree topology and a real testbed. Simulation results have shown that ARC is effective in achieving fairness while maintaining good aggregate bandwidth with reasonable energy efficiency, especially low traffic situations that are the common case in sensor networks. However, ARC was not compared to other related approaches.

CODA: One of the first algorithms in literature that refer directly to congestion control and avoidance in WSNs is the Congestion Detection and Avoidance algorithm (CODA) [27]. This algorithm constitutes the base for this research field and it is one of the most cited algorithms. CODA attempts to face congestion by implementing three mechanisms: Congestion detection, open-loop hop-by hop backpressure notification, and closed-loop multi-source AIMD-like traffic control to mitigate congestion. Congestion detection is of prime importance in CODA. It is used to detect whether there is congestion in an

TABLE I
CONGESTION CONTROL ALGORITHMS

Protocol/ Mechanism	Congestion Detection	Congestion Notification	Congestion Mitigation	Traffic Direction	Fairness	Energy Conservation
ARC [11]	The event if the packets are successfully forwarded or not	Implicit	Traffic Control	Source to Sink	Yes	Good
CODA [27]	Buffer occupancy and wireless channel load	Explicit	Traffic Control	Source to Sink	No	Good
CCF [29]	Packet Service Time	Implicit	Traffic Control	Source to Sink	Yes	Good
Fussion [30]	Buffer Occupancy and Wireless Channel load	Implicit	Traffic Control	Source to Sink	No	Good
CONSISE [31]	Wireless Channel Load	Implicit	Traffic Control	Sink to Sensor	Yes	Good
COMUT [32]	Cluster/ Traffic Intensity Estimation	Explicit	Priority Based Traffic Control	Source to Sink (Cluster-by-Cluster)	No	Good
Sen- TCP [33]	Buffer Occupancy/ Packet inter-arrival Time	Implicit	Traffic Control	Source to Sink	No	Good
BGR [34]	Buffer Occupancy and Wireless Channel Load	Implicit	Resource and Traffic Control	Source to Sink	Yes	N/A
TARA [28]	Buffer Occupancy and Wireless Channel load	Explicit	Recourse Control	Source to Sink	Yes	Good
PCCP [35]	Packet Interarrival Time/ Packet Service Time	Implicit	Priority Based Traffic Control	Source to Sink	Yes	N/A
RCRT [55]	Sink decides based on ”time to recover loss”	Implicit	Sink Based Traffic Control	Sink to Source	No	N/A
HTAP [36]	Buffer Occupancy	Implicit	Recourse Control	Source to Sink	No	Good
CONSISE [31]	Wireless Channel load	Implicit	Traffic Control	Sink to Source	No	Good
UHCC [64]	Buffer Occupancy and Wireless Channel load	Implicit	Traffic Control	Source to Sink	Yes	Good
FACC [38]	Wireless Channel load	Implicit	Traffic Control	Source to Sink	Yes	Good
CADA [40]	Buffer Occupancy and Wireless Channel load	Implicit	Recourse and Traffic Control	Source to Sink	No	Good
DAIPaS [42]	Buffer Occupancy and Wireless Channel load	Implicit	Resource Control	Source to Sink	Yes	Good

area in the network in order to activate the rest of the mechanisms. Congestion detection is performed using the present and past channel loading conditions, along with the current buffer occupancy. Due to the high energy consumption of persistent channel listening, CODA employs a sampling scheme that activates channel monitoring when it is needed. Once congestion is detected, the nodes notify explicitly their upstream neighbor nodes via a backpressure mechanism. Backpressure signals are propagated towards the source and the nodes that receive the signals must take decisions based on their local congestion policy, in order to reduce the traffic in the network. Decisions concern rate reduction, packet drops etc. Also each node that

TABLE II
CONGESTION AVOIDANCE ALGORITHMS

Protocol/ Mechanism	Congestion Detection	Congestion Avoidance Mechanism
Siphon [43]	Buffer Occupancy, Wireless Channel load and Sink Decides	Traffic redirection through Virtual Sinks
Light- Weight Buffer Management [44]	Buffer occupancy	Traffic Control
CoSMoS [45]	Per node load collection technique	Rate-based control
Buffer and Rate Control Based Congestion Avoidance [46]	Buffer Occupancy and Wireless Channel load	Traffic Control
CAEE [47]	Buffer Occupancy	Load Collection through a Mobile Sink
TADR [48]	Buffer Occupancy	Resource Control
ANAR [49]	MAC layer load	Traffic redirection through alternative paths
Priority Based Medium Access Protocol [50]	Buffer Occupancy	Most Loaded Nodes get higher priority to Medium
TALONet [51]	Buffer Occupancy and Wireless Channel load	Resource and Traffic Control
LACAS [52]	N/A	Learning Automata adjust flows Rate
Flock-CC [39]	Buffer Occupancy and Wireless Channel load	Resource Control
LVCC [41]	Buffer Occupancy	Traffic Control

TABLE III
RELIABLE DATA TRANSPORT MECHANISMS

Protocol/ Mechanism	Hop- by- Hop /End- to -End	Traffic Direction	Reliability
PSFQ [57]	Hop-by-Hop	Downstream	Packet
ESRT [53]	End-to-End	Upstream	Event
Directed Diffusion [13]	Hop-by-Hop	Upstream	Event
RMST [58]	Hop-by-Hop	Upstream	Packet
GARUDA [59]	Hop-by-Hop	Downstream	Packet and Destination Related
STCP [54]	End-to-End	Upstream	Event and Packet
RCRT [55]	Hop-by-Hop	Upstream	Packet
EDCCP [56]	Hop-by-Hop	Upstream	Packet

receives a backpressure message decides whether to further propagate it, based on its congestion condition. Finally, in order to control congestion when multiple sources are transmitting to a single sink, CODA implements a closed-loop, multi-source AIMD-like traffic regulation. In this case when a source's event rate is higher than a pre-specified threshold, which is always a fraction of the maximum theoretical throughput of the channel, it requires an ACK packet from the sink in order to maintain its rate. In cases of lost ACK packets, the source reduces its sending rate. CODA has been tested both in ns-2 [63] and in a real testbed. Simulations conducted in ns-2 took into account both randomly generated topologies ranging from 30 to 120 nodes, while a tree-based topology was used in the real testbed. Results showed that CODA significantly improves the performance of data dissemination applications such as Directed Diffusion [13], by mitigating hotspots and reducing the energy tax with low fidelity penalty on sensing applications. However, the AIMD-like traffic regulation may not be very effective in WSNs because it results in a saw-tooth rate behavior that may violate the QoS requirements (e.g., fidelity of the reported events). Furthermore, the end-to-end nature of the closed-loop mechanism may result in reduced responsiveness causing increased latency and high error rates, especially during long periods of congestion. Also backpressure signals and ACK control messages consume additional energy and bandwidth. CODA was not compared against other related approaches.

Following CODA, the notion of fairness began to gain momentum. Moreover, algorithms shifted from explicit congestion notification to implicit. Also cross layer techniques appeared in literature.

CCF: Ee *et al.* proposed a distributed and scalable mechanism for congestion control and fairness (CCF) [29] for many-to-one routing in WSNs. CCF provides congestion detection on the basis of packets service time, and congestion mitigation through traffic control. CCF controls congestion in a hop-by-hop manner and each node uses exact rate adjustment based on its available service rate and child node number. In particular, CCF assumes a tree routing structure having the sink acting as a root and all data sources as leaves. Each sensor receives and forwards packets from its upstream neighbors; each upstream neighbor is the root of an upstream sub-tree. The sensor learns the number of data sources in each of those upstream sub-trees, measures its own downstream forwarding rate, computes per-source fair rate, which is propagated upstream such that the data sources do not send packets beyond the rate. There are two alternative scenarios that this algorithm could be applied: a) all nodes are generating data and routing them to the sink and b) most nodes in the network are silent, only the nodes that detect an event generate data. Such routing structures often

result in the sensors closer to the base station experiencing congestion, which inevitably cause packets originating from sensors further away from the base station to have a higher probability of being dropped. The basic concept for controlling congestion consists of the following steps that repeatedly run at each sensor node: a) Measure the average rate r at which packets can be sent from this mote, b) divide the rate r among the number of children motes downstream n, to give the per-node data packet generation rate $rdata = r/n$, adjust the rate if queues are overflowing or about to overflow, and c) compare the rate $rdata$ with the rate rdata.parent sent from the parent and use and propagate the smaller rate downstream. CCF was tested using both simulations (a randomly generated topology of 116 nodes, with a per-node maximum degree of 5 and a maximum network depth of 6) and actual implementation in UC Berkeley's sensor motes (a network of 10 motes). CCF was shown to achieve simple fairness, where each node was able to receive almost the same throughput. However, in applications where different sensors (e.g., geographically deployed in different places) need to gain different throughput (i.e., priority-dependent throughput), CCF will not be applicable. Also, the rate adjustment in CCF relies only on packet service time, something which could lead to low utilization when some nodes do not have enough traffic or the packet error rate is high.

Fusion: Hull *et al.* [30] proposed a scheme called 'Fusion' for mitigating congestion control in WSNs. In general, Fusion detects congestion by monitoring the queue size of each node and performing channel sampling at fixed intervals. In the presence of congestion, Fusion provides implicit congestion notification to all nodes in a radio neighborhood by setting a congestion bit in the header of every outgoing packet. Congestion is mitigated on the basis of traffic control. More specifically, Fusion combines three congestion control techniques that operate at different layers: a) hop-by-hop flow control, b) source rate limiting scheme, and c) prioritized MAC. In hop-by-hop flow control, each sensor sets a congestion bit in the header of every transmitted packet. Using the broadcast characteristic of the wireless medium, every packet provides congestion feedback to all nodes in a radio neighborhood with every transmission. Thus, there is no need for explicit control messages, which waste a large portion of the already limited bandwidth. Hop-by-hop flow control consists of two components: congestion detection and congestion mitigation. Congestion detection is done based on the sensor's queue size. If the sensor's queue space gets beyond a specified limit, a congestion bit is set. Otherwise the congestion bit is removed. Congestion mitigation is a way to control the nodes' transmission rate in order to prevent queues at their next-hop node from overflowing. When a sensor overhears to a packet with the congestion bit set, it stops forwarding data. Otherwise, the congestion would grow bigger, and eventually the whole network will collapse. Rate limiting is a way to limit the sending rate of a sensor. Each sensor listens to the traffic its parent forwards, to estimate N, the total number of unique sources routing through the parent. A token bucket scheme is used to regulate each sensor's send rate. A sensor accumulates one token every time it hears its parent forward N packets, up to a maximum number of tokens. The sensor is allowed to send only when its token count is above zero and each transmission costs one token. In the prioritized MAC mechanism, the MAC layer provides assistance to sensors in order to react fast to congestion. A standard CSMA MAC layer is used, with a modification that implements a prioritization scheme. According to this scheme, congested nodes have higher priority compared to the other nodes. Specifically, if a sensor is congested its back-off window is the one-fourth the size of a non-congested sensor's back off window, allowing queues to drain more quickly and increasing the likelihood congestion control information will propagate throughout a sensor's neighborhood. The performance of Fusion was evaluated in an indoor testbed of 55 Crossbow Mica2 nodes using both event-based and periodic data traffic. Fusion claims to achieve good throughput and fairness at high offered loads. However, the rate adjustment used in Fusion is not smooth, something which may affect link utilization and fairness. Also the frequent use of the wireless radio for channel probing leads to energy wastage. Fusion was not compared against other related approaches.

COMUT: Karenos *et al.* proposed COMUT [32], a cluster-based congestion control mechanism for supporting multiple classes of traffic in WSNs. In COMUT, each cluster node detects congestion through traffic intensity estimation. This estimation is broadcast to the cluster head, which evaluates the congestion level. Congestion is mitigated on the basis of an AIMD-like traffic regulation. More specifically, COMUT consists of three different parts. Cluster formation, traffic intensity estimation, and rate regulation. In the cluster formation procedure sensors are organized into clusters. In each cluster, a cluster head called sentinel is elected. COMUT employs ZRP (Zone Routing Protocol) to assist in the formation of clusters. After the cluster is formed the level of congestion of each cluster must be estimated. To perform this estimation COMUT calculates the traffic intensity within and across multiple clusters. Due to the fact that traffic intensity in highly affected by the number of incoming and existing flows, COMUT involves a queuing network where each sensor is modeled as a queue. Once traffic intensity is calculated, congestion is controlled through AIMD-like source rate adjustment. The congested cluster through its sentinel node informs the other sentinels, and through them the source, for its condition. COMUT controls congestion through rate reduction and takes into account multiple classes of traffic. Thus, the sending rate of the low importance flow is dropped to a minimum if packets with higher importance exist along the congested path. The performance of COMUT was evaluated through ns-2 [63] simulation tool, using randomly generated topologies of 60–140 nodes. Simulation results show that COMUT is highly successful in abating congestion and in reducing wasteful packet drops, achieving energy savings. Packets from flows of high importance were delivered with extremely high fidelity. Researchers showed that congestion is controlled and all important flows can be admitted and delivered with minimum drops, achieving energy savings. COMUT was not compared against other related approaches.

SenTCP: Wang *et al.* proposed SenTCP [33]. SenTCP is an open-loop hop-by-hop congestion control protocol with two special features. Firstly, it jointly uses average local packet service time and average local packet inter-arrival time in order to

detect congestion by estimating the current local congestion degree in each intermediate sensor node. The use of packet arrival time and service time not only precisely calculates congestion degree, but effectively helps to differentiate the reason of packet loss occurrence in wireless environments, since arrival time (or service time) may become small (or large) if congestion occurs. Secondly, SenTCP uses hop-by-hop congestion control. In SenTCP, each intermediate sensor node will issue feedback signal backward and hop-by-hop. The feedback signal, which carries local congestion degree and the buffer occupancy ratio, is used for the neighboring sensor nodes to adjust their sending rate in the transport layer. The use of hop-by-hop feedback control can remove congestion quickly and reduce packet dropping, which in turn conserves energy. SenTCP was tested on a simulated simple linear-like topology of 20 source nodes and was compared to TCP. Simulation results and comparison with TCP showed that SenTCP can reduce packet dropping resulted from buffer overflow and in-turn energy would be conserved. SenTCP also effectively overcame the problem of differentiating congestion and packet error loss. The throughput of SenTCP was almost not influenced by packet error probability.

Another evolution to the subject is the shift to network layer using routing techniques in order to mitigate congestion.

BGR: Popa *et al.* proposed Biased Geographical Routing (BGR) [34] protocol to reactively split traffic when congestion is detected. BGR detects congestion based on buffer occupancy and wireless usage, exponentially averaged to eliminate noise. Wireless usage is measured by periodically sampling wireless medium. Congestion notification is performed implicitly, on the basis of a single congestion bit added to each packet. Thus, each node that promiscuously listens to the packets sent by its neighbors, is able to detect their congested status. Congestion is mitigated using two algorithms, namely: In-Network Packet Scatter (IPS) and End-to-End Packet Scatter (EPS). IPS alleviates transient congestion by splitting traffic immediately before the congested areas. In contrast, EPS alleviates long term congestion by splitting the flow at the source, and performing rate control on the basis of the AIMD strategy. EPS selects the paths dynamically, and uses a less aggressive congestion control mechanism on non-greedy paths to improve energy efficiency. The 'bias' used in BGR determines how far the trajectory of splitting traffic will deviate from greedy route (which is always the shortest path). BGR was tested on a random topology of 400 nodes simulated in ns-2. Results showed that BGR works well for flows where the distance between the source and the destination is large enough to allow the use of non-interfering multiple paths. For short-range flows, where multiple paths could not be used, the throughput obtained by BGR is smaller with at most 14%, as the short-range flows interfere with split flows of long-range communications. However, by increasing long-range flows throughput fairness among the different flows was improved. On the other hand, it is worth noting that because the bias is randomly chosen, BGR likely makes congestion worse under some situations. In addition, BGR needs node location information provided by either GPS or other coordinate system. This overhead is non negligible. Also, the AIMD strategy is not very effective in WSNs because it provokes a saw-tooth rate behavior that may violate the QoS requirements. In addition, AIMD-like mechanisms take a long time for data rates to converge in low-rate wireless links.

Later on, the early efforts to introduce resource control in order to mitigate congestion [28], [36] appeared. Also cross layer optimizations and fairness continue to gain momentum.

TARA: Kang *et al.* proposed the Topology Aware Resource Adaptation (TARA) protocol [28]. TARA focuses on the adaptation of the network's extra recourses in case of congestion, alleviating intersection hot spots. A graph-coloring problem is used to determine the needed topology for the resource adaptation strategy. TARA measures not only the buffer occupancy but also the channel loading in order to detect congestion. As soon as the congestion level hits the upper watermark, it declares congestion and becomes a hot spot node. At this point, the hot spot node needs to quickly locate two important nodes: the distributor and the merger. Then, a detour path can be established, starting at the distributor and ending at the merger. As suggested by their names, the distributor distributes the incoming traffic between the original path and the detour path, whereas the merger merges these two flows. Thus, in the case of congestion and the creation of hot-spot, traffic is deflected from the hot-spot through the distributor node along the detour and reaches the merge node, where the flows are merged. As soon as congestion has been alleviated the network stops using the detour path. For quick adaptation, the distributor node keeps in its memory which neighbor is on the original path. The performance of TARA was evaluated on a random topology of 81 nodes simulated in ns-2. Detailed simulation results have shown that TARA can energy efficiently absorb incoming traffic load. The results have also demonstrated that TARA performs very close to an ideal off-line resource control algorithm in terms of both fidelity satisfaction and energy conservation. TARA was compared against five strategies for congestion control, which do not correspond to protocols or mechanisms found in nature. Results showed that TARA outperformed all these strategies. However, it is worth pointing out that TARA requires knowledge about the whole network topology, which makes the protocol impractical for large scale networks.

PCCP: Wang *et al.* proposed a hop-by-hop node priority-based upstream congestion control protocol for WSNs [35], [64]. PCCP refutes the congestion control protocols that argue in favor of providing equal fairness to each sensor node in a multi-hop WSN (e.g., CCF [29]) by attaching a weighted fairness to each sensor node. PCCP offers different degrees of priority indexes such that a sensor node with a higher priority index enjoys a higher bandwidth and also sensor nodes that inject more traffic get more bandwidth. PCCP further defines the priority index for both self generating traffic and transit traffic, based on which the queue length for source and the transit traffic is allocated. PCCP infers the degree of congestion through packet inter-arrival time and packet service time and then imposes hop-by-hop congestion control depending on the measured congestion degree and the priority index. PCCP uses implicit congestion notification by piggybacking the congestion information in the header of data packets, thus avoiding additional control packets. PCCP allows the application layer to dynamically override the priority index of any sensor node(s) of

any particular region. This feature might be required by many applications of WSN. PCCP was tested on a small tree-based topology of 7 nodes (using both single-path and multi-path hardwired routing) and a linear topology of 10 to 40 nodes. Simulation series neglected the details of MAC protocols, but assumed that MAC protocols provide even access opportunities for each neighboring node. Simulation results showed that: 1) PCCP achieves high link utilization and flexible fairness; 2) PCCP achieves small buffer size; therefore it can avoid/reduce packet loss and therefore improve energy-efficiency, and provide lower delay. PCCP was compared against the CCF [29] for the case of single path routing. Results showed that CCF achieves lower throughput than PCCP in the interval when a node does not generate sufficient traffic. Researchers claimed that this is because CCF cannot effectively allocate the remaining system capacity and use a work-conservation scheduling algorithm. Also in the presence of packet losses, PCCP was shown to achieve much higher throughput than CCF. The reason was attributed to the fact that CCF only uses packet service time to detect congestion therefore it cannot detect either under-utilized links or nodes. On the other hand, PCCP also uses packet inter-arrival time to detect congestion, thus it is able to detect under-utilized links and nodes.

HTAP: Sergiou *et al.* proposed Hierarchical Tree Alternative Path (HTAP) [36]. HTAP is a scalable and distributed framework for minimizing congestion and assuring reliable data transmissions in event based networks. HTAP is a hop-by-hop algorithm that employs an implicit way for informing the other node for congestion. It mitigated congestion through a resource control technique. So, when congestion is about to happen, alternative paths are created from the source to sink, using the plethora of a network's unused nodes, in order to safely transmit the observed data. The creation of alternative paths involves several nodes, which are not in the initial shortest path from the source to the sink. According to simulation results, the use of these nodes leads to a balanced energy consumption, avoiding the creation of "holes" in the network and prolonging network lifetime. Random topology is employed in the evaluation section. No fairness results are presented by authors.

The HTAP algorithm consists of four major parts

- Flooding with level discovery functionality: Through this procedure, each node discovers its neighbor nodes and updates its neighbor table. In addition, sensor nodes are placed in levels from the source to the sink.
- Alternative Path Creation Algorithm: In order to avoid congestion each candidate congested receiver is sending a backpressure packet to the sender. So the sender stops the transmission of packets to the candidate congested receiver and searches in its neighbor table to find the least congested receiver in order to continue the transmission of data. The dynamic change of the receivers leads to the creation of new routes from the source to the sink.
- The Hierarchical Tree Algorithm: A hierarchical tree is created beginning at the source node. Connection is established between each transmitter and receiver using a 2-way handshake. Through this packet exchange, the congestion state of each receiver is communicated to the transmitter. The combination of the two algorithms implements Hierarchical Tree Alternative Path (HTAP) algorithm. Specifically when the neighbor nodes of a specific node is below a specified threshold the APC algorithm applies, the HT applies otherwise.
- Handling of Powerless (Dead Nodes): Special care is taken in the HTAP algorithm concerning the nodes which their battery is exhausted. Thus, when a node is going to lose its power, it is immediately extracted from the network and the tables of its neighbor nodes are updated.

CONSISE: Vedantham *et al.* proposed Congestion Control from Sink to Sensor (CONSISE) [31]. CONSISE approaches congestion problem in a different way compared to the other algorithms. From Sink to Sensors instead from Sensor to Sink. In this paper authors state that the factors that can contribute to sink-to-sensor congestion can be the reverse path contention and broadcast storms (due to packets which are broadcast from sinks to sensors). The actual functionality of CONSISE lies on the fact that sensor nodes are able to determine and adjust their sending rate based on the congestion level around their location at the end of each epoch. Upstream nodes are informed about the downstream nodes' sending rate through an explicit feedback and based on that they also adjust their sending rates. Then, downstream nodes select their preferred upstream node and notify it that it can send data in a higher data rate, while concurrently the rest of the nodes, which are not selected, reduce their data rate.

The latest efforts in WSN congestion control utilize more the cross layer concept, while they also focus on performance. Moreover, much effort is taking place in resource control for congestion mitigation.

UHCC: Wang *et al.* proposed Upstream Hop-by-Hop Congestion Control Protocol (UHCC) [64]. UHCC is a protocol based on a cross layer design that tries to reduce packet losses while guaranteeing priority-based fairness with lower control overhead. It consists of two components: Congestion Detection and Rate adjustment. An index called congestion index is responsible for providing the congestion level of each node. The congestion index takes as inputs the unoccupied buffer size and the traffic rate at the MAC layer. Based on the congestion index every upstream traffic rate is adjusted with its node priority to mitigate congestion hop-by-hop. UHCC protocol is simulated on a tree based topology and it is compares against CCF [30] and PCCP [35] protocols. Simulation results show that UHCC achieves higher throughput, better priority-based fairness and lower packet loss ratio.

FACC: Yin *et al.* in [38] deal beside congestion control, also with the fairness issue. They propose a "Fairness-Aware Congestion Control Scheme" (FACC). In this algorithm intermediate nodes are categorized into "near-sink nodes" and "near-source nodes". Near-source nodes maintain a per-flow state and allocate an approximately fair rate to each passing flow by comparing the incoming rate of each flow and the fair bandwidth share. This means that, if for example congestion occurs at an intermediate sensor, the generating rates of source nodes are forced to slow down, in accordance with this nodes' available bandwidth. Eventually, the whole network adapts toward the maximum congestion-free throughput. Furthermore, the lower generating rates will alleviate wireless interference

and contention. "Near-Source node" process comes with the following functions:

- Estimation of the Available Bandwidth.
- Computation of the Flow Arrival Rate.
- Estimation of the Number of Active Flows.
- Transmission Control on Near-Source Nodes.

On the other hand, near-sink nodes do not need to maintain a per-flow state and use a lightweight probabilistic dropping algorithm based on queue occupancy and hit frequency. They actually implement three mechanisms. A Stateless Fair Queue Management Mechanism, a Hop-by-Hop Backpressure mechanism and Fairness of the Stateless Queue-Management mechanism. The first mechanism in order to achieve fairness attempts to give more chances to those flows with lower occupancy. Thus, arriving packets that belong to higher occupancy flows have higher dropping probabilities. By using the Hop-by-Hop backpressure mechanism, FACC informs the "near-source node" for a drop packet in its flow in order for the node to adjust its sending rate.

FACC has been implemented in ns-2 simulation tool [63] and results show that FACC improves the number of dropped packets, throughput and energy consumption compare to the backpressure mechanism of CODA [27] and "no congestion control algorithm".

CADA: Fang *et al.* proposed CADA [40], an approach for Congestion Avoidance Detection and Alleviation in WSNs. In this algorithm, the congestion level of a node is measured by an aggregation of buffer occupancy and channel utilization. CADA actually counts the growing rate of the buffer's occupancy and when it exceeds a certain limit, the node is considered congested. On the other hand if the packet delivery ratio decreases drastically, while the local channel loading reaches the maximum achievable channel utilization, it infers that there is channel congestion. For congestion mitigation CADA employs both resource control and rate control depending on the case. If congestion takes place in an intersection hotspot, then resource control applies, while if congestion takes place in a convergence hotspot, traffic control applies. The performance of CADA was evaluated using random topologies of 500–5000 nodes and a number of congestion control scenarios in the ns-2 [63] simulator. CADA was compared against TARA and a no congestion control strategy. Simulation results prove that CADA present better results concerning throughput, energy consumption, end to end delay, and average per hop delay in comparison with TARA [28] and the no congestion control strategy.

DAlPaS: Sergiou *et al.* proposed a congestion control and avoidance algorithm called "Dynamic Alternative Path Selection Algorithm"(DAlPaS) [42], that attempts to choose an alternate path in case of congestion taking into account a number of basic performance parameters. Complementary to Energy Aware Protocols [65], [66] that find the lowest energy route or energy sufficient paths to forward data and base their path alternation decision on these conditions, DAlPaS also takes into consideration the node's congestion situation (both in terms of buffer occupancy and channel interference). On the other hand, while congestion control and reliable data transmission protocols like [28], [36], and [67] base their "alternate path" decision on a congestion threshold or the path's cost, DAlPaS also counts the node's remaining power. DAlPaS is a completely dynamic and distributed algorithm.

DAlPaS operates in two stages: soft and hard. When the soft stage applies, a node that receives packets from more than one flows keeps servicing the flow from which it receives packets with the higher rate and informs the nodes from which the other flows are coming to change destination node. Using this proactive method, the network avoids possible hotspots, especially when the load is not so big in the network (transient conditions). Nodes enter the hard stage when they must prohibit flows from reaching them. In this stage a node becomes temporarily or permanently unable to accept any more packets from any flows. To enforce this prohibitive state a node uses the so called "Flag Decision Mechanism". The flag decision algorithm recognizes and advertises a nodes unavailability under the following circumstances:

- Buffer Occupancy is reaching its upper limit.
- Low Remaining Power.
- Higher level node unavailability.

DAlPaS algorithm is a hop-by-hop congestion control algorithm that employs a resource control method for congestion mitigation. It also notifies the network for congestion implicitly while it provides good energy results. DAlPaS algorithm is evaluated under random topologies while according to authors, is compared favorably to TARA [28] algorithm.

B. Congestion Avoidance Algorithms

Siphon: Wan *et al.* proposed Siphon [43]. Siphon is a source-to-sink congestion control protocol that aims at maintaining application fidelity, congestion detection, and congestion avoidance by introducing some virtual sinks (VS) with a longer range (IEEE 802.11 Wi-Fi) multi-radio (such as Stargate) within the sensor network. VSs can be distributed dynamically so that they can tunnel traffic events from regions of the sensor field that are beginning to show signs of a high traffic load. At the point of congestion, these VSs divert the extra traffic through them to maintain the required throughput at the base s tation. The siphon algorithm mainly aims at addressing the VS discovery, operating scope control, congestion detection, traffic redirection, and congestion avoidance. The VS discovery works as follows: the physical sink sends out a control packet periodically with a signature byte embedded in it. The signature byte contains the hop count of the sensor nodes that should use any particular VS. Each ordinary sensor node maintains a list of neighbors through which it can reach its parent VS. Finally each VS maintains a list of its neighbor VSs. Each VS has a dual radio interface: a long range one to communicate with other VSs or with a physical sink (if applicable), and a regular low-power radio to communicate with the regular sensor nodes. In the case of congestion, a sensor node enables the redirection bit in its header and forwards the packet to its nearest VS. When the VS finds the redirection bit enabled, it routes the packets using its own long range communication network toward the physical sink, bypassing the underlying sensor network routing protocols. Siphon uses a combination of hop-by-hop and end-to-end congestion control depending on

the location of congestion. If there is no congestion, it uses hop-by-hop data delivery model. In case of congestion, it uses hop-by-hop data delivery model between source nodes and the VS at point of congestion and an end-to-end approach between the VS handling the congestion and the physical sink.

Summarizing, Siphon is a set of fully distributed algorithms that employs CODA [27] mechanisms for congestion detection (buffer occupancy and wireless channel load) as well as a Post-Facto Congestion Detection mechanism in which physical sink extract conclusions on possible network overload. The novelty of Siphon lies on the fact that it employs a set of virtual sinks for congestion mitigation through a traffic redirection way. Simulations implemented in ns-2 [63] simulator on a random topology. Also Siphon has been evaluated using experimental implementation through a testbed and is being compared to CODA [27] algorithm.

Light Weight Buffer Management: Chen *et al.* proposed light weight buffer management technique for congestion avoidance [44]. This technique is based on the fact that a sensor y sends a packet to another sensor x only when x has the buffer space to hold the packet. Taking in account that the remaining buffer of a sensor node changes whenever it receives or forwards a packet to a neighbor node, the node incorporates in the packet header its buffer state. This is done by using one bit to indicate that its buffer is full or several bits to indicate the exact remaining buffer. So neighboring nodes receive or overhear the buffer state of their neighbor and they cache its condition. According to the buffer state, the neighbor nodes decide whether to transmit, or not, new packets. This scheme avoids packet drops due to buffer overflow. Each node can adapt not only its own data rate, but also the data rate of its connected neighbors, since when the upstream nodes are congested the other nodes are forced to reduce its data rate. This procedure is iterative and finally leads to a maximum congestion-free throughput. This approach is different from other traffic control approaches due to the fact that it never drop packets. This algorithm is a hop-by-hop congestion avoidance. The proposed scheme is compared against global rate control, CODA's [27] backpressure mechanism and no congestion control. All simulations have been performed in random topology. Finally, the authors do not present specific energy related results.

CoSMoS: Karenos *et al.* proposed "COngestion avoidance for Sensors with a MObile Sink" (CoSMoS) [45]. In this work the additional challenges introduced by a mobile sink are addressed. Firstly, the rate of path reconfigurations needs to be increased in order to achieve reliable data delivery. Secondly effective load estimation techniques need to be implemented since path reconfiguration can result in sudden load changes along the paths and thirdly transient periods of reduced path quality must be proactively prevented. CoSMoS is a scheme that is based on a joint routing and congestion control approach. Cosmos scheme consists of two parts: A low cost, low complexity routing scheme that effectively considers the paths dynamic reliability variations during sink mobility and a regional load collection technique to estimate the maximum sustainable load of each node within a region and along a path. CoSMoS algorithm has been implemented in Mica-2 motes. Experimental results present that it manages to balance congestion and reliability to achieve higher delivery ratios without hurting throughput. No energy results are provided by authors.

Buffer and Rate Control Based Congestion Avoidance: Alam *et al.* proposed a "Buffer and Rate Control Based Congestion Avoidance" protocol [46]. This protocol consists of three schemes. These are: the Upstream Source Count, the Buffer Occupancy based rate control, and the Snoop based MAC level ACK. Using the first two schemes the protocol controls the rate of upstream nodes. This fact provides two advantages. The first is that congestion, due to media access contention, is reduced as the upstream nodes proactively decrease their rate, while the second is the fact that congestion due to buffer overflow is avoided as the upstream nodes defer transmission of packets whenever their downstream nodes buffer is full. In the third scheme (Snoop based MAC level ACK) explicit ACK are avoided. Instead, each node may overhear its own transmitted packet while forwarded by its downstream node. To accomplish this, the upstream node MAC address and a sequence number are appended into the MAC frame. Simulation results are provided by authors comparing this protocol with Shortest Path Routing with no congestion control, snoop based with implicit acknowledgement and Source count and buffer occupancy based rate. Results present that this protocol can reduce collision drop Rate, increase delivery ratio and improve the network's energy efficiency.

CAEE: Khan *et al.* proposed CAEE protocol designed for "Congestion Avoidance and Energy Efficiency in WSNs" [47]. The distinguished features of this protocol lie on the fact that it introduces the concept on Mobile Sinks. Specifically, in this case, the network is divided into clusters called mini-sinks. The cluster head is called data collector node. The main responsibility of a data collector node is to receive and store the collected data from the sensor field to the mini-sink. The mobile sink periodically visits each mini-sink in the sensor field for data retrieval. Simulation results (against the case where just a static sink is used) prove that the CAEE protocol can increase the network's lifetime since packets travel on a few hops (until mini-sinks) and the collected by mobile sink. Concurrently congestion hot spots can be alleviated since mini-sinks represent multiple collection points. The performance of CAEE is evaluated using OMNET++ simulation tool. A uniform but random topology is selected. CAEE is compared with the case when a static sink is employed. Simulation results present that CAEE protocol outperforms the static sink scenarios concerning packet count and energy.

TADR: He *et al.* proposed a traffic-aware dynamic routing (TADR) algorithm [48], to route packets around the congestion areas and scatter the excessive packets along multiple paths consisting of idle and under-loaded nodes. Enlightened by the concept of potential in common physics, the TADR algorithm is designed through constructing a mixed potential field using depth and normalized queue length to force the packets to steer clear of obstacles created by congestion and eventually move towards the sink.

Simulations have been performed in TOSSIM [68] simulation tool. A random topology have been used results show that TADR achieves its objectives and improves the overall

throughput by around 370% as compared to a benchmark routing protocol. Furthermore, TADR has low overhead suitable for large scale dense sensor networks.

ANAR: Hsu *et al.* proposed an Adaptive NAV-Assisted Routing (ANAR) [49] protocol to alleviate the network congestion. ANAR protocol is based on the cross-layer information and employs the existing information (the Network Allocation Vector (NAV)) from the Request-To-Send (RTS) and the Clear-to-Send (CTS) packets within the MAC scheme. Through the NAV vectors a congestion free probability is computed. This probability is carried within the route discovery process and determine the feasible route for packet delivery. The protocol is dynamic since it is able to adaptively switch between the selected paths while the level of network congestion has been changed. ANAR protocol has been implemented in ns-2 simulation tool [63] and has been compared to AODV and LBAR protocol, on a random topology. Results present better performance in terms of packet delivery ratio, end-to-end delay and power consumption.

Priority Based Medium Access Protocol: Rajsekar *et al.* proposed a Priority Based Medium Access Protocol for Congestion Avoidance [50]. This MAC protocol gives proportional access based on source count value. For example a node that carries a higher amount of traffic gets more access time than others. Each node then calculates its contention window on a provided equation. Simulation series have been performed in MATLAB and claim that an optimal contention window size can minimize collision in the MAC layer and effectively help transmit all packets without delays. This protocol in not compared to any other protocol since only the concept of the algorithm is implemented in MATLAB.

TALONet: Huang *et al.* proposed TALONet as a Power-Efficient Grid-Based Congestion Avoidance Scheme Using Multi-Detouring Technique [51]. TALONet implements three schemes: different transmission power levels in order to alleviate congestion in data link layer, buffer management for avoiding buffer level congestion, and a multi-path detouring technique in order to increase resources for congested traffic flows. The operation of TALONet consists of three phases. These are the network formation phase the data dissemination phase and the framework updating phase. In the network formation phase each node, after receiving a control packet from sink containing its location and information about the side length of each square grid, imaginarily builds a virtual grid framework G and figures out the coordinates of all virtual grid points cp in G. In this case nodes can be normal or TALON. TALON nodes are considered the nodes that are close to grid's cross points. During the Data Dissemination phase the TALON nodes are responsible for collecting and relaying the sensing data. During this phase normal nodes through the help of a grid-based routing protocol forward their data to their closer TALON. Then, this TALON forwards these data to its closer TALON until data reaches the sink. Finally, during the framework updating phase, in order to save power, the sink broadcasts control packets including offsets for all nodes. Then the network enters again into the network formation phase. TALONet is implemented in ns-2 [63] simulation tool on a grid topology. Results present that TALONet performs better in terms of power consumption and packet drops, in comparison with TARA [28], "no congestion control", and backpressure algorithms. Results are also provided for the energy performance of the algorithm.

LACAS: Misra *et al.* proposed an adaptive learning solution for congestion avoidance in WSNs named "Learning Automata-Based Congestion Avoidance Algorithm in Sensor Networks" (LACAS) [52]. The target of this work is to control the data rate of intermediate nodes in order to avoid congestion before this reaches the sink. To achieve this, code capable of taking intelligent actions (called automata) is developed at each of the network's nodes that are capable of controlling the rate of flow of data at the intermediate nodes based on probabilistically how many packets are likely to get dropped if a particular flow rate is maintained. In this case an "automaton" "learns" from past behaviors and chooses a better data rate in order to avoid congestion. Simulation results under metrics like energy consumption, throughput and collisions prove that LACAS is able to control congestion in an efficient way.

Flock-CC: Antoniou *et al.* proposed the Flock-based Congestion Control (Flock-CC) protocol [19], [39]. This approach focuses on designing a robust and self-adaptable congestion control protocol for WSNs. Flock-CC adopts a Swarm Intelligence paradigm inspired by the collective behavior of bird flocks having global self-properties achieved collectively without explicitly programming them into individual nodes. The main idea is to 'guide' packets (birds) to form flocks and flow towards the sink (global attractor), whilst trying to avoid congestion regions (obstacles). The direction of motion of a packet flock is influenced by repulsion and attraction forces between packets, as well as the field of view and the artificial magnetic field in the direction of the artificial magnetic pole (sink). In particular, packets are 'flying' through the network while being attracted to nodes with low wireless channel loading, and being repelled from nodes with high buffer occupancy. Thus, in Flock-CC, congestion is inferred using both buffer occupancy and wireless channel loading. Congestion notification is implicitly performed having each node broadcasting (using a small control packet) its buffer occupancy and the wireless channel loading in its vicinity to all nodes within the node's transmission range. Each packet synthesizes the attraction and repulsion forces to and from neighboring packets as well as the global magnetic force towards the sink and moves in an oriented manner through the network whilst avoiding congestion regions. Flock-CC is simple to implement at the individual node (each node follows a small set of rules), and involves minimal information exchange. Flock-CC was tested on both lattice and random topologies of 300 nodes using a number of scenarios for different network and traffic conditions. Performance evaluations showed the effectiveness of the Flock-CC protocol in balancing the offered load by exploiting available network resources. Flock-CC was shown to provide graceful performance degradation in terms of packet delivery ratio, packet loss, delay and energy tax under low, high and extreme traffic loads. In addition, the proposed approach achieved robustness against failing nodes, scalability in different network sizes and outperformed typical conventional approaches. Flock-CC was also qualitatively compared against AntHocNet [69] and AntSensNet [70].

LVCC: Antoniou *et al.* proposed the Lotka-Volterra based Congestion Control (LVCC) [41] protocol. LVCC focuses on streaming applications in wireless sensor networks and on how congestion can be prevented by regulating the rate of each traffic flow based on the Lotka-Volterra population model. LVCC detects congestion on the basis of buffer occupancy, while congestion avoidance is performed by means of traffic control. The traffic flows initiated by each node play the role of competing species and the buffer (queue) capacity of the parent node can be seen as the limiting resource. LVCC provides hop-by-hop rate adaptation by regulating the traffic flow rate at each node. Each node is in charge of self-regulating and self-adapting the rate of its traffic flow i.e., the rate at which it generates or forwards packets. The traffic flows compete for available buffer capacity at their one-hop-away receiving node involved in the path leading to the sink. Each sending node is expected to regulate its traffic flow rate in a way that limiting buffer capacities at all receiving nodes along the network path towards the sink are able to accommodate all received packets. The sending rate evolution of each flow will be driven by variations in buffer occupancies of nodes along the network path towards the sink. Due to the decentralized nature of the LVCC protocol, and in order to satisfy the need for low communication overhead, each node regulates its traffic flow rate using local information (i.e., from one-hop away neighbors). LVCC involves minimal exchange of information and computation burden and is simple to implement at the individual node. Performance evaluations revealed that LVCC achieves adaptability to changing traffic loads, scalability and fairness among flows, while providing graceful performance degradation as the offered load increases. LVCC was not compared against related congestion avoidance approaches found in WSN literature.

C. Reliable Data Transport Algorithms

ESRT: Sankarasubramaniam *et al.* proposed Event to Sink Reliable Transport (ESRT) [53]. ESRT considers reliability at the application level and provides stochastically reliable delivery of packets from sensors to the sink. It is an end-to-end protocol trying to guarantee a desired reliability through regulation of sensor report frequency. It provides reliability for applications, not for each single packet. The sink uses congestion feedback from sensor nodes to broadcast a notification to adjust the reporting rate with two goals: i) to receive a sufficient number of packets from the sink, and ii) to receive only as many packets as necessary in order to avoid congestion and save energy. The algorithms of ESRT runs on the sink, with minimal functionality required at resource constrained sensor nodes. The ESRT protocol operation is determined by the current network state, based on the reliability achieved and congestion condition in the network. Firstly, it needs to periodically compute the factual reliability r based on successfully received packets in a time interval. Secondly, ESRT deduces the required sensor report frequency f from r. Thirdly and finally, ESRT informs all sensors about f through an assumed channel with high power. ESRT identifies five characteristic operation regions: i) No Congestion, Low reliability, ii) No Congestion, High reliability, iii) Congestion, High Reliability, iv) Congestion, Low Reliability and v) Optimal Operating Region—which essentially translates to No Congestion, Medium-High Reliability. The target is to identify its current state and bring the network into OOR (Optimal Operating Region). If the event-to-sink reliability is lower than required, ESRT adjusts the reporting frequency of source nodes aggressively in order to reach the target reliability level as soon as possible. If the reliability is higher than required, then ESRT reduces the reporting frequency conservatively in order to conserve energy while still maintaining reliability. This self-configuring nature of ESRT makes it robust in random and dynamic topologies in WSN. An additional benefit resulting from ESRT is energy-conservation since it can control the sensor reporting frequency. A disadvantage with this algorithm is the fact that all nodes are treated equally. Therefore, in case of congestion in one region of the network all nodes are forced to reduce their data rate, affecting negatively the network's throughput. Moreover ESRT does not handle different types of events requiring different levels of reliability. Summarizing ESRT is an end-to-end congestion control protocol that focuses on reliability. It provides fairness among the nodes since data rate reduction applies in all nodes in the network, even if congestion appears in a specific area in the network. ESRT has been implemented in ns-2 [63] simulation tool and results are provided concerning reliability.

Directed Diffusion: Intanagonwiwat *et al.* proposed Directed Diffusion [13]. Directed Diffusion is a data centric protocol because all communication is for named data. All nodes in a directed diffusion-based network are application-aware. This enables diffusion to achieve energy savings by selecting empirically good paths (small delay) by caching and processing data in-network (e.g., data aggregation). Directed diffusion consists of four basic elements: interests, data messages, gradients, and reinforcements. An interest message is a query from a sink node to the network, which indicates what the application wants. It carries a description of a sensing task that is supported by a sensor network. Data in sensor networks is the collected or processed information of an event (e.g., physical phenomenon), is named (addressed) using attribute-value pairs and a sensing task is diffused throughout the sensor network as an interest for named data. This dissemination sets up gradients within the network designed to "draw" events (i.e., data matching the interest). A gradient is direction state created in each node that receives an interest. This direction is set toward the neighboring node from which the interest was received. Events start flowing towards the sinks of interests along multiple gradient paths. To improve performance and reliability, the empirically "good paths" (e.g., small delay) are reinforced by the sink and their data rate increases. On the other hand unreliable paths (e.g., high delay) are negatively reinforced and pruned off. Directed Diffusion has been implemented in ns-2 [63] simulation tool. Results are provides in comparison with Omniscient Multicast and Flooding concerning average dissipated energy, average delay and event delivery ratio.

GARUDA: Park *et al.* proposed GARUDA [59]. GARUDA provides reliable point-to-multipoint data delivery from the sink to the sensors. GARUDA consists of the following elements.

- an efficient pulsing based solution for reliable short-message delivery;

- a virtual infrastructure called the core that approximates a near optimal assignment of local designated servers, which itself is instantaneously constructed during the course of a single packet flood;
- a two-stage NACK based recovery process that effectively minimizes the overheads of the retransmission process, and performs out-of-sequence forwarding to leverage the significant spatial re-use possible in a WSN;
- a simple candidacy based solution to effectively support the different notions of reliability that might be required in a WSN.

GARUDA has been implemented if ns-2 [63] simulation tool and results focus on reliability and energy.

STCP: Iyer *et al.* proposed STCP [54]. STCP is a generic, scalable and reliable transport layer protocol where the majority of functionalities is implemented in the sink. STCP supports networks with multiple applications and provides additional functionalities such as controlled variable reliability and congestion detection and avoidance. In STCP, before transmitting packets, sensor nodes inform the sink through a "Session Initiation Packet". Through this packet, the sink is informed about the number of flows initiated from a source, the type of data, the transmission rate, and the required reliability. As soon as the sink receives this packet it sends an ACK packet to the source node, and the source node can start sending packets. STCP packet headers consists of the sequence number, a clock field, flow id, and congestion notification bit. Since the sink knows the rate of transmission from the source, the expected arrival time for the next packet can be found. The sink maintains a timer and sends a negative acknowledgement (NACK) if it does not receive a packet within the expected time. Also, sensor nodes specify the required reliability for each flow in the session initiation packet. Reliability is measured as the fraction of packets successfully received. Concerning congestion control, nodes inform the sink whether they are experiencing buffer overflow, by setting their congestion notification bit, while the sink informs the source of a congested path by setting the congestion bit on the ACK packet. In this case, a source node may alter its routing path or decrease the sending rate to mitigate congestion.

RCRT: Paek *et al.* proposed RCRT [55], a protocol that focuses on reliable delivery of sensor data form source to sink, while avoiding congestion collapse. RCRT focuses on the transport layer and its traffic management functionality resides on the sink. RCRT attempts to guarantee 100% reliable data delivery based on a NACK scheme. So in case of packet losses, the sink requests the missing packets from the source by sending a NACK with the missing packet numbers. RCRT implements at sink three basic components: the congestion detection component which detects congestion through round trip time, rate adaptation, and rate allocation which decreases flow rates to control congestion; congestion detection performed using as congestion indicator the "time to recover loss". This means that as long as the network is able to repair quickly enough the packet losses (e.g., around on Round Trip Time) the network in not congested. In other case it figures out that there are congested spots in network. In case of congestion RCRT applies a rate adaptation mechanism to control it. Also RCRT applies a rate allocation mechanism on which specific rates are allocated to each flows when the application differs (e.g., video transmission etc.). STCP has been implemented in TOSSIM simulation tool and results are provided also for energy spent and packet's latency.

Extended DCCP: Liu *et al.* proposed an extension to Datagram Congestion Control Protocol with a new congestion control component [56]. DCCP is a transport layer protocol designed for providing a standard way to introduce congestion control and congestion control negotiations into multimedia applications. Extended DCCP comes with the following added functions:

- Buffering of received packets at the receivers,
- retransmission of lost or corrupted packets by the senders,
- detection and deletion of duplicated packets at the receivers,
- in-order delivery of received packets to the application program at the receivers.

In this case the sender has four states: Normal State, Congestion State, Failure State (route change or link failure), and Error State (transmission error). Overall, extended DCCP has the ability to provide a reliable operation, with a good aggregate throughput.

VIII. Discussion

In this section, we present a few directives, which, according to our opinion, are important for a network engineer to follow in order to design and develop new congestion control algorithms in WSNs. These directives follow the comparison between the algorithms as presented in Section VII and its summarization in Tables I, II, and III.

It is certain, that the first task that should be accounted in this effort, is the application. Application will define the way that congestion should be detected, mitigated or avoided. WSNs are currently being employed in a plethora of applications ranging from medical to military, and from home to industry. All WSN applications can be categorized under three data delivery models: a) event-based, b) continuous-based (streaming) and c) query-based.

Typically, when event-driven applications are involved, WSNs operate under light load but large, sudden, and correlated-synchronized impulses of data may suddenly arise in response to a detected or monitored event. Most event-driven applications (e.g., target tracking, fire detection, monitoring of wildlife in forests) are interactive, delay intolerant (real-time) and mission critical. That means that data generated from sensor nodes should be delivered within short span of time through a sink node to a processing center for further actions. The data traffic generated by a single sensor node may be of very low intensity. However, very bursty traffic may be generated by a set of sensors due to a common event or a phenomenon. In addition, the converging (many-to-one) nature of packets from multiple sending nodes to one or more sink nodes may lead to congestion around the sinks. Event-based applications can lead to transient congestion phenomena due to aperiodic and short term packet bursts.

Continuous-driven (streaming) applications e.g., video surveillance, traffic control systems, health monitoring, and industrial process control can be observed in multimedia WSNs

[71]. In streaming applications, sensor nodes send their data continuously to the sink at a constant or regulated (by a formula) rate. Streaming applications may server real-time or non-real-time data. Real-time data is delay-constrained and has a certain bandwidth requirement. Packet losses can be tolerated to a certain extent. On the other hand, non-real-time data can be send when the sink may want to collect periodic data from the sensor field. In this context, delay and packet losses are both tolerated. In streaming applications, the increasing reporting rate of nodes, perhaps due to demand of higher data fidelity, in conjunction with the uncontrolled use of scarce network resources may lead to congestion. More specifically, packet flows left uncontrolled (i.e., to reach high sending rates, or to use the same paths to the sink) are likely to cause congestion even if local contention is minimized. Streaming applications may lead to persistent congestion phenomena.

Query-driven applications in WSNs are similar to event-driven applications, except that the data is pulled by the sink (as a response to a query) whereas in event-driven applications the data is pushed from source nodes to the sink (triggered by an event). Additionally, a query may be also used to manage or reconfigure the sensor nodes. It is important to note that the commands from the sink constitute one-way traffic and require high reliability.

In a WSN involving an event-driven or a query-driven application, where a transient congestion situation is expected to arise, congestion mitigation or avoidance mechanisms should be adopted as a counteraction traffic control method. This method is simple and efficient and it has been employed by the majority of congestion control algorithms [11], [27], [29]–[33], [35], [37], [38], [41]. On the other hand if the application expects heavy and persistent data load (continuous-driven application), while all data must reach the sink, then a resource control method should be employed. In this case, the data load is distributed through multiple and alternative paths, to several nodes in the networks, resulting in congestion mitigation or avoidance while it extends network lifetime. Examples like [28], [36], [39], and [42] support this kind of applications.

Concerning congestion detection, the situation is a bit different. In this case we notice that the decision on which method to choose, is, in some way, based on the MAC protocol that it is chosen and the topology control algorithm. If the MAC protocol is able to handle efficiently enough packet collisions in the medium and is able to deliver without delay the packets to the sink, congestion control algorithms can detect congestion only by measuring the buffer occupancy. Similarly, topology control algorithms relax the already densely initial placement of nodes and reduce contention in the MAC layer. Algorithms like [36], [41] are typical examples of approaches that adopt this concept. On the other hand, if the application demands more global solutions that do not rely on the effectiveness of the underlying MAC protocol, then both buffer occupancy and wireless channel load are considered as the mean for congestion detection. The majority of algorithms ([27], [28], [30], [34], [37], [39], [40], [42]) employ this method for congestion detection. As far as the other methods for congestion detection that we presented in Section V, we believe that they can be employed only in specific cases and not as a rule, as we discussed before.

From the perspective of congestion notification the situation is clear. Explicit congestion notification has already been abandoned because of the need for special control messages to notify the involved nodes that congestion is occurring. Besides the two very first efforts [27], [53], all subsequent congestion control algorithms adopted the implicit notification scheme since it does not add any extra load to the already loaded network.

Thus, we believe that a network engineer that needs to develop a new congestion control algorithm can base their initial consideration on the aforementioned directives which are shown to depend heavily on the application running over the WSN as well as the underlying MAC protocol. It is also important to stress that all algorithms and schemes that apply in WSNs, should be lightweight and exhibit low energy consumption.

IX. Future Directions

WSNs are expected to be deployed for several mission-critical tasks and operate unattended for extended periods of time. Due to the unpredictable nature of network operation, an effective and efficient congestion control mechanism should exhibit remarkable survivability and robustness to external stimuli and internal perturbations or loss of units, as well as excellent scaling properties as the number of sensor nodes scales up. Self-adaptation should be one of the major strengths of congestion control mechanisms as they must respond to the addition or removal of nodes, as well as to sudden changes in the environment. Due to the memory-constrained and energy-constrained nature of WSNs, novel congestion control approaches should be simple to implement at the individual node level and operate with minimal exchange of information. End-to-end congestion control approaches will not be effective in such error prone environments because the end-to-end nature may result in reduced responsiveness, causing increased latency and high error rates, especially during long periods of congestion. Therefore, WSNs necessitate autonomous and decentralized congestion control strategies that promise fast, effective, and efficient relief from congestion. Decentralized approaches are expected to adopt a hop-by-hop model where all nodes along a network path can be involved in the procedure. Each node should make decisions based only on local information (e.g., buffer load, channel load) since none of them has complete information about the system state. In traditional Internet congestion control approaches, queue (buffer) occupancy and buffer drops are often taken as congestion indication. However, simulation studies conducted by [27] and [30], revealed that in WSNs, where the wireless medium is shared using Carrier Sense Multiple Access (CSMA)-like protocols, wireless channel contention losses can dominate buffer drops and increase quickly with the offered load. The problem of channel losses is worsened around hotspot areas, as for example, in the area of an event, or around the sink. In the former case, congestion occurs if many nodes report the same event concurrently, while in the latter case congestion is experienced due to the converging (many-to-one) nature of packets from multiple sending nodes to a single sink node. These phenomena result in the starvation of channel capacity in the vicinity of senders, while the wireless medium capacity can reach its upper limit faster than queue occupancy [72]. Thus,

queue occupancy alone cannot accurately serve as an indication of congestion. Novel WSN congestion control approaches are expected to take into account both metrics (queue occupancy or buffer drops and wireless channel contention losses) in order to infer congestion phenomena [11]. Finally throughout the evolution of Congestion Control and Avoidance algorithms we notice that researchers devote a large portion of the evaluation of their algorithms to a number of performance metrics (power, end-to-to end delay etc.). We believe that the next step in this field will focus on Performance Controlled Algorithms in which proper congestion handling will be a basic factor for their success. Already the evolution of Wireless Sensor Networks to Wireless Multimedia Sensor Networks strike the road to this direction.

X. Conclusion

In this paper, we presented a survey on Congestion Control and Avoidance approaches in Wireless Sensor Networks. We have briefly analyzed the types of congestion in WSNs and reviewed several protocols, techniques, and mechanisms dealing with this problem. We have also classified them according to their basic attributes. Based on the problems that arise from the constrained nature of WSNs as well as taking into account the inefficiencies of existing congestion control approaches, we derived some potential directions for the performance improvement of future congestion control approaches.

References

[1] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 54–61, Dec. 2004.

[2] T. Haenselmann, "Sensor networks," in *Free (GNU) Textbook on Wireless Sensor Networks*, Apr. 2005.

[3] A. Cerpa *et al.*, "Habitat monitoring: Application driver for wireless communications technology," in *Proc. SIGCOMM: Workshop Data Commun. Latin Amer. Caribbean*, 2001, pp. 20–41.

[4] E. Biagioni and K. Bridges, "The applications of remote sensor technology to assist the recovery of rare and endangered species," *Int. J. High Perform. Comput. Appl., Special Issue Distrib. Sensor Netw.*, vol. 16, no. 3, pp. 315–324, Aug. 2002.

[5] L. Schwiebert, S. K. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *Proc. 7th Annu. Int. Conf. MobiCom*, 2001, pp. 151–165.

[6] H. T. Kung and D. Vlah, "Efficient location tracking using sensor networks," in *Proc. IEEE WCNC*, Mar. 2003, pp. 1954–1962.

[7] R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1163–1171, Aug. 2003.

[8] C. Chien, I. Elgorriaga, and C. McConaghy, "Low-power direct-sequence spread-spectrum modem architecture for distributed wireless sensor networks," in *Proc. Int. Symp. Low Power Electron. Des.*, 2001, pp. 251–254.

[9] R. J. Cramer, M. Z. Win, and R. A. Scholtz, "Impulse radio multipath characteristics and diversity reception," in *Proc. IEEE ICC*, 1998, vol. 3, pp. 1650–1654.

[10] E. Shih *et al.*, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proc. 7th Annu. Int. Conf. MobiCom*, 2001, pp. 272–287.

[11] A. Woo and D. E. Culler, "A Transmission control scheme for media access in sensor networks," in *Proc. 7th Annu. Int. Conf. MobiCom*, 2001, pp. 221–235.

[12] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proc. Int. Conf. Distrib. Comput. Syst.*, 2003, pp. 46–55.

[13] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[14] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. 5th Annu. ACM/IEEE Int. Conf. MobiCom*, 1999, pp. 174–185.

[15] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestmant, "A survey of gossiping and broadcasting in communication networks," *Networks: An International Journal*, vol. 18, no. 4, pp. 319–349, 1988.

[16] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, RAP: A real-time communication architecture for large-scale wireless sensor networks, Charlottesville, VA, USA, Tech. Rep., 2002. [Online]. Available: http://portal.acm.org/citation.cfm?id=900537

[17] C. Sergiou and V. Vassiliou, "Estimating maximum traffic volume in wireless sensor networks using fluid dynamics principles," *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 257–260, Feb. 2013.

[18] C. Sergiou, "Performance-aware congestion control in wireless sensor networks using resource control," Ph.D. dissertation, University of Cyprus, Nicosia, Cyprus, 2012.

[19] P. Antoniou, "Nature-inspired congestion control and avoidance in wireless sensor networks," Ph.D. dissertation, University of Cyprus, Nicosia, Cyprus, 2012.

[20] J. Zhao *et al.*, "A survey of congestion control mechanisms in wireless sensor networks," in *Proc. 6th Int. Conf. IIH-MSP*, 2010, pp. 719–722.

[21] G. Srinivasan and S. Murugappan, "A survey of congestion control techniques in wireless sensor networks," *Int. J. Inf. Technol. Knowl. Manag.*, vol. 4, no. 2, pp. 413–415, Jul.–Dec. 2011.

[22] R. Chakravarthi, C. Gomathy, S. Sebastian, K. Pushparaj, and V. B. Mon, "A survey on congestion control in wireless sensor networks," *Int. J. Comput. Sci. Commun.*, vol. 1, no. 1, pp. 161–164, Jan.–Jun. 2010.

[23] C. Wang, M. Daneshmand, B. Li, and K. Sohraby, "A survey of transport protocols for wireless sensor networks," *IEEE Netw.*, vol. 20, no. 3, pp. 34–40, May/Jun. 2006.

[24] C. Wang, K. Sohraby, B. Li, and W. Tang, "Issues of transport control protocols for wireless sensor networks," in *Proc. ICCCAS*, May 2005, pp. 422–426.

[25] E. Dashkova and A. Gurtov, "Survey on congestion control mechanisms for wireless sensor networks," in *Internet of Things, Smart Spaces, Next Generation Networking*, vol. 7469, S. Andreev, S. Balandin, and Y. Koucheryavy, Eds. Berlin, Germany: Springer-Verlag, 2012, ser. Lecture Notes in Computer Science, pp. 75–85.

[26] D. J. Flora, V. Kavitha, and M. Muthuselvi, "A survey on congestion control techniques in wireless sensor networks," in *Proc. ICETECT*, 2011, pp. 1146–1149.

[27] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in *Proc. 1st Int. Conf. Netw. SenSys*, 2003, pp. 266–279.

[28] J. Kang, Y. Zhang, and B. Nath, "TARA: Topology-aware resource adaptation to alleviate congestion in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 7, pp. 919–931, Jul. 2007.

[29] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. SenSys*, 2004, pp. 148–161.

[30] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. SenSys*, 2004, pp. 134–147.

[31] R. Vedantham, R. Sivakumar, and S.-J. Park, "Sink-to-sensors congestion control,"*Ad Hoc* Networks, vol. 5, no. 4, pp. 462–485, May 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B7576-4JM0T76-1/2/14742f9f

[32] K. Karenos, V. Kalogeraki, and S. V. Krishnamurthy, "Cluster-based congestion control for supporting multiple classes of traffic in sensor networks," in *Proc. 2nd IEEE Workshop EmNets*, 2005, pp. 107–114.

[33] C. Wang, K. Sohraby, and B. Li, "SenTCP: A Hop-by-Hop congestion control protocol for wireless sensor networks," in *Proc. IEEE INFOCOM (Poster Paper)*, Mar. 2005, pp. 107–114.

[34] L. Popa, C. Raiciu, I. Stoica, and D. S. Rosenblum, "Reducing congestion effects in wireless networks by multipath routing," in *Proc. IEEE ICNP*, 2006, pp. 96–105.

[35] C. Wang, K. Sohraby, V. Lawrence, B. Li, and Y. Hu, "Priority-based congestion control in wireless sensor networks," in *Proc. Int. Conf. Sensor Netw. Ubiquitous, Trustworthy Comput.*, 2006, vol. 1, pp. 22–31.

[36] C. Sergiou, V. Vassiliou, and A. Paphitis, "Hierarchical Tree Alternative Path (HTAP) algorithm for congestion control in wireless sensor networks," *Ad Hoc Netw.*, vol. 11, no. 1, pp. 257–272, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2012.05.010

[37] G. Wang and K. Liu, "Upstream Hop-by-Hop congestion control in wireless sensor networks," in *Proc. IEEE 20th Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2009, pp. 1406–1410.

[38] X. Yin, X. Zhou, R. Huang, Y. Fang, and S. Li, "A fairness-aware congestion control scheme in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 5225–5234, Nov. 2009.
[39] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht, and L. Michael, "Congestion control in wireless sensor networks based on bird flocking behavior," *Comput. Netw. J.*, vol. 57, no. 5, pp. 1167–1191, Apr. 2013.
[40] W.-W. Fang, J.-M. Chen, L. Shu, T.-S. Chu, and D.-P. Qian, "Congestion avoidance, detection and alleviation in wireless sensor networks," *J. Zhejiang Univ.—Sci. C*, vol. 11, no. 1, pp. 63–73, 2010. [Online]. Available: http://dx.doi.org/10.1631/jzus.C0910204
[41] P. Antoniou and A. Pitsillides, "A bio-inspired approach for streaming applications in wireless sensor networks based on the Lotka-Volterra competition model," *Comput. Commun.*, vol. 33, no. 17, pp. 2039–2047, Nov. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.07.020
[42] C. Sergiou and V. Vassiliou, "DAlPaS: A performance aware congestion control algorithm in wireless sensor networks," in *Proc. 18th ICT*, May 2011, pp. 167–173.
[43] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: Overload traffic management using multi-radio virtual sinks in sensor networks," in *Proc. 3rd Int. Conf. Embedded Netw. SenSys*, 2005, pp. 116–129.
[44] S. Chen and N. Yang, "Congestion avoidance based on lightweight buffer management in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 9, pp. 934–946, Sep. 2006.
[45] K. Karenos and V. Kalogeraki, "Facilitating congestion avoidance in sensor networks with a mobile sink," in *Proc. IEEE RTSS*, 2007, pp. 321–332.
[46] M. M. Alam and C. S. Hong, "Buffer and rate control based congestion avoidance in wireless sensor networks," in *Proc. KIPS*, May 2007, pp. 1291–1293.
[47] M. I. Khan, W. N. Gansterer, and G. Haring, "Congestion avoidance and energy efficient routing protocol for wireless sensor networks with a mobile sink," *J. Netw.*, vol. 2, no. 6, pp. 42–49, 2007.
[48] T. He, F. Ren, C. Lin, and S. Das, "Alleviating congestion using traffic-aware dynamic routing in wireless sensor networks," in *Proc. IEEE SECON*, 2008, pp. 233–241.
[49] Y.-P. Hsu and K.-T. Feng, "Cross-layer routing for congestion control in wireless sensor networks," in *Proc. IEEE Radio Wireless Symp.*, 2008, pp. 783–786.
[50] G. Rajsekar *et al.*, "Collision avoidance scheme in energy constrained wireless sensor networks using MAC protocol," in *Proc. ICCCN*, 2008, pp. 1–4.
[51] J.-M. Huang, C.-Y. Li, and K.-H. Chen, "TALONet: A power-efficient grid-based congestion avoidance scheme using multi-detouring technique in wireless sensor networks," in *Proc. Wireless Telecommun. Symp.*, 2009, pp. 1–6.
[52] S. Misra, V. Tiwari, and M. Obaidat, "LACAS: Learning automata-based congestion avoidance scheme for healthcare wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 466–479, May 2009.
[53] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003, pp. 177–188.
[54] Y. Iyer, S. Gandham, and S. Venkatesan, "STCP: A generic transport layer protocol for wireless sensor networks," in *Proc. 14th ICCCN*, 2005, pp. 449–454.
[55] J. Paek and R. Govindan, "RCRT: Rate-controlled reliable transport for wireless sensor networks," in *Proc. 5th Int. Conf. Embedded Netw. SenSys*, 2007, pp. 305–319.
[56] Y.-M. Liu and X.-H. Jiang, "An extended DCCP congestion control in wireless sensor networks," in *Int. Workshop Intell. Syst. Appl.*, 2009, pp. 1–4.
[57] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy, "P.S.F.Q: A reliable transport protocol for wireless sensor networks," in *Proc. 1st ACM Int. Workshop WSNA*, 2002, pp. 1–11.
[58] F. Stann and J. Heidemann, "RMST: Reliable data transport in sensor networks," in *Proc. 1st Int. Workshop Sensor Net Protocols Appl.*, Anchorage, AK, USA, Apr. 2003, pp. 102–112.
[59] S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz, "A scalable approach for reliable downstream data delivery in wireless sensor networks," in *Proc. 5th ACM Int. Symp. MobiHoc*, 2004, pp. 78–89.
[60] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 63–74, Aug. 2006. [Online]. Available: http://doi.acm.org/10.1145/1151659.1159922
[61] M. Ghalehnoie, N. Yazdani, and F. Salmasi, "Fuzzy rate control in wireless sensor networks for mitigating congestion," in *Proc. Int. Symp. Telecommun.*, 2008, pp. 312–317.
[62] J. Jin, M. Palaniswami, and B. Krishnamachari, "Rate control for heterogeneous wireless sensor networks: Characterization, algorithms and performance," *Comput. Netw.*, vol. 56, no. 17, pp. 3783–3794, Nov. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128612003131
[63] The Network Simulator ns-2 (v2.1b8a), Oct. 2001.
[64] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu, "Upstream congestion control in wireless sensor networks through cross-layer optimization," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 786–795, May 2007.
[65] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, 2001.
[66] R. P. Mann, K. R. Namuduri, and R. Pendse, "Energy-aware routing protocol for *Ad Hoc* wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 5, pp. 635–644, 2005.
[67] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient broadcast: A robust data delivery protocol for large scale sensor networks," *Wireless Netw.*, vol. 11, no. 3, pp. 285–298, 2005.
[68] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. 1st Int. Conf. Embedded Netw. SenSys*, 2003, pp. 126–137.
[69] G. D. Caro, F. Ducatelle, and L. M. Gambardella, "Anthocnet: An adaptive nature-inspired algorithm for routing in mobile *ad hoc* networks," *Eur. Trans. Telecommun.*, vol. 16, no. 5, pp. 443–455, 2005.
[70] L. Cobo, A. Quintero, and S. Pierre, "Ant-based routing for wireless multimedia sensor networks using multiple QoS metrics," *Comput. Netw.*, vol. 54, no. 17, pp. 2991–3010, Dec. 2010.
[71] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, 2007.
[72] M. Vuran, V. Gungor, and O. Akan, "On the interdependence of congestion and contention in wireless sensor networks," in *Proc. ICST SenMetrics*, San Diego, CA, USA, Jul. 2005.

Charalambos Sergiou (M'09) received the Diploma in engineering from Hellenic Air Force Academy, Athens, Greece, in 2000 and the M.Sc. degree in advance information technology and the Ph.D. degree from the University of Cyprus, Nicosia, Cyprus, in 2007 and 2012, respectively. He is currently a Postdoctoral Researcher with the University of Cyprus. His research interests include wireless ad hoc and sensor networks, focusing on overload and congestion control.

Pavlos Antoniou received the Dip.-Ing (M.Sc. equivalent) degree from the National Technical University of Athens, Athens, Greece, in 2005 and the Ph.D. degree from the University of Cyprus, Nicosia, Cyprus, in 2012. From September 2005 to June 2009, he served as a Research Associate with the Department of Computer Science. He worked for the GINSENG Project funded by the European Union and the locally funded MiND2C project dealing with performance control in wireless sensor networks. He is currently a member of the Special Teaching Staff with the Department of Computer Science, University of Cyprus. His current research interests include congestion control and avoidance based on nature-inspired techniques such as swarm intelligence and population biology for providing adaptation, and robustness and self-organization in autonomous decentralized environments.

Vasos Vassiliou (M'97) received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 1999 and 2002, respectively. From 2002 to 2004, he was an Assistant Professor with the Department of Computer Science, Intercollege, Nicosia, Cyprus. From 2004 to 2005 and from 2005 to 2011, he was a Lecturer and a Visiting Lecturer, respectively, with the Department of Computer Science, University of Cyprus. He is currently an Assistant Professor and a Co-Director of the Networks Research Laboratory with the Department of Computer Science, University of Cyprus. He has published several articles in international conferences and journals and participates actively in COST actions and local and European projects. His research interests include network architectures [IPv6 and multiprotocol label switching (MPLS)], mobile networks (Mobile IP, mobile MPLS, ad hoc and sensor networks), wireless communications (protocol enhancements for 3G/4G cellular wireless networks), and quality of service and traffic engineering for computer and telecommunication networks. Dr. Vassiliou is a member of The Institution of Engineering and Technology and Association for Computing Machinery. He has served in several technical program committees of international conferences such as the IEEE Global Communications Conference, the IEEE Vehicular Technology Conference, the IEEE International Symposium on Personal Indoor and Mobile Radio Communications, and European Wireless He also serves as an Associate Editor for the Journal of Telecommunication Systems.