

Congestion-aware and traffic load balancing scheme for routing in WSNs

Omer Chughtai^{1,2} · Nasreen Badruddin¹ · Azlan Awang¹ · Maaz Rehan²

Published online: 15 January 2016
© Springer Science+Business Media New York 2016

Abstract Congestion in a Wireless Sensor Network (WSN) is one of the causes of performance degradation due to severe packet loss that leads to excessive energy consumption. Solutions in WSNs try to avoid and overcome congestion by selecting sensor nodes with sufficient buffer space and adjusting the traffic rate at the source node over the shortest discovered route that usually decreases the End-to-End (ETE) throughput. On-demand routing protocols have the potential to discover the least congested route when it is required. In a WSN, most of the on-demand routing protocols replace the routing metric of the prevalent routing protocol with their proposed routing metric and keep the route discovery mechanism intact, which is not sufficient to increase the performance of the WSN. To address these problems, a novel Congestion-aware and Traffic Load balancing Scheme (CTLS) for routing has been proposed. The CTLS proactively avoids congestion through a novel route discovery mechanism to select the optimum node based on a composite metric. If congestion occurs, CTLS tries to detect it in a timely manner and alleviates it reactively using a novel ripple-based search approach. The simulation results show that the CTLS performs better as compared to the congestion avoidance, detection and alleviation and no congestion control schemes in terms of packet delivery ratio, ETE delay, throughput, and energy consumption per data packet in a resource constraint wireless network.

Keywords Congestion-aware routing · Traffic load balancing · On-demand routing · Congestion avoidance · Congestion detection · Congestion alleviation · WSN

1 Introduction

Recent technological advancements in electronic Integrated Circuits (ICs) and Micro Electro Mechanical Systems (MEMS) have enabled the development of smart wireless devices that can work in a multi-hop ad hoc fashion. The emergence of multi-hop wireless ad hoc networks has made a significant impact on a wide range of fields and services in many industrial and military environments. Services like monitoring, data acquisition and connectivity under limited power, resources and broadcast range possess a special class of wireless ad hoc networks, known as Wireless Sensor Networks (WSNs) [1]. WSNs consist of a set of wireless nodes connected through wireless links without any central administration. Applications of WSNs include precision agriculture, animal tracking, medical diagnostics, and disaster management. With such applications, utilizing the network resources fairly to balance the traffic load and to avoid congestion is a challenging task. In this regard, an important factor is the routing protocol that routes the data traffic from the source to the sink over the discovered route.

Routing plays an important role in increasing the performance of WSNs [2]. Depending on how the routes are discovered, established and maintained, routing protocols are generally classified as proactive, reactive, and hybrid. Apart from these classifications, routing can also be classified as cooperative routing and non-cooperative routing. Most of the standardized routing protocols available in multi-hop wireless ad hoc networks can be used in WSNs [3], due to their similar wireless characteristics. A large number

✉ Omer Chughtai
umar.chughtai@gmail.com

¹ Centre for Intelligent Signal and Imaging Research (CISIR),
Universiti Teknologi PETRONAS, Tronoh, Malaysia

² COMSATS Institute of Information Technology, Islamabad,
Pakistan

of routing algorithms and protocols have been proposed in WSNs. The majority of these routing protocols has been designed with the aim to find an energy-aware shortest path so that the ETE delay and energy consumption by the sensor nodes can be minimized. For performance measures such as, energy, reliability, ETE throughput and delay, on-demand routing protocols are considered to be more appropriate in a resource constrained network with limited power, resources and broadcast range, such as WSNs [4,5].

In WSNs, most of the reactive routing protocols are usually designed by taking into account the energy consumption as its primary consideration, like [6]. However, the batteries of the sensor nodes can be recharged, for example, through cost effective energy harvesting mechanisms. Therefore, other important factors like reliability as PDR, ETE delay and throughput are somehow equally important. These performance measures can be considered to avoid congestion and to balance the energy consumption in order to increase the performance of the entire network. Based on these factors, the single criterion on-demand or time-driven routing protocols like Sensor Protocols for Information via Negotiation (SPIN) [7], Direct-Diffusion [8], Gradient-Based Routing (GBR) [9], Low-Energy Adaptive Clustering Hierarchy (LEACH) [10], Dynamic Source Routing (DSR) [11], Ad hoc On-demand Distance Vector (AODV) [12], etc., are considered suitable in multi-hop WSNs [4].

Amongst these classical routing protocols, SPIN cannot guarantee for the delivery of data packets to the sink and it needs to maintain the global network topology. Direct-Diffusion is a Data-Centric routing protocol in which, it is mandatory for the base-station to produce and re-send the periodic updates. These periodic updates reduce the channel utilization for the data packets that decreases the overall gain in the network. Additionally, it also maintains multiple paths that can be utilised simultaneously, which may affect the performance of the network in term of interference due to route coupling. GBR is the variant of direct diffusion, where each sensor node has to maintain the number of hops to the destination, even when the sensor nodes do not want to participate in data forwarding or are not on an active route. The major disadvantage of such routing protocols is that they do not work in the application that requires continuous data forwarding to the base-station as discussed in [13], because, the continuous exchange of data packets may result in congestion. LEACH is a hierarchy based routing protocol, in which, a representative or an administrative node among the set of sensor nodes, such as cluster head, needs to be elected. However, in this work, flat-based routing is used because of the equal role and functionality assigned to each sensor node deployed in the network.

Congestion usually occurs when a sensor node is utilized as a relay node for multiple flows during data communication and drops the packet it receives beyond its capacity.

Congestion also occurs when the demand of the resources exceeds the available capacity of the system, which eventually increases the ETE delay and packet loss. Avoiding congestion is relatively equivalent to the traffic load balancing in wireless ad hoc networks. The probable effect of congestion and unfair traffic distribution will result in unstable paths that can overload the sensor nodes and soon deplete the energy of some sensor nodes, which consequently results in a network partitioning. This phenomenon is generally known as the hot spot problem, and hence the network performance suffers.

Congestion management techniques are mainly categorized as either congestion avoidance or congestion control. Congestion avoidance is a proactive scheme, whereas, congestion control is a reactive scheme. A congestion avoidance algorithm is used to prevent the network from overloading and to allocate the resources fairly across the network proactively, to balance the traffic load. Congestion control algorithms, on the other hand, are used to recover the network from congestion reactively, and try to regulate the traffic flow to make the network in an operational state. Two types of congestion generally occur in a network, the first one is based on the characteristics of the sensor node which is generally known as a node level congestion and the second one represents the characteristics of a wireless link which is a link level congestion [14].

The rest of the paper is organized as follows: Sect. 2 discusses related work on routing mechanisms in WSNs and the issues associated with these mechanisms to avoid, detect, and alleviate congestion with the motivation that has driven us to design and develop the proposed scheme. In Sect. 3, the proposed mechanism with congestion avoidance, detection, and alleviation is presented. The performance of the proposed mechanism, its evaluation and the results and discussion are presented in Sect. 4 and finally, Sect. 5 concludes the paper with some future recommendations.

2 Related work

Congestion avoidance, detection, and alleviation are important considerations in WSNs. To avoid congestion proactively, the routing protocol plays an important role to select the best nodes and to route the data traffic from the source to the destination. To detect congestion in a timely manner during data forwarding, sensor nodes monitor the buffer occupancy and the channel utilization. On the other hand, congestion alleviation schemes control congestion reactively either by adjusting the source traffic rate or by re-discovering the new route. All these three mechanisms are able to increase the performance and to balance the traffic load in multi-hop WSNs. The following subsections describe some of the

related works on congestion avoidance, detection, and alleviation in WSNs.

2.1 Congestion avoidance and detection in WSNs

Congestion avoidance through a route discovery procedure to balance the traffic load and to increase the network performance has been considered rarely except from some notable exceptions like [15, 16]. In these techniques, the assumption is that the node knows the geographical location of itself, its neighbours, and the sink, using some localization techniques based on external entities such as the Global Positioning System (GPS). This is an impractical solution in WSNs [17] due to factors such as volume, energy consumption, and cost. Based on the literature, the sink node is assumed to be a super node such that it covers the whole sensor field. However, this assumption is also considered impractical in large scale sensor networks as discussed in [18]. Congestion avoidance and detection mechanisms are proposed in [19], which utilizes the sink to sensor node notifications, but is mainly concerned with the same class of traffic. Moreover, congestion is detected by the combination of the buffer occupancy of a sensor node and channel utilization of the wireless medium.

An enhanced congestion avoidance and detection protocol that uses the buffer capacity and the difference between the weighted buffers of the sensor nodes has been proposed in [20]. In [21], congestion aware routing has been proposed by assigning the priorities to the data packets. In this scheme, multiple sinks are deployed at the border of the sensor field to gather the appropriate incoming data. The deployment of multiple sink nodes is also a challenging task due to factors such as sink placement, number of sink nodes, and the services provided by the sink node(s). Another mechanism to avoid and detect congestion in a WSN has been proposed in [18], in which a representative node is selected from a single broadcast range that implicitly avoids congestion. It detects congestion by periodically monitoring the buffer occupancy and the channel utilization.

A multi-path routing to avoid congestion in a multi sink environment is discussed in [22]. In this approach, a ratio based on the number of nodes near to the source to the number of nodes near to the sink has been considered. Monitoring the queue of the nodes near to the sink helps to ensure the load balancing and fairness. This technique avoids congestion by notifying the predecessor nodes (the nodes from where it receives the data or the backward nodes) to decrease the traffic send rate; but, this eventually drops the ETE throughput. All of the above techniques either monitor the traffic rate or buffer occupancy and the channel utilization to avoid and detect congestion. The next section provides a related work on alleviating congestion in a WSN.

2.2 Congestion control in WSNs

Congestion control algorithms have also been proposed to balance the traffic load in WSNs. Some of the solutions are based on a single-metric and some are based on multi-metric criterion. Most of the studies have been conducted to mitigate congestion by adjusting the data traffic rate at the source node. To mitigate congestion in multi-hop WSNs, the state of a sensor node is monitored and generally shared with the neighbour nodes. Congestion can be controlled at various layers of the Open System Interconnection (OSI) model. Most of the solutions to control congestion are contingent with the transport layer of the OSI model. A solution based on the traffic rate control at the transport layer to mitigate congestion in WSNs has been studied in [23]. In [19], congestion is controlled reactively by adjusting the traffic rate of the source node. In this mechanism, the source node adjusts its rate by receiving the notification sent by the intermediate node(s) where congestion has been detected. This notification is sent through an explicit congestion notification message to the source node. The overhead in this technique is that the source nodes require continuous feedback from the sink to maintain the traffic rate. The congestion status is piggybacked in [20], by taking the difference between the buffer occupancy of a current node and the neighbour node. After receiving this piggybacked message from the source node, it either adjusts the traffic rate or diverts the traffic to an alternate route if the detour path is available.

Even though the traffic rate control is able to effectively mitigate congestion, but it reduces the overall gain in terms of throughput and is usually based on a single class of traffic. The technique to control congestion via source rate adjustment is proposed in [24], which considers an impartial assumption that the sink covers the entire sensor field as also discussed in [18]. A technique used to control congestion based on the adaptive compression has been proposed in [25] in which, the queue is monitored and adjusted adaptively in the case of congestion. Here, congestion is controlled by decreasing the packet transmission rate, which reduces the ETE throughput. A mechanism to avoid, detect, and alleviate congestion in a WSN has been proposed in [18]. Here, sensor nodes periodically monitor the buffer occupancy and the channel utilization to detect the congestion state. It alleviates congestion by sending a notification to the 2-hop predecessor node to bypass the congested node or to send the notification to the source node through predecessor nodes to adaptively adjust the traffic rate.

Apart from the schemes where the traffic rate is adjusted at the source node, attempts made to mitigate congestion through other techniques have also been explored in WSNs. A topology-aware strategy to alleviate congestion has been proposed in [26]. However, this scheme requires local and global knowledge about the topology, which might create an

overhead in a dense network with a large number of nodes. A solution based on the network layer to mitigate congestion using a cost-to-benefit ratio is proposed in [27]. This cost is computed as the ratio of the total number of packets received at the sink to the total number of packets relayed by the intermediate nodes.

Along with the single path routing, multi-path routing to balance the traffic load and to increase the throughput has also been proposed in the literature. However, the cost to maintain the multiple paths is expensive in WSNs. A traffic aware routing mechanism to mitigate congestion has been proposed in [28]. It uses a hybrid virtual potential field, integrating the normalized queue length and the depth of a node from the sink. It maintains the routing table according to the least depth from the sink that is the shortest path. However, in order to alleviate congestion, it might discover a longer path, which increases the ETE delay. In WSNs, congestion might also occur due to the topological orientation of the sink node as discussed in the next section.

2.3 Possible scenarios of congestion in WSNs

A WSN with a relatively large number of sensor nodes with multiple sources that are transmitting the data packets simultaneously to the sink may create congestion hotspots when the traffic load exceeds a particular capacity of the sensor node. In the case of single sink networks, congestion might occur when multiple sources use the same route to send the data packets or when multiple sources are within a single transmission range and try to send the data to the sink as shown in Fig. 1a.

Another possible occurrence of congestion hotspot is near the sink node in a single sink network environment. Generally, this type of congestion is due to the traffic convergence. The possibility of congestion near to the sink is usually because of the sink node placement at the edge of the network instead at the centre, as shown in Fig. 1b. Such type of congestion can be removed by introducing a mobile sink in

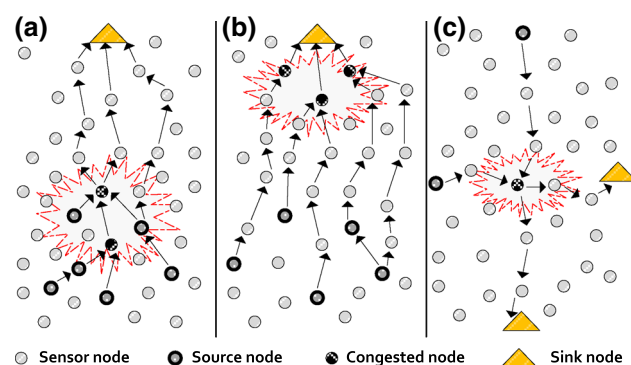


Fig. 1 The congestion hotspot scenarios in mono and multi-sink WSNs

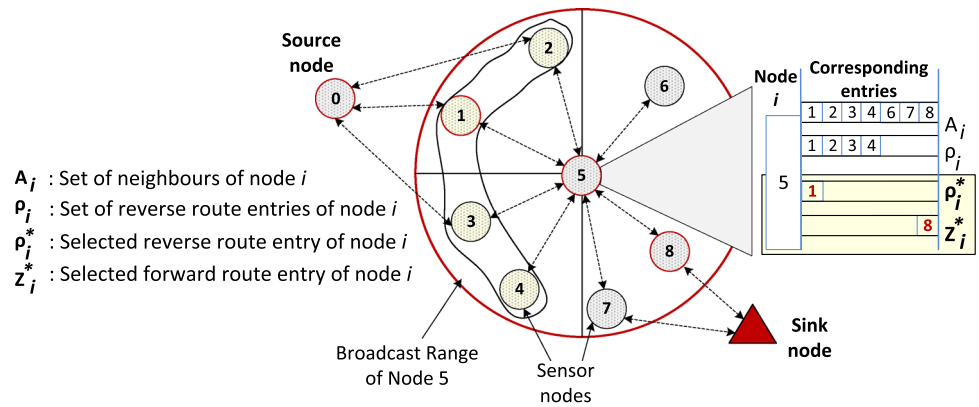
a static network. However, a single mobile sink is not sufficient due to the network scalability in WSNs [29] and the solutions with number of sinks increase the state complexity [30]. To avoid such complexity and the complication to define the mobility pattern of the mobile sink along with the high control overhead in WSN, all the sensor nodes are assumed static. In the case of a multi-sink network, the chances of congestion are due to the intersection hotspot. This type of scenario usually occurs when the intermediate node has the path to the sink node and the buffer capacity of that node exceeds its maximum limit as depicted in Fig. 1c. Based on the congestion scenarios shown in Fig. 1, a static network topology is considered with multiple sensor nodes and a single sink placed at the centre of the network in the proposed methodology. All the sensor nodes distributed in a sensor field have the equal significance and capabilities. However, the sink node is more powerful in resources such as, memory space, processing capabilities and energy.

3 Motivation

Generally, on-demand routing protocols are based on a single-criterion; however, better performance can be achieved by using a multi-criterion approach as discussed in [31]. Furthermore, the variants of such routing protocols in WSNs replace the existing routing metric with their proposed routing metric and keep the route discovery mechanism intact. Replacing the metric without modifying the route discovery procedure is not enough to increase the performance of the entire network because; it can generate ETE delay in finding the destination according to the desired metric(s). Moreover, the majority of these types of protocols selects the same route for the whole communication session, which can cause severe information loss due to congestion, and hence leads to performance degradation of the network. Additionally, during data forwarding, congestion can happen due to a high traffic rate, which is generally detected by notifying the source node in a timely manner. The source node receives the notification and alleviates it in a reactive manner by adjusting the source traffic rate. However, these types of mechanisms degrade the performance of the network due to the reduction of ETE throughput [28].

With these motivations, in this work, a routing scheme has been proposed that includes three different mechanisms. Firstly, it avoids congestion proactively. Secondly, it detects congestion in a timely manner and thirdly, it alleviates congestion in a reactive manner. Considering the features of WSNs, the proposed mechanism maintains a minimal amount of information without any extra overhead, such as by using periodic 1-Hop messages, or some localization techniques or some external entities like the GPS. The problem with most of the standardized on-demand routing mech-

Fig. 2 Broadcast range of a sensor node with reverse and forward route entries



anisms is the strictness in maintaining the forward route formation. This strictness has also been reduced in the proposed mechanism in order to provide flexibility in the form of multiple options as reverse route entries while selecting the best forward node during forward route formation. Moreover, congestion alleviation is carried out by applying a local ripple-based search mechanism, instead of adjusting the source traffic rate in order to maintain the ETE throughput and to increase the overall network performance.

The key motivation of this research work is to address the issues, which are not encountered in the classical on-demand routing protocols for WSNs. For this purpose, a novel Congestion-aware and Traffic Load balancing Scheme (CTLs) for routing has been proposed to avoid, detect, and alleviate congestion from the network.

4 Congestion-aware and traffic load balancing scheme

In this section, the proposed CTLs for routing is presented. CTLs contains the routing mechanism that is used to avoid, detect and alleviate congestion in order to balance the traffic load in WSNs. This section starts with the description of the network model and the assumptions made in a single sink network environment. Furthermore, an overview of the proposed CTLs is presented along with the system model. Finally, the detailed description of different components of the system with their functionalities is discussed.

4.1 Network model and assumptions

The WSN that has been considered in this research work consists of a flat topology deployed over a sensor field. Each sensor node deployed in the sensor field has a built-in a wireless transceiver and has been pre-programmed to identify itself through a unique node identifier or an Internet Protocol (IP) address. All the sensor nodes are assumed to have equal characteristics in terms of initial energy, processing capabil-

ities, sensing range, and memory. However, the sink node is considered to have higher capabilities in terms of storage and energy.

The network is modelled as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of nodes and \mathcal{L} is the set of wireless links. Each node in \mathcal{N} is equipped with a single omnidirectional antenna connected through a wireless link (i, j) such that, $(i, j) \in \mathcal{L}$. Each sensor node is able to maintain two different types of routing entries. These routing entries are the forward and the reverse route entries, formed during the route discovery procedure. Let ρ_i and Z_i be the set of reverse and forward route entries, respectively, of sensor node, i . Here, A_i is the set of neighbours of a node i such that, $A_i \subseteq \mathcal{N}$, $Z_i \subseteq A_i$ and $\rho_i \subseteq A_i$. Figure 2 shows some of the basic identifiers used in the detailed methodology, where, ρ_i^* and Z_i^* are the selected reverse and forward route entries at node, i . Let λ_i , η_i and Ω_i be the consumed energy, the participation level and the signal strength metrics of a node i , respectively. Here, η is the total number of active flows that a node accommodates in a particular time t . Each node in the sensor field is able to compute its λ , η and Ω in the network. The next section describes an abstract view of the proposed system model.

4.2 System model and overview of CTLs

To solve the congestion problem highlighted in Sects. 1 and 2, the proposed CTLs for routing is designed and developed with the aim to avoid, detect and alleviate congestion in order to balance the traffic load in WSNs. For this purpose, CTLs is divided into three major components. The first component in CTLs is used to avoid congestion proactively. In the second component, timely detection of congestion is carried out and the third component alleviates congestion reactively. An abstract view of the CTLs with these components is shown in Fig. 3. In the first component, a novel route discovery mechanism with reverse and forward route procedures is proposed. The second and third components provide the novel procedures to detect and alleviate congestion in a timely and

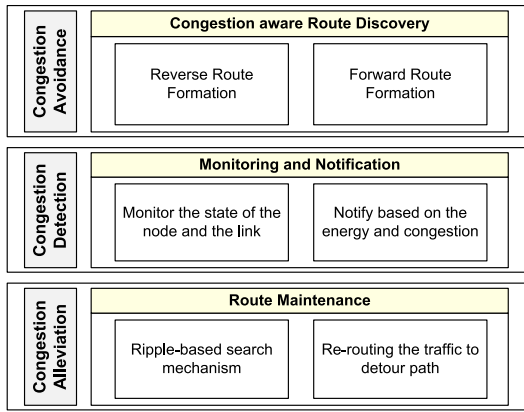


Fig. 3 An abstract view of the proposed mechanism

reactive manner by monitoring the buffer, channel utilization and traffic path shifting to an alternate route, respectively. The traffic path shifting is performed either using the ripple-based search mechanism to bypass the congested node/link or deviating the traffic to the sub-optimum least congested route. The proposed CTLS is a loop-free, single-path, on-demand reactive routing protocol that discovers the route whenever there is a demand from the application. The following is the description of the data flow model of the proposed CTLS.

4.3 Data flow model of the proposed CTLS

The proposed CTLS aims to provide a traffic load balanced environment through congestion avoidance, detection and alleviation. These components are integrated in each sensor node in a WSN. Initially, a source node is triggered by the application to send the data to the desired location. In order to perform this functionality, a sensor node performs a series of steps. These steps are modelled by using level 0, Data Flow Diagram (DFD) as shown in Fig. 4. The main functionalities of these components are as follows:

1. **Congestion Avoidance** When a source node is triggered by the application, the first step that is performed by the sensor node is to check the availability of the route to the desired location (destination) through a *Check Route Availability* process. If the route to the destination is not available in the routing table, then the sensing node broadcasts a request by incorporating the routing metric, which is determined through the process of the composite metric composition. The *Composite Metric Composition* process is invoked whenever a sensor node needs to send the broadcast request to determine the route to the desired location. To determine the composite metric, three different routing metrics are used

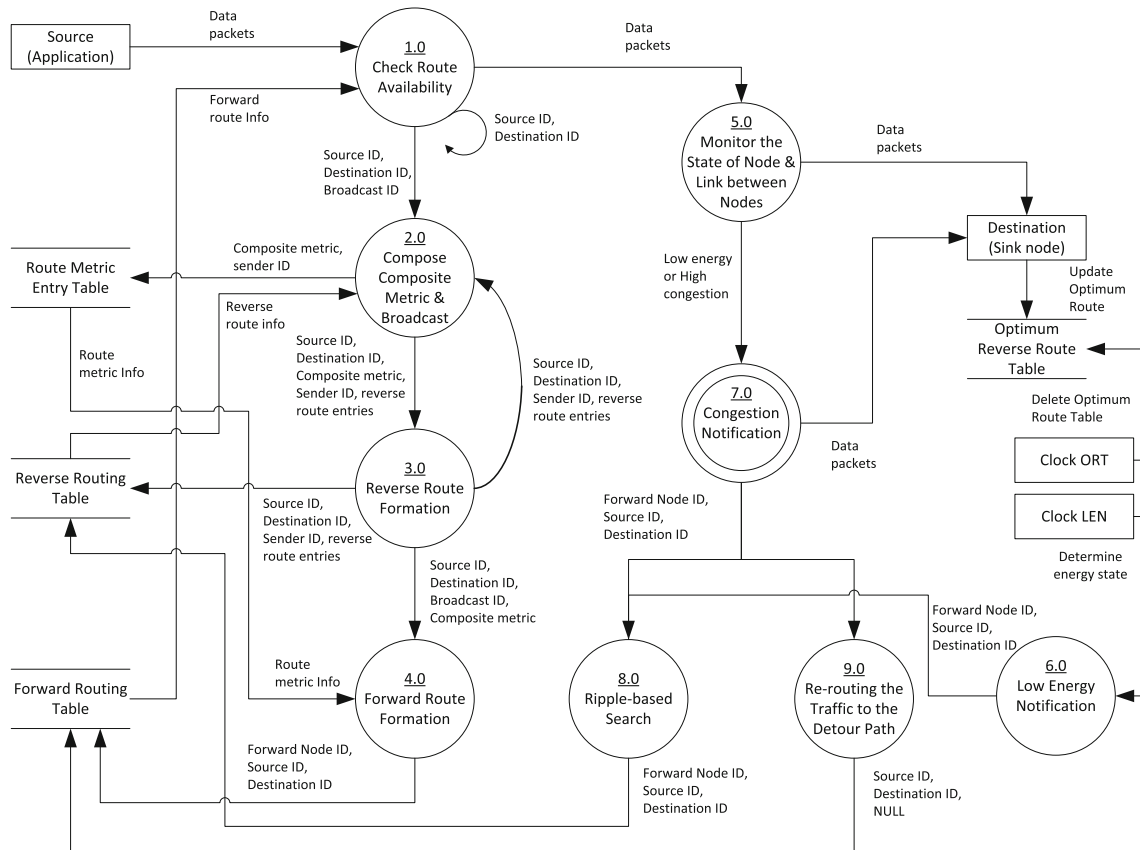


Fig. 4 Level 0, data flow diagram (DFD) of CTLS

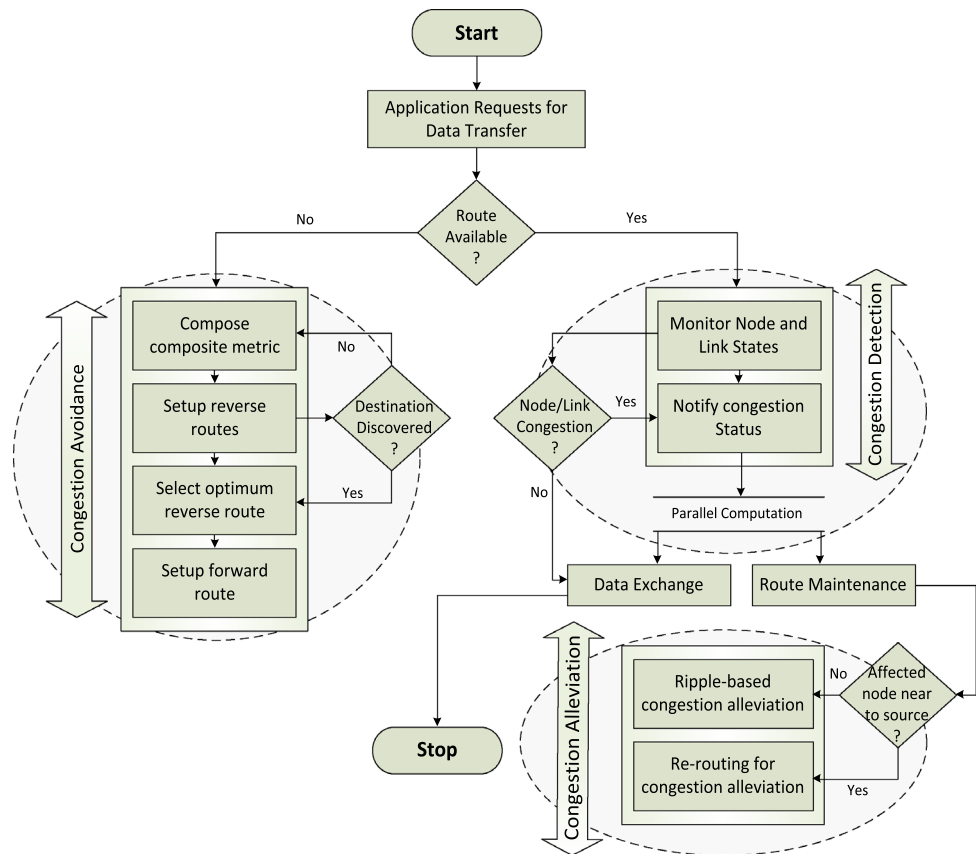
that exhibit the characteristics of the sensor nodes and the link between these nodes. The *Reverse Route Formation* process is started when an intermediate sensor node receives a broadcast request. It maintains a loop-free reverse route entry in the data store that is known as a Reverse Routing Table. During this process, when the desired location is determined, then the process of Forward Route Formation is carried out. The *Forward Route Formation* is invoked when a broadcast request is received at the destination. This process selects the optimum route and the optimum forward node based on the ETE composite metric maintained at each node. The Forward Route Formation maintains the selected node in the data store of the Forward Routing Table and sends a uni-cast reply to the selected forward node towards the source node. After this process, a route to the destination is available and the data packet is sent to the desired location over the discovered route. During data forwarding, the buffer and the link between the sensor nodes are monitored to detect congestion.

2. Congestion Detection A process to *Monitor the State* of the node and the link between the nodes is initiated in order to detect congestion. If the node or the link between the nodes seems that it will be congested in the near

future, then the process to notify the source or the predecessor node is triggered. The *Congestion Notification* process is invoked by the sensor node when congestion or low energy is detected. As per the position of the affected node, a notification is sent either to bypass the affected area or re-route the data traffic to an alternate route in order to alleviate congestion.

3. Congestion Alleviation A *Ripple-based Search* is activated when a sensor node receives a notification message. In this process, the congested node or the link is bypassed in order to maintain the route. Another procedure to alleviate congestion is to *Re-Route* the data traffic to an alternate congestion-aware and energy efficient route. This process is also initiated by the node after determining the approximate position of the congested or the affected node over the active route. An event-based *Optimized Route Table (ORT)* clock (T_{ORT}) that is used to maintain and discard the entries available for optimum route selection is maintained at the sink. The value of T_{ORT} is adaptive as per the number of flows passing through a particular sensor node. CTLS also consider this procedure, when the residual energy of the nodes is less than or equal to 10% of the maximum energy. The sequence followed by different components in the proposed CTLS is shown in the Component Sequence Dia-

Fig. 5 Component flow diagram of CTLS



gram (CFD) in Fig. 5. In the next section, a detailed description of CTLS is presented.

4.4 Congestion avoidance in CTLS

Congestion avoidance in CTLS is mainly carried out through a novel route discovery procedure. In this procedure, the characteristics of the sensor nodes and the link quality between these nodes are determined in order to avoid congestion. These characteristics include the energy, traffic load of the sensor node and the quality of the link in terms of the signal strength. The proposed route discovery mechanism is initiated by the source node when there is a demand to send the data traffic. This route discovery mechanism has some novel aspects as compared to the standard on-demand reactive routing protocols. Firstly, it can have multiple requests to choose at the sink as per the orientation of the sensor nodes in the form of network topology to select the optimum route to the sink based on the hop count. Secondly, it temporarily maintains multiple options (sensor nodes) at each node during the route discovery process and selects the best option as per the routing criteria given in the route request message. The other entries except the selected ones are purged from the routing table to make CTLS a less loaded single path routing protocol. Finally, instead of only considering the hop count criterion to select the route, CTLS provides the flexibility to use any of the routing metrics, applicable to perform routing.

4.4.1 Route discovery procedure of CTLS

The route discovery procedure of CTLS is inspired by those routing protocols that are reactive in nature. This means, whenever there is a need for an application to send the data, the routing protocol will discover the route. To do this, two types of entries are usually maintained in the routing table. These entries are the forward and the reverse route entries. The reverse route entries are maintained when a source node broadcasts the Route Request Message (RREQM) to discover the sink. The forward route entries are maintained when the sink sends a uni-cast Route Reply Message (RREP) to the source. Generally, the reverse route entries in the standard routing protocols are maintained, based on the source identifier, (S_{ID}), and broadcast identifier, (B_{ID}). Typically, in the on-demand routing, if the node has already received and maintained a request based on the S_{ID} and B_{ID} , then it will process and discard the subsequent requests received at the same node. In contrast to this, the intermediate nodes in CTLS, process and temporarily maintain the reverse route entries and at the same time prevent routing loops. There is no extra overhead in maintaining these entries except some processing cost, which is almost negligible as compared to the cost used in [12]. This feature gives more flexibility to select the best forward node among various alternatives at

each stage during the route discovery. The procedure to maintain the reverse entries and select the optimum ETE route based on the defined criteria is presented in Algorithm 1.

Algorithm 1: Maintain multiple loop-free reverse route entries to avoid congestion in WSNs

Initialization:

For each node $N_i \in N$
 $\rho_i = NULL; P_i = NULL; A_i = NULL;$
End For

Route Discovery:

```

01: Request to Send the Data
02: IF route exists
03:   Data exchange over the least congested route
04: ELSE
05:   FOR each node  $i \in N$ 
06:     Broadcast RREQM by node  $i$  // Along with  $\rho_i$ 
07:     IF ( $j == S_{ID}$  or  $j \in \rho_i$ ) //  $j$  is the receiver
08:       Discard this entry
09:     ELSE IF ( $j \notin \rho_i$  and  $j \neq D_{ID}$ )
10:       Add reverse entry at node  $j$  and maintain in  $\rho_j$  and  $P_j$ 
11:     ELSE IF ( $j == D_{ID}$ )
12:       Add reverse entry with node  $i$  at  $D$ , maintain in  $\rho_j, P_j$ 
13:       IF No. of nodes in  $A_D == 1$ 
14:         CALL: Optimized ETE reverse route selection
15:       ELSE
16:         Wait for multiple RREQM till timer  $T_{ORT}$  expires
17:       END IF
18:     END IF
19:   IF  $T_{ORT}$  expires
20:     CALL: Optimized ETE reverse route selection
21:   END IF
22: END FOR
23: END IF

```

In the proposed CTLS, the sink responds to the first route request received, if the forward node has sufficient resources to accommodate the traffic flow. The destination node can wait for multiple route requests based on the timer, T_{ORT} , to select the best option in the form of the optimum ETE routing cost. However, this waiting time usually creates an ETE delay as compared to other techniques that do not wait. The value of the waiting time is described in [32]. The intermediate nodes along the way back to the source select the appropriate forward nodes according to the instructions given in algorithm 2. The CTLS Route Request Message (RREQM) format contains various fields aligned with a 32-bit header format as shown in Fig. 6. The first 8-bit value identifies the type of the message as shown in Table 1. The *Routing Metric* field in the RREQM format shows the routing criteria used to select the next hop node during the route discovery. In CTLS, the routing metric is based on the composite metric. The sink does not respond to the RREQM, if the remaining energy of the next hop node is less than the threshold value.

The subcomponents of the route discovery involves two main procedures to avoid congestion proactively, these components are the Reverse Routes Formation (RRF) and the Forward Route Formation (FRF). During RRF, the proposed mechanism determines the routing metrics to form the composite metric and avoid routing loops. Furthermore, it also discovers the sink node over the shortest route based on the

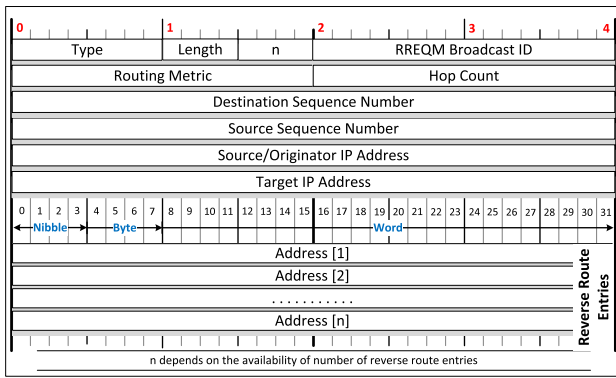


Fig. 6 Route request message (RREQM) Format

Table 1 Message types to discover the congestion-aware route in CTLS

CTLS route discovery message types	Value
Route request message (RREQM)	1
Route reply message (RREPM)	2
Congestion/low-energy notification (CLEN)	3

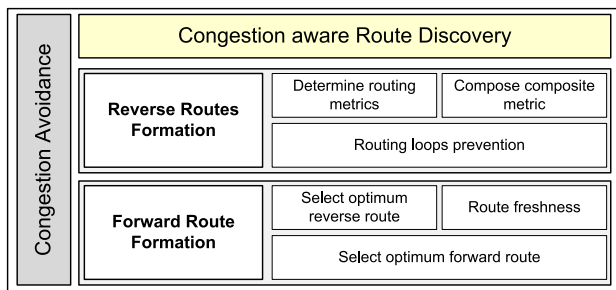


Fig. 7 Congestion avoidance based on congestion aware route discovery

hop count criterion. During FRF, the sink node responds over the shortest route and selects the optimum forward sensor node along the way to the source to discover the least congested and energy-efficient ETE route. The subcomponents to avoid congestion are shown in Fig. 7. The detailed procedure of congestion avoidance in CTLS through route the discovery mechanism is discussed in the following subsections.

4.4.2 Reverse routes formation

The CTLS broadcasts the RREQM along with the routing-metric and reverse-route-entries fields available in the routing table. Initially, for a particular SD pair, there is no entry available at the source node; so, there will be no identifier stored in the reverse-route-entry field. On receiving the RREQM from the intermediate node, the receiving node will check whether it has already received the RREQM against the same S_{ID} and B_{ID}. If the node has received the RREQM for the very first time, it will add the reverse route entry into the reverse routing table. However, if RREQM has already

been received with the same S_{ID} and B_{ID} then, irrespective of the previous hop, the receiving node will compare its own ID with the entries present in the reverse-route-entries field of the received RREQM.

If the node ID of the receiving node matches with any entry available in the reverse-route-entries field, then it simply discards the request to make it loop-free. However, if its own ID does not match with the entries in the reverse-route-entries field, then it adds and maintains the entry based on the source, destination and previous hop identifiers. In this procedure, the destination and the intermediate node(s) can have multiple entries available in their routing tables. Based on these entries, the sensor node can select the best option amongst various alternatives during FRF as discussed in [33] in order to forward the data packets along the optimum route to the destination.

Determine routing metrics for CTLS Most of the classical routing protocols in multi-hop wireless networks discussed in the literature have considered a single routing metric to discover the ETE route. The discovered route based on a single routing criterion (minimum hop count) might cause severe packet loss and excessive energy consumption due to congestion. The proposed CTLS considers four different routing metrics for the selection of a route to forward the data packet. These metrics are the hop count, H, consumed energy, λ, the participation level, η and the signal strength, Ω. Among these four routing metrics, H is used to discover the sink over the shortest possible distance from the source node based on the hop count criterion; whereas λ, η and Ω exhibit the characteristics of the node and the wireless channel to select the best forward node(s) along the way to the source node. The description of these routing metrics and the determination of the composite routing metric are explained in the following subsections.

1. Hop Count The hop count, H shows the distance from a specific node to another node in terms of the number of nodes along the path. This metric is an important factor to evaluate the performance of the protocol. It is assumed that with the decrease in the number of hops along the route, the ETE delay experienced by the protocol decreases with reference to a single flow. In CTLS, the hop count is not used as the selection criterion to select the forward sensor nodes but, it is utilized to discover the sink node over the shortest route so that the protocol experiences less ETE delay in finding the sink node. The objective function that represents H is expressed as follows.

$$\phi_H(S, D) = \min_{r \in \mathcal{R}_{SD}} |\mathcal{N}'_{r,S,D}| \tag{1}$$

In (1), $\phi_H(S, D)$ is an objective function that is used to find the route with the minimum number of hops from S

to \mathcal{D} . Here, $\mathcal{N}'_{r,S,\mathcal{D}}$ is the set of nodes in a specific route, r , for a specific \mathcal{SD} pair such that, $\mathcal{N}'_{r,S,\mathcal{D}} \subseteq \mathcal{N}$. $|\mathcal{N}'_{r,S,\mathcal{D}}|$ represents the cardinality (number of elements) of a set and $r \in \mathcal{R}_{SD}$ represents a route among all the possible routes \mathcal{R} from S to \mathcal{D} such that, $\mathcal{R}_{SD} \in \mathcal{R}$. The objective is to find a best route with the minimum hop count from S to \mathcal{D} , represented as follows.

$$r^*_{HS,\mathcal{D}} = \underset{r \in \mathcal{R}_{SD}}{\operatorname{argmin}} |\mathcal{N}'_{r,S,\mathcal{D}}| \tag{2}$$

In (2), $r^*_{HS,\mathcal{D}}$ shows a selected route with the minimum number of hops from S to \mathcal{D} . A sink node is discovered based on (2) during the RRF. After discovering the sink node, the forward nodes are selected during FRF, based on the other routing metrics discussed as follows.

2. Consumed Energy of the Node The energy parameter is an important performance measure to evaluate the protocol, especially in multi-hop WSNs. This is because WSNs are energy-constrained networks in which the efficient utilization of energy increases the network lifetime and decreases the energy consumption per data packet. The equation to express the energy metric, λ , is shown as follows.

$$\lambda = \frac{e_c}{e_{\max}} \tag{3}$$

In (3), the energy metric, λ , is computed as the ratio of the energy consumed, (e_c), to the initial energy, (e_{\max}), of the node. In this work, a weak node is defined as a node with remaining energy, less than 10% of the initial energy as expressed in the following condition.

$$(e_{\max} - e_c) < 0.1(e_{\max}) \tag{4}$$

The node satisfying the condition shown in (4) does not participate in data forwarding because, it is considered as a weak node [34] that can lead to network partitioning. The objective function for λ can be expressed as follows.

$$\phi_\lambda(\mathcal{S}, \mathcal{D}) = \min_{r \in \mathcal{R}_{SD}} \left(\sum_{i \in \mathcal{N}'_r \setminus \mathcal{D}} \lambda_{i,r,\mathcal{S},\mathcal{D}} \right) \tag{5}$$

In (5), $\phi_\lambda(\mathcal{S}, \mathcal{D})$ is an objective function to get the minimum energy consumption route, r , from \mathcal{S} to \mathcal{D} . Here, $\lambda_{i,r,\mathcal{S},\mathcal{D}}$ represents the energy consumed by a node, i in a particular route r , identified by a set of nodes, \mathcal{N}'_r , from \mathcal{S} to \mathcal{D} . The objective is to find the route, with the minimum consumed energy amongst all the possible routes, \mathcal{R}_{SD} , represented as follows.

$$r^*_{\lambda,\mathcal{S},\mathcal{D}} = \underset{r \in \mathcal{R}_{SD}}{\operatorname{argmin}} \left(\sum_{i \in \mathcal{N}'_r \setminus \mathcal{D}} \lambda_{i,r,\mathcal{S},\mathcal{D}} \right) \tag{6}$$

In (6), $r^*_{\lambda,\mathcal{S},\mathcal{D}}$ represents the selected route, that corresponds to the minimum energy consumption route amongst all the routes, \mathcal{R}_{SD} , for a pair of \mathcal{S} and \mathcal{D} .

3. Participation Level The participation level, η , is used to monitor and avoid node level congestion. It indicates the number of active flows passing through a node. The equation that represents the participation level metric, η , is given below.

$$\eta = \frac{\eta_c}{\eta_{\max}} \tag{7}$$

In (7), η_c represents the current number of flows passing through a node and η_{\max} is the total number of sources in the network. A node with greater number of flows is more prone to congestion as compared to less number of flows. The objective function for η can be expressed as follows.

$$\phi_\eta(\mathcal{S}, \mathcal{D}) = \min_{r \in \mathcal{R}_{SD}} \left(\sum_{i \in \mathcal{N}'_r \setminus \mathcal{D}} \eta_{i,r,\mathcal{S},\mathcal{D}} \right) \tag{8}$$

In (8), $\phi_\eta(\mathcal{S}, \mathcal{D})$ is an objective function to represent the minimum number of the participation level of the nodes along a route, r , from \mathcal{S} to \mathcal{D} . Here, $\eta_{i,r,\mathcal{S},\mathcal{D}}$ represents the participation level at node, i in a particular route r , identified by a set of nodes, \mathcal{N}'_r . The objective is to find the route with the minimum participation level route among all the possible routes, \mathcal{R}_{SD} , represented as follows.

$$r^*_{\eta,\mathcal{S},\mathcal{D}} = \underset{r \in \mathcal{R}_{SD}}{\operatorname{argmin}} \left(\sum_{i \in \mathcal{N}'_r \setminus \mathcal{D}} \eta_{i,r,\mathcal{S},\mathcal{D}} \right) \tag{9}$$

In (9), $r^*_{\eta,\mathcal{S},\mathcal{D}}$ corresponds to the selected ETE route based on the participation level for a particular \mathcal{SD} pair.

4. Received Signal Strength The Received Signal Strength, Ω , is used to represent the link quality. The higher the signal strength, the better is the link quality. It is calculated as follows.

$$\Omega = \frac{I_{RSS}}{I_{R_xThresh}} \tag{10}$$

In (10), I_{RSS} is the received signal strength indicator and $I_{R_xThresh}$ is the receiver sensitivity that is the signal level based on the maximum available distance between the sensor nodes. The nodes beyond the maximum distance as per the threshold value cannot receive the signal from its neighbouring nodes. The objective function that represents Ω can be expressed as follows.

$$\phi_\Omega(\mathcal{S}, \mathcal{D}) = \max_{r \in \mathcal{R}_{SD}} \left(\sum_{(i,j) \in \mathcal{L}_r} \Omega_{(i,j),r,\mathcal{S},\mathcal{D}} \right) \tag{11}$$

In (11), $\phi_{\Omega}(\mathcal{S}, \mathcal{D})$ is an objective function that represents the route, r , with the maximum ETE signal strength route, from \mathcal{S} to \mathcal{D} . Here, $\Omega_{(i,j),r,\mathcal{S},\mathcal{D}}$ represents the signal strength of a link (i, j) where, $(i, j) \in \mathcal{L}_r$ and \mathcal{L}_r is the set of links available in a specific route, r . The objective is to select the route with the maximum signal strength represented as follows.

$$r_{\Omega_{\mathcal{S},\mathcal{D}}}^* = \underset{r \in \mathcal{R}_{\mathcal{S}\mathcal{D}}}{\operatorname{argmax}} \left(\sum_{(i,j) \in \mathcal{L}_r} (\Omega_{(i,j),r,\mathcal{S},\mathcal{D}}) \right) \tag{12}$$

In (12), $r_{\Omega_{\mathcal{S},\mathcal{D}}}^*$ represents the selected route with the maximum signal strength compared to the other available routes, $\mathcal{R}_{\mathcal{S}\mathcal{D}}$, for a particular $\mathcal{S}\mathcal{D}$ pair.

Composition of composite metric There are two major approaches available in the literature for multi-metric composition. The first one is the lexical approach and the second one is the additive metric composition approach [35]. In the lexical metric composition, the decision is based on the hierarchy or priority of the different metrics. This means that the routing metric to select the sensor node is first computed as per the highest priority metric; and, in the case of a tie, the second metric is compared and so on. On the other hand, the additive metric composition approach offers more flexibility. With this approach, the priorities are generally assigned based on the weighting factors in accordance with the user preferences or the application requirements. Instead of using the lexical approach where a sensor node must contain all the fields against each routing metric in the routing table, an additive metric composition approach is a better choice where a single field is required for maintaining the routing cost.

To make a composite metric from the above defined routing metrics, represented in (3), (7), and (10), the weighted additive metric composition approach is used. In the proposed CTLS, the sink is discovered based on (2) and the other metrics mentioned in (3), (7) and (10), are used to select the best forward node. The detailed procedure to select the forward node is explained in the next section. As three different routing metrics have been considered to select the forward node, the constants as weights or the preference levels corresponding to each variable λ , η and Ω are α , β and γ , respectively, such that $\alpha + \beta + \gamma = 1$. The total cost, C_{cost} , for the composite metric composition can be represented as:

$$C_{cost} = (\alpha \times \lambda) + (\beta \times \eta) + \left(\gamma \times \frac{1}{\Omega} \right) \tag{13}$$

In (13), Ω is used in the denominator; this is because the selected node is required to have the minimum consumed energy, and participation level and the maximum signal strength. So, to cope with this criterion, Ω is used in the denominator to determine the maximum value in the form of the minimum cost.

Each metric in (13) is the normalized metric with reference to its respective maximum/minimum achievable value. As in (13), three different routing metrics are used; so, based on the total number of possibilities against three inputs there can be eight different possible combinations as output. These combinations are based on the application requirements as discussed in [32]. In the case of the composite metric, the objective is to select the route with the minimum cost based on the multi-objective criteria represented as follows.

$$\phi_{\lambda,\eta,\Omega}(\mathcal{S}, \mathcal{D}) = \min_{r \in \mathcal{R}_{\mathcal{S}\mathcal{D}}} \left(\sum_{i \in \mathcal{N}'_r \setminus \mathcal{D}} \left((\alpha \times \lambda_i) + (\beta \times \eta_i) \right) + \left(\sum_{(i,j) \in \mathcal{L}_r} \gamma \times \left(\frac{1}{(\Omega_{i,j})} \right) \right) \right) \tag{14}$$

In (14), $\phi_{\lambda,\eta,\Omega}(\mathcal{S}, \mathcal{D})$ is an objective function to represent the route with the minimum ETE cost from \mathcal{S} to \mathcal{D} . This cost integrates all the three parameters including λ , η and Ω . Here, the objective is to select the optimum minimum cost route represented as follows.

$$r_{(\lambda,\eta,\Omega)_{\mathcal{S},\mathcal{D}}}^* = \underset{r \in \mathcal{R}_{\mathcal{S}\mathcal{D}}}{\operatorname{argmin}} \left[\sum_{i \in \mathcal{N}'_r \setminus \mathcal{D}} \left((\alpha \times \lambda_i) + (\beta \times \eta_i) \right) + \left(\sum_{(i,j) \in \mathcal{L}_r} \gamma \times \left(\frac{1}{(\Omega_{i,j})} \right) \right) \right] \tag{15}$$

In (15), $r_{(\lambda,\eta,\Omega)_{\mathcal{S},\mathcal{D}}}^*$ is the objective function that corresponds to the optimum ETE route, r , with the optimum cost among all the possible costs from \mathcal{S} to \mathcal{D} .

Prevention of routing loops In CTLS, previous hops are maintained according to the reverse route entries available in the RREQM. However, considering only the previous hop along with the \mathcal{S}_{ID} and \mathcal{B}_{ID} might create a routing loop as depicted in Fig. 8. Here, node J is the source node and \mathcal{D} is considered as the destination. With reference to Fig. 8a, suppose that J initiates the RREQM, which will be received by I . Since node I is not the destination, it will again rebroadcast this RREQM with the new entries, which will be received by nodes $J \rightarrow K \rightarrow L \rightarrow M$ and \mathcal{D} . Node

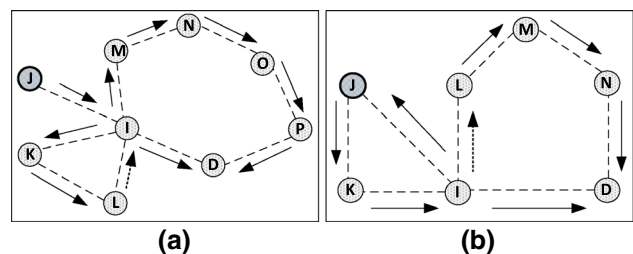


Fig. 8 Example of routing loops based on previous hop

J will simply drop this request because it is the source node. All the other nodes will rebroadcast the RREQM except \mathcal{D} . Here, \mathcal{D} can wait for multiple RREQMs to be received as discussed earlier in CTLS route discovery procedure. Now, K will rebroadcast the RREQM, which will be received by L and I . As I is the initiator node for this rebroadcast, it will discard the RREQM sent by K . However, L will receive this rebroadcast message. Now, after receiving the RREQM from L , it will again rebroadcast the RREQM and this will be received by node I . Here, node I will create a routing loop via $I \rightarrow K \rightarrow L$ and then I . The same applies to the case shown in Fig. 8b, where node I creates a loop via $J \rightarrow K \rightarrow I$ and then J . Therefore, to overcome this problem, the following condition is used, which creates a loop free entry.

$$if (J \notin \rho_I) \text{ then } nexthop_I = J \tag{16}$$

In (16), J is a node ID that receives the RREQM from node I . In this condition ρ_I corresponds to the reverse routing entries available in the routing table of node I .

Forward route formation When the sink receives the RREQM, it adds/updates and maintains the previous hop entry. The proposed CTLS responds to the first RREQM received at the sink without any delay and maintain a forward route entry with the node from where the RREQM is received, if and only if the energy level of the forward node is above the threshold value. The sink maintains the ETE cost in the Optimum reverse route table (ORT). The entries in the ORT are maintained until the timer, T_{ORT} , expires. The timer, T_{ORT} , depends on either the number of neighbours of the sink and/or the requirement defined by the WSN application. However, based on the network characteristics, the optimum time to wait for multiple requests is very critical.

The FRF has two sub-procedures, firstly, it acknowledges the optimum path from where the first RREQM is received at the sink; and secondly, it selects the optimum node while going back to the source in order to discover the ETE least congested and energy-efficient route as presented in Algorithm 2. Based on the entries received during RREQM, a sensor node is able to select the best node as per the defined criteria among various nodes from the reverse routing table and set up the forward entry. After selecting the best forward node, CTLS purges all the other entries from the routing table to maintain a single entry path for a particular SD pair. The RREPM format contains various fields in order to maintain the forward route entry as shown in Fig. 9. In order to align with the 32-bit header format, a reserved field has been introduced for the future reference.

Algorithm 2: Pseudocode for an optimized ETE reverse route selection based on RREQMs at the sink

```

Notations:
 $C_{ETE}$       ETE Cost during CTLS route request procedure
 $C_{cached}$    Cached ETE cost during the route selection at sink node
 $S_{ID}$  and  $B_{ID}$  Source identifier and broadcast request identifier
 $T_{ORT}$       Timer for the Optimum Route Table (ORT)
01: IF RREQM with same  $S_{ID}$  and  $B_{ID}$  exists in ORT and
     $C_{ETE} > C_{cached}$  then
02:   Discard the RREQM message
03: ELSE IF RREQM with same  $S_{ID}$  and  $B_{ID}$  exists and
     $C_{ETE} \leq C_{cached}$  then
04:   CALL: Cooperative forward node selection procedure
05: ELSE
06:   IF  $T_{ORT}$  is off then
07:     start  $T_{ORT}$  and cache the  $C_{ETE}$ 
08:   CALL: Cooperative forward node selection procedure
09:   END IF
10: END IF
11: IF  $T_{ORT}$  expires then
12:   Purge other entries from ORT as per  $S_{ID}$  and  $B_{ID}$ 
13:   CALL: Cooperative forward node selection procedure
14: END IF
    
```

During RREPM, the forward sensor nodes along the way to the source nodes are selected according to the cost expressed in (15). The forward node at each stage (Intermediate sensor nodes) from the sink to the source during the FRF is based on the composite metric. This cost is computed based on the three different routing criteria that integrate the cost of the consumed energy, λ , the participation level, η , and the signal strength, Ω . The procedure to compute the routing cost has already been explained in the compute composite metric section. During this procedure, only a single entry of the forward route is maintained at each intermediate node against a particular SD pair. All the reverse route entries are purged except the optimum selected entry. This procedure is adapted to make a single entry routing protocol that avoids the extra overhead of maintaining multiple routing entries. The detailed procedure to select the optimum forward sensor node is presented in Algorithm 3. Based on this algorithm, the optimum sensor nodes are selected that leads to an energy-efficient and congestion-aware route as discussed in the following text.

Optimum route selection In CTLS, the first RREQM is maintained at the sink and is replied back with the RREPM to the source. However, the subsequent RREQMs are also checked at the sink, if the new received request is better than the old maintained request, then it is updated and replied back to the

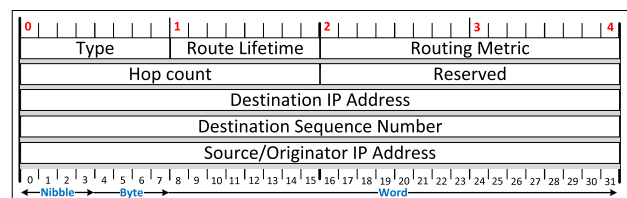


Fig. 9 Route response message (RREPM) Format

source. The requests maintained at the destination node(s) are discarded after the expiration of T_{ORT} . In some of the cases, it has been analysed that the optimum subsequent requests replied after the first route reply will not reach the source node. This is because these replies usually manage to merge at some common point with the already discovered least congested path while going towards the source. During the process of the FRF, the route freshness is also carried out based on the destination sequence number as discussed in the following subsection.

Algorithm 3: Optimized ETE cost for sensor node selection as per the composite metric using Equation (15)

Notations:	
A_i	Set of Neighbours of a node, i
ρ_i	Set of reverse route entries of node, i
S_{ID}, D_{ID}	The identifiers of the source and sink nodes, respectively
C_i	The cost based on the routing criteria of node, i
C_{ETE_i}	The ETE cost received and maintained at node, i
P_S	The selected ETE path
$Z_{j_i}^*$	The selected forward sensor node
T_{C_i}	The sum of C_{ETE_i} and C_i , based on eqs. (3), (7) and (10).

RREQM Phase: To maintain the routing cost	
01:	FOR each node $i \in N$
02:	Broadcast $RREQM$ by node i
03:	$j \in (A_i - \rho_i)$ // J is the node that receives the RREQM
04:	IF $(A_i \cap \rho_i) = \emptyset$
05:	$T_{C_i} = C_i$
06:	ELSE IF order of $(A_i \cap \rho_i) == 1$
07:	$T_{C_i} = C_{ETE_i} \times C_i$
08:	ELSE IF order of $(A_i \cap \rho_i) > 1$
09:	$j_i^* = \underset{j \in (A_i \cap \rho_i)}{\operatorname{argmin}} C_{ETE_i}$
10:	$T_{C_i} = C_{ETE_{j_i^*}} \times C_i$
11:	END IF
12:	Add T_{C_i} in $RREQM$
13:	END FOR

RREPM Phase: Cooperative forward node selection	
14:	DO WHILE $j_i^* \neq S_{ID}$ // Process starts from the sink node
15:	IF $A_i \notin S_{ID}$
16:	$j_i^* = \underset{j \in \rho_j}{\operatorname{argmin}} (C_{cost}(i, j))$ // j_i^* is the selected forward node from ρ_j , where ρ_j , is the set of reverse route entries
17:	ELSE
18:	$j_i^* = S_{ID}$
19:	END IF
20:	Add Forward Entry with j_i^* at node i and maintain $Z_{j_i^*}, P_S$
21:	Unicast Route Reply Message (RREPM) to j_i^*
22:	END WHILE
23:	IF $j_i^* == S_{ID}$
24:	Send data packets along P_S
25:	END IF

Route freshness In most of the reactive routing protocols, a monotonous increasing destination sequence number is used for route freshness. The destination sequence number is the highest sequence number maintained in each RREQM. The higher the destination sequence number in the RREQM, the more recent is the routing information. In CTLS, the destination sequence number is the same as used in [12]; however, along with the destination sequence number, the state of the node and the cost of the ETE route are also considered. If

the new discovered route is better than the old route in terms of the energy-efficiency and congestion, then the new fresh route is adapted otherwise it is discarded. The entry against the destination sequence number is updated whenever a node receives RREQM or RREPM.

4.5 Congestion detection in CTLS

The detection of congestion in a timely manner is very important for efficient network resource utilization in order to balance the traffic load. CTLS detects congestion by using two different procedural steps. The first procedure monitors the status of the buffer and the wireless link quality. The second procedure sends a notification message against the detected event with reference to the traffic congestion or low energy of the sensor node. The detailed procedure to detect and notify congestion is shown in Fig. 10 and discussed as follows.

4.5.1 Monitoring the state of sensor node and wireless channel

During data forwarding, the nodes on an active route, monitor their energy consumption and congestion status. The energy consumption is monitored in order to make CTLS, energy aware. In parallel to this, congestion status of the sensor node and the link between the nodes is monitored with reference to its buffer occupancy and channel utilization, respectively. If congestion occurs inevitably due to high traffic load or traffic convergence at a particular sensor node during data exchange, it will be notified to the predecessor nodes, which are at least 2-hops away in order to bypass the congested node/link. In the proposed CTLS, both node and link level congestion behaviour are detected and notified in a timely manner. This is because the node level congestion increases the queuing delay that results in severe packet loss which increases the energy consumption per data packet. However, the link level congestion results in re-transmission of the data packets due to the unavailability of a wireless medium. The detailed description of congestion detection procedure is explained in the following subsection.

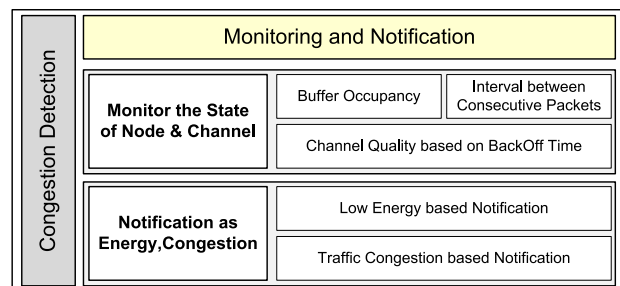


Fig. 10 Congestion detection in CTLS

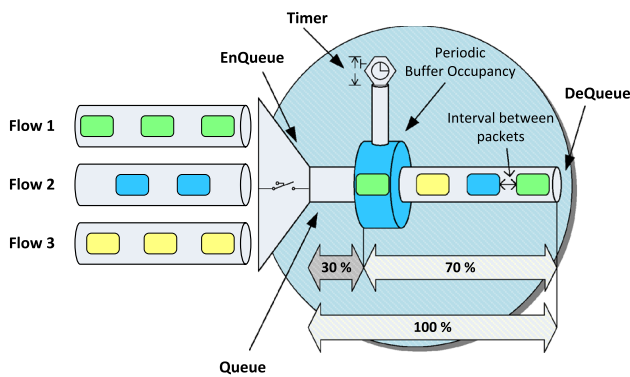


Fig. 11 Monitor the buffer occupancy of a sensor node.

Monitoring the energy status of the node The energy of a sensor node is monitored during data forwarding. If the energy status of the node is greater than the threshold value, then the data will be exchanged without any low energy notification. However, if the energy status falls below the threshold value, then the predecessor nodes are being informed through the notification process in order to increase the lifetime of the network. The lifetime is the time at which the first node runs out of its energy reserve. The energy consumption in CTLS is not monitored periodically; instead, it is monitored against the transmission or reception of the packets.

Monitoring the buffer occupancy Two different strategies are developed to monitor/check the buffer status of the node in order to measure the node level congestion. Initially, the buffer occupancy of the node is checked till its 70% utilization [18] and is represented as follows.

$$C_{rem} > 30\% (C_{max}) \text{ or } C_c < 70\% (C_{max}) \quad (17)$$

In (17), C_{max} is the maximum capacity, C_{rem} is the remaining capacity and C_c is the consumed capacity of a buffer. If C_c is less than 70% of its C_{max} , then the sensor node is considered to be in a safe state as shown in Fig. 11. However, if the C_c become greater than C_{max} , then a notification to bypass the interval between the consecutive packets is monitored as discussed in the following subsection.

Monitoring the interval between consecutive packets The interval between the consecutive packets is monitored, if the buffer occupancy reaches 70% of its maximum capacity. When a node receives the packets and $C_c > 70\%(C_{max})$ then CTLS monitors the gap/interval in terms of packet inter arrival time between these packets. If the interval is large, then the node is considered to be in an alert state; however, if the interval is small, then a sensor node is considered to be in a congested state and a notification process is triggered to alleviate congestion. To monitor the interval between the two consecutive packets, an Exponential Weighted Moving Average procedure is adapted, expressed as follows.

$$interval = (1 - X) \times interval_{prv} - X \times interval_{new} \quad (18)$$

Equation (18) represents the exponential smoothing function calculated by using an Exponential Weighted Moving Average (EWMA) procedure to remove the jittering effect. Here, X is the *EWMA-smoothing-factor* in the range of 0 to 1 and $interval_{prv}$ and $interval_{new}$ are used to represent the received time of the previous and the new packet, respectively.

Monitoring the channel quality The channel utilization in order to detect the link level congestion in the proposed CTLS is based on the number of times a node goes into the Backoff stage during the DATA-ACK procedure of CSMA/CA. A Backoff stage is the waiting stage in which a node waits for a particular random time interval when the channel is busy and cannot be accessed. This might be due to the fact that the other neighbouring nodes have occupied the channel. The more a sensor node goes into the Backoff stage, the more is the chance that the other neighbouring nodes have occupied the channel to transmit their packets; hence, it infers a high chance of link or channel utilization. Based on this mechanism, link level congestion is detected and is notified to the previous hop nodes in order to bypass the congested link based on the Ripple-based search or to re-route the traffic over an alternate route. The Backoff limit to mitigate congestion depends on the maximum number of retries for a packet. In this procedure, it is considered as 5 by considering that the maximum re-transmission against a packet is 7 as discussed in [36].

4.5.2 Notification based on congestion and low-energy

The proposed CTLS sends the notification message either by detecting the low-energy or by congestion status of the sensor node. Both of these notifications are sent to the predecessor nodes in order to maintain the route from the source to the sink. Based on the notification, the predecessor nodes reactively alleviate congestion. The notification message is represented as the Congestion/Low-Energy Notification (CLEN). The CLEN message is sent towards the previous hop nodes on an active route to bypass the congested or the weak node. To identify the notification whether it is due to the weak node or due to the congested, a type field is defined in the message format. If its value is 1, then it is considered that the notification is due to congestion; otherwise, the notification is due to low-energy. In the CLEN packet, a flag bit Y is used to recover the route from congestion without deleting the routing entries. The field *Destination Count* is used to maintain the number of routes affected due to con-

gestion. The procedure of sending the CLEN with reference to congestion is presented in Algorithm 4.

Algorithm 4: Congestion and Low Energy Notification

Notations:

I_{CDP}	Interval between two Consecutive Data Packets (CDP)
I_{min}	Minimum interval (0.25) between the two CDP
$E_{consumed}$	Energy consumed by the sensor node
E_{Thresh}	Maximum consumed energy of the node or minimum residual energy of the node

Congestion or low-energy notification

If any one of the event occurred during data transfer:

1. Variation in the traffic load estimation exceeds the capacity
2. Packet Loss due to high traffic load at the intermediate nodes
3. The residual energy of a node is less than 10% of the initial energy

```

01: IF  $I_{CDP} < I_{min}$  or  $E_{consumed} \geq E_{Thresh}$  then
02:   IF Congested/weak node is closer to the source and the
      predecessor node is about to congest then
03:     Discover a new least congested and energy efficient route
04:   ELSE IF Congested/weak node is at the middle or near to
      the sink then
05:     Call: Ripple-based search to bypass the congested node
      or weak node
06:   END IF
07:   Send CLEN towards the predecessor node(s)
08: END IF
    
```

Error notification based on low energy The energy level of the sensor nodes is notified based on the defined threshold value. If the remaining energy of the sensor nodes along the active route is above the threshold, then the data packets will be forwarded to the sink without any low energy notification. However, if the energy level of the node falls below the threshold value, then a CLEN message is composed and sent to the predecessor node(s) with a value as 0 in the type field of CLEN packet. When the predecessor node receives this CLEN message, it will first check the type of the message and the corresponding flag bit, Y . If the value in the type field is equal to 0, then the predecessor node(s) tries to bypass the low-energy sensor node in order to maintain the route r .

Error notification based on congestion If the sensor node that is expected to be congested in the near future, sends a CLEN message to its predecessor node(s) by assigning the value to the type field as 1. The CLEN message is a uni-cast message and the address of the predecessor node is fetched from the routing table of the sensor node. The receiver of this message will again re-send this message to its predecessor node. That is how, this message will propagate to 2-hops away from the affected node. It is a unicast message that is used to notify the predecessor nodes to bypass the congested area.

4.6 Congestion alleviation in CTLS

The route in the proposed CTLS is recovered reactively, if congestion occurs inevitably. This reactive procedure is man-

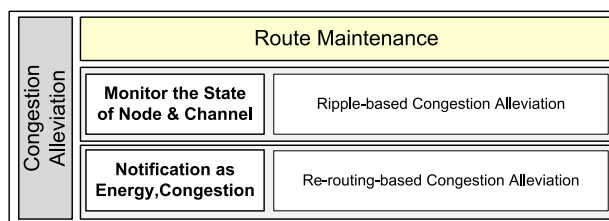


Fig. 12 Congestion alleviation during data forwarding

aged by using the congestion alleviation mechanism, which makes the network in an operational state. The proposed CTLS can be applied to any network with ad hoc nodes that communicate in a multi-hop fashion. Additionally, in CTLS, a network-based algorithm is proposed to control congestion; however, in the majority of the cases, congestion is alleviated by decreasing the source traffic rate. This rate adjustment is carried out in response to the notification received from the network but, this decreases the ETE throughput. In CTLS, both the node and the link level congestion are detected and controlled. A node that detects congestion in a timely manner is allowed to send the notification message to its predecessor nodes. Congestion in the proposed CTLS is controlled by either of the following two procedures as shown in Fig. 12. The first congestion control mechanism is the ripple-based search that bypasses the congested node and the second mechanism re-routes the data traffic over an alternate least congested route. The detailed description of these two mechanisms to alleviate congestion for route maintenance is explained in the following subsections.

4.6.1 Congestion alleviation based on the ripple-based search

The process to alleviate congestion in order to bypass the congested node/link or area using a ripple-based search. This mechanism is initiated by receiving a CLEN message from the affected node that has detected low-energy or congestion. Based on this notification, the predecessor node determines the position of the low-energy node or the node that is to be congested. The position of the node is determined by using the number of hops an affected node is away with respect to the source and the sink on an active route. If the affected node is closer to the sink or at the middle of the active route with respect to the number of hops, then a local repair strategy using the ripple-based search is adapted to bypass the congested area or node/link. In the literature, this type of mechanism is generally known as local repair. But it is directly associated with the link breakage in the case of a network where the sensor nodes are considered as the mobile nodes. The algorithm to bypass the congested node using ripple-based search is explained in algorithm 5. In this algorithm, N_C is considered as a node that is likely to congest

in a near future and N_C^P represents a predecessor node of N_C such that, $(N_C$ and $N_C^P) \in \vartheta_r$.

In ripple-based search, the predecessor sensor node broadcasts a Local Recovery Message (LRM) to its 4-hop neighbours. This 4-hop broadcast executes with respect to the Time to Live (TTL) value. The TTL value indicates the validity time for the expiration of this LRM. The sensor nodes that receive this LRM will respond, if the receiving sensor node has the address to the sink and its ID does not match with the ID of the affected node. If this happens, then the node will send a reply to the node from where this broadcast message was generated, to maintain the ETE route. During this process, the packets are being buffered by the node that broadcast this request. Based on this mechanism, a node to be congested in the near future is bypassed, reactively. It is quite possible that the sensor nodes may not reply during this LRM or a time-out occurs to receive the LRM reply, in such a case, the following mechanism is adapted, that discovers the least congested and energy-efficient route.

Algorithm 5: Congestion alleviation using ripple-based search

Notations:

N_C^P	The predecessor node of the node that is to be congested
S_{ID}	The source node identifier
D_{ID}	The destination node identifier
N_C	Sensor node to be congested
B_{ID}	Broadcast request identifier
ϑ_r	The set of nodes on an active route, r
H_i	Hop count at sensor node i
F_C	Forward node of N_C

```

01:  $N_C^P$  broadcasts the request with  $S_{ID}, D_{ID}, B_{ID}, N_C$  till
    TTL = 4-hops away // TTL = Time To Live
02:  $i = N_C^P$ 
03: FOR each node in  $A_i$ 
04:   IF ( $i == N_C$ ) and  $TTL \neq 0$  then
05:     Discard the request
06:   ELSE IF ( $i \neq N_C$ ) and ( $i \in \vartheta_r$ ) and  $TTL \neq 0$ 
    and ( $H_i \geq H_{N_C^P}$ ) || ( $i == F_C$ ) then
07:     Add reverse route entry with  $N_C^P$  based on (15)
08:     Add forward route entry
09:     Send unicast Reply to  $N_C^P$ 
10:   ELSE IF ( $TTL \neq 0$ )
11:     Re-broadcast the request, Repeat Step : 1
12:     Add reverse route entry with  $N_C^P$ 
13:   END IF
14:   Decrement TTL
15:   IF ( $TTL = 0$ ) then
16:     Send notification to source node to re-discover the new
    least congested and energy-efficient route
17:   END IF
18: END FOR

```

4.6.2 Congestion alleviation based on the detour path

Contrary to the ripple-based search mechanism, if the affected node is near to the source, then a new route with the least congestion, consumed energy and high signal strength is discovered based on the mechanism explained in the route discovery procedure. This mechanism is also adapted, if the

node that has generated the LRM could not able to receive the response. To determine the detour path, a notification is sent to the source about the status of the node in the route network. When the source receives this message, it re-generates the RREQM to find the detour path. The procedure to discover the detour path is the same as discussed in algorithm 1, in the route discovery procedure of the CTLS, except for some modifications, such as, the source node will generate a new RREQM with a different broadcast ID along with a different sequence number. In this procedure, the congested nodes and the weak nodes in terms of low-energy will be bypassed in the route discovery procedure.

5 Performance evaluation

In this section, the performance of the proposed CTLS is evaluated as compared to CADA and NOCC [18] using Network Simulator (ns-2.35) [37]. In this study, a carrier frequency of 2.4 GHz has been utilized according to the specifications used in GS1011 [38] for MAC and physical layers. GS1011 specifications have been used because they are suitable for an ultra low-power wireless chip, which supports the IEEE 802.11 radio with a high data rate with an ultra-low power consumption. The energy model used by the sensor nodes contains the transmission (T_x), reception (R_x), idle/listening and sleep powers. To access the performance of the CTLS scheme with respect to CADA and NOCC, the sensor nodes in a WSN have been deployed in an area of 700 m \times 700 m. There is only one sink that has been placed at the centre of the network topology. In the topology, different sensing nodes (sources) are randomly selected, ensuring that the sensing nodes are not in the direct transmission range of the sink and are at least 2-hops away. All the sensor nodes deployed in the sensor field are considered to have the same transmission power. The simulation parameters used in this work are summarized in Table 2.

5.1 Simulation results

Various performance measures have been considered in the evaluation of CTLS, CADA and NOCC. These performance measures are PDR, ETE delay, average throughput, and energy consumption per data packet, and have been evaluated with respect to the packet inter-arrival time, number of sources, and the number of nodes in the network.

5.1.1 Impact of varying packet interarrival time

PDR is an important parameter to evaluate the performance of the network. It is a ratio between the packets received successfully at the receiver to the packets sent by the source. As the packet inter-arrival time increases, the PDR also increases

Table 2 Simulation parameters

Simulation parameters	Value
PHY/MAC protocol	IEEE 802.11
Max. Tx range	100 m
Max. Carrier sensing range	120 m
txPower (+9dBm)	0.495 W (26.946 dBm)
rxPower	0.462 W (26.646 dBm)
idlePower	0.0000231 W (-16.363 dBm)
sleepPower	0.000495 W (-3.0539 dBm)
Packet interval	Varies or 20 packets/s
Data packet size	80 Bytes
No. of runs	30
Simulation time	100 s

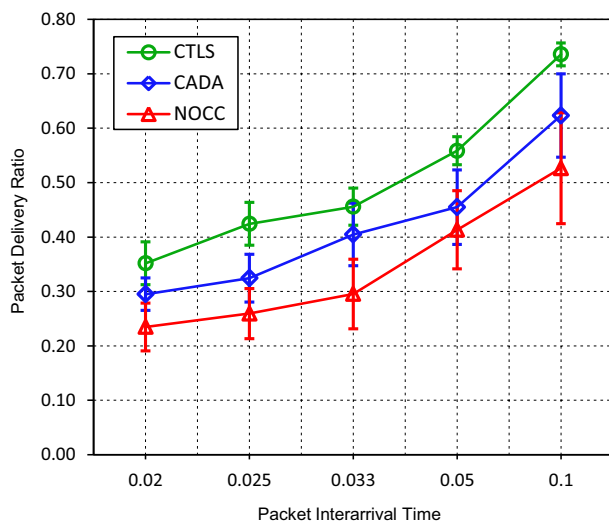


Fig. 13 Packet delivery ratio versus packet inter-arrival time with 8 different random sources for the simulation time of 100 s in 25-node network

as shown in Fig. 13. The reason for this increase is that the higher packet inter-arrival time generates lower data traffic and perhaps there are less chances of collision of data packets. It can be observed that the rate of increase in PDR in the case of CTLS is higher than CADA and NOCC. This is due to the fact that in CTLS, the forward nodes are selected based on λ , η and Ω from which λ and η are related to the node's characteristics. In contrast to this, CADA selects the node with high signal strength and residual energy without considering the buffer state of the node. That usually selects a route with some common nodes, which are used to pass several flows to the sink. This increases the packet loss that leads to a smaller rate of increase in PDR as compared to CTLS. On the other hand, NOCC does not consider the characteristics of the nodes and the links between the nodes therefore, it always tries to select the shortest route to the sink that soon become congested which increases severe packet loss. This

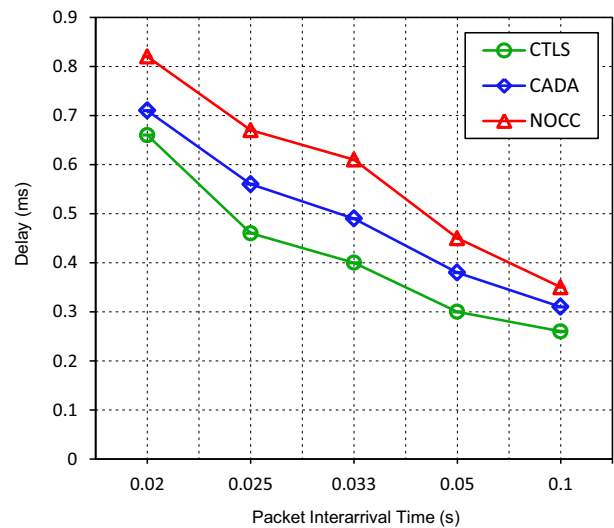


Fig. 14 ETE delay versus packet inter-arrival time with 8 different random sources for the simulation time of 100 s in 25-node network

loss actually results in a smaller rate of increase in PDR in the case of NOCC as compared to CADA and CTLS.

The ETE delay is also an important parameter to evaluate the network performance. The ETE delay usually increases with the increase in the transmission, processing, and the queuing delays. The ETE delay usually decreases either with the shortest route to the sink subject to the minimum number of traffic flows or by selecting the nodes with appropriate buffer space. Generally, the ETE delay decreases with the increase in the packet inter-arrival time because of the decrease in the packet generation rate in the network. As shown from the Fig. 14, the ETE delay for all the protocols is very high at 0.02 s of packet inter-arrival time, this is because the traffic rate is very high and therefore, the probability of congestion and packet loss is high. The rate of decrease in the ETE delay increases in the case of NOCC and CADA as compared to CTLS. This is because in CADA, the same representative nodes are selected for multiple flows and in NOCC, the shortest route to the sink is selected; however, in both cases, the traffic and the buffer states of the node are not considered while discovering the route that ultimately leads to excessive packet loss and hence increases the ETE delay. On the other hand, CTLS selects the next hop node based on the number of flows accommodated by the node.

To measure the efficiency of the protocol in wireless sensor networks, throughput plays a significant role. It measures the efficiency in terms of the data transfer with respect to time. Here, time is considered as the difference between the last packet-received-time and the first-packet-received-time. The trend of throughput with respect to the packet inter-arrival time is shown in Fig. 15. Here, CTLS shows a smaller rate of decrease in the throughput as compared to CADA. This is because; CTLS finds the shortest route with high energy

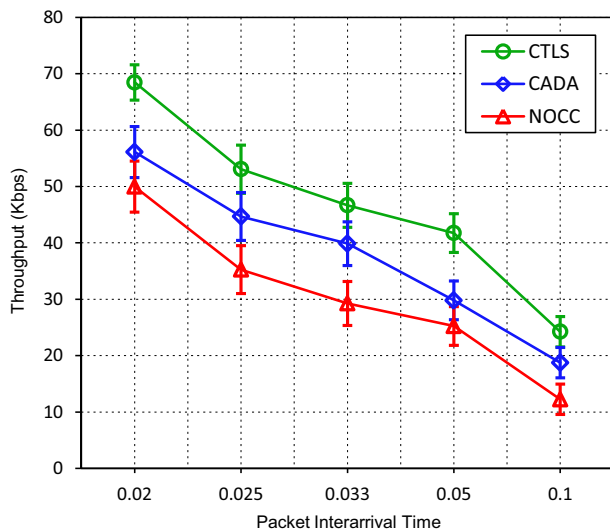


Fig. 15 Throughput versus packet inter-arrival time with simulation time 100 s and 8 different sources are running in 25-node network

and the least congestion. However, CADA avoids congestion by only considering the energy and received signal strength of the node while selecting the representative nodes. Therefore, the chances are high that the sensor nodes along the route become bottlenecks. Conversely, NOCC does not handle congestion and always follow the shortest route to the sink. Therefore, the nodes in the shortest path using NOCC and the representative node in CADA accommodate multiple traffic flows. Due to this, the buffer overflows and a high packet loss occur that ultimately increases the rate of decrease in the throughput.

As sensor nodes have very limited battery power, the efficient utilization of energy is an important consideration. Most of the energy is consumed during transmission and reception states. Efficient utilization of the sensor nodes during route discovery to forward the data packets results in effective energy consumption. The energy consumption by all protocols is shown in Fig. 16. It can be seen that the rate of decrease in the energy consumption per data packet by CTLS is smaller as compared to CADA and NOCC. This is because the composite routing metric used in CTLS facilitates the routing process with precise information about the sensor nodes and the link between these nodes. CADA has utilized various timers to update its states and the characteristics of the neighbouring nodes by periodically sending the 1-Hop messages, hence more energy is depleted, which increases energy consumption. In NOCC, due to an increase in the number of retransmissions, the energy consumption is very high. In contrast to this, CTLS, tries to select the node with high residual energy and less number of flows that reduces the chances of congestion in the network. Additionally, there are no such timers as used in CADA, to update the state of the nodes periodically. Therefore, this reduces the number of

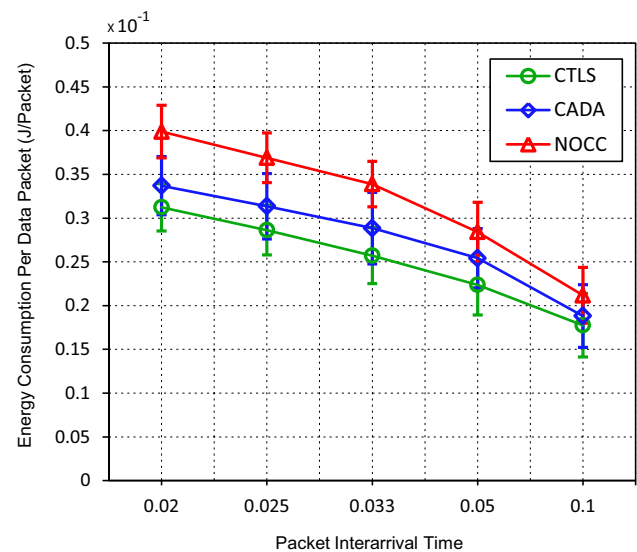


Fig. 16 Energy consumption per data packet vs. packet inter-arrival time with simulation time 100 s and 8 sources are running in 25-node network

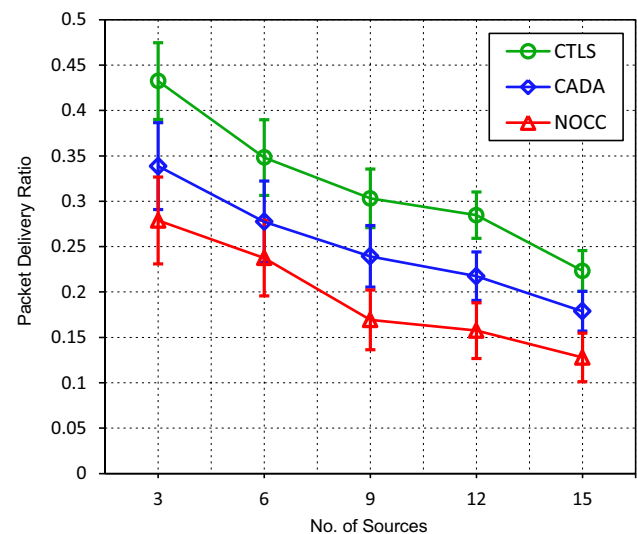


Fig. 17 Packet delivery ratio versus variable number of sources with simulation time 100 s and packet inter-arrival time 0.02 s (50 pkts/s) in 25-node sensor network

control packets which decreases the rate of decrease in the energy consumption as compared to CADA and NOCC.

5.1.2 Impact of varying number of sources

Figure 17 shows the PDR behaviour of all three protocols with respect to the number of sources. Initially, all three protocols show high value of PDR because of only three sources. However, with the increase in the number of sources the chances of congestion become high. The PDR decreases with the increase in congestion. The rate of decrease in the

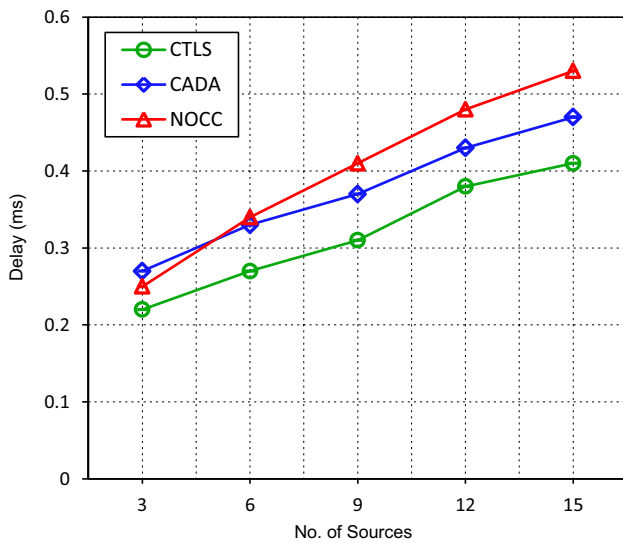


Fig. 18 ETE delay versus variable number of sources with simulation time 100 s and packet inter-arrival time as 0.02 s (50 pkts/s) over 25-node network

PDR is smaller in the case of CTLS as compared to the CADA and NOCC; this is because CTLS avoids congestion by selecting the least congested node and alleviates congestion by using the ripple-based search, if it occurs inevitably. However, CADA avoids congestion by selecting only one representative node in a broadcast range with better energy; here, the buffer occupancy or the characteristics of the node that directly impact in the avoidance of congestion has not been considered. Conversely, in the case of NOCC, there is no such mechanism to avoid and mitigate congestion. All the nodes in NOCC try to use the same shortest path to the sink. This usually drops the packets, which are beyond the maximum capacity of the buffer of a sensor node, therefore, the rate of decrease in the PDR is high. These behaviours result in reducing the PDR in CADA and NOCC as compared to CTLS.

The lower ETE delay experienced by the packets reveals better performance. It is calculated as the time experienced by the packets to reach the sink. It can be observed from Fig. 18, that the average ETE delay experienced by all three protocols is high with the increase in the number of sources. Here, CTLS shows a smaller rate of increase in the ETE delay because of the fact that the shortest route to the sink is selected with the least cost in terms of consumed energy and participation level. Due to the number of sources, congestion occurs that is detected in a timely manner and is mitigated based on the ripple-based search. Therefore, CTLS shows a smaller rate of increase in the ETE delay as compared to the other two protocols. On the other hand, CADA depicts a higher rate of increase in the ETE delay as compared to CTLS, with the increase in the number of sources. This is because of the fact that the route is selected by considering

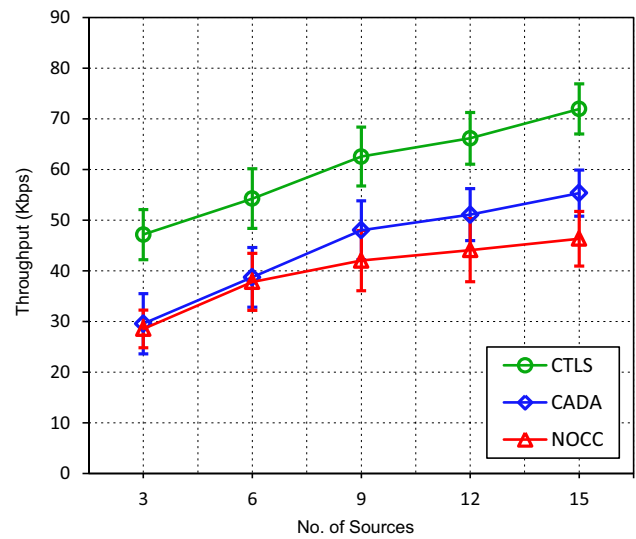


Fig. 19 Average throughput versus variable number of sources with simulation time 100 s and packet inter-arrival time 0.02 s (50 pkts/s) in 25-node network

the high remaining energy and high signal strength. Both these metrics do not directly relate to the congested state of the sensor node. However, in CADA, the sensor node with the highest signal strength reveals a longer path and that leads to an increase in the ETE delay. In the case of NOCC, the delay is very high; this is because the data path contains common nodes for multiple flows, which increases the chances of bottleneck that leads to fill up the queues early and hence the average ETE delay increases.

Figure 19 shows the throughput behaviour of all three protocols with respect to the number of sources. With the increase in the number of sources, the chances of congestion become high. The rate of an increase in the throughput is higher in the case of CTLS as compared to CADA and NOCC; this is because CTLS distributes the traffic over the network by considering a composite metric to select the forward node that decreases the chances of bottlenecks. Additionally, with the least control overhead, the occupation of wireless channel for the data packets increases that increases the throughput. On the other hand, in CADA, due to the periodic messages the control overhead increases. Furthermore, CADA alleviates congestion by sending a message to the predecessor node that is two hops away to discover the new detour path to the sink. However, the newly discovered route is not necessarily a congestion-aware route with respect to the data traffic. If the detour path cannot be found within a particular time, then the affected sensor node sends a message to the source to decrease the traffic rate. These behaviours result in reducing the rate of increase in the throughput in CADA as compared to CTLS.

The energy consumption per data packet with respect to the number of sources is shown in Fig. 20. It can be seen

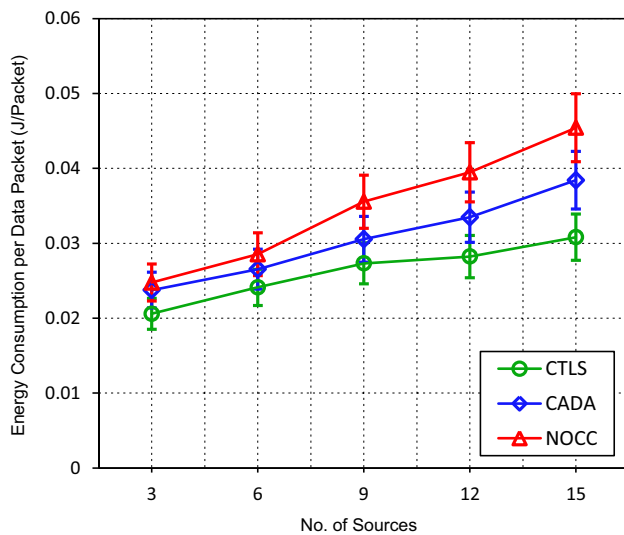


Fig. 20 Energy consumption per data packet vs. no. of sources with simulation time 100 s, packet inter-arrival time 0.02 s (50 pkts/s) in 25-node network

that the energy consumption increases with the increase in the number of sources. CADA and NOCC show higher rate of energy consumption as compared to CTLS. In CADA, the representative nodes are selected by broadcasting a periodic 1-Hop message. This actually selects a node within a broadcast region and that node usually accommodates all the flows passing through that region. Due to this, congestion occurs and the packets are being dropped at the intermediate nodes and hence, the energy consumption increases. In contrast to this, CTLS selects those nodes that contain less number of traffic flows; due to this, the chances of congestion are reduced and the rate of increase in the energy consumption decreases as compared to CADA and NOCC. NOCC shows higher rate of increase in the energy consumption; this is because all the nodes try to use the same shortest path, so the packets beyond the maximum capacity of the buffer leading to have a higher rate of increase in the energy consumption per data packet.

5.1.3 Impact of varying number of sensor nodes

The PDR with respect to the number of nodes is shown in Fig. 21. It can be seen that the PDR increases in all three protocols with the increase in the number of nodes. This is because the chances of collision of packets and number of packet loss decrease. With the decrease in the number of sensor nodes and more number of traffic flows, the chances of interference, collision and congestion become high and the channel utilization increases. The PDR in CTLS is higher as compared to CADA and NOCC because of the minimum number of control packets and the selection procedure of the forward sensor nodes or the relay nodes. In CADA, the con-

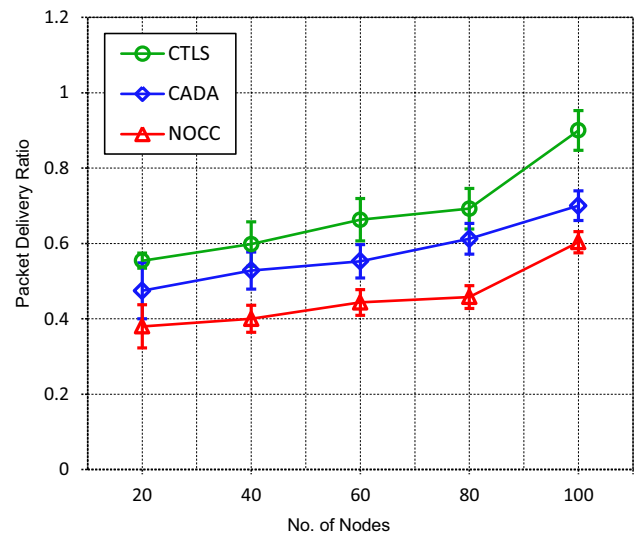


Fig. 21 Packet delivery ratio versus no. of nodes with simulation time 100 s and packet inter-arrival time 0.033 s for 7 different number of sensing nodes

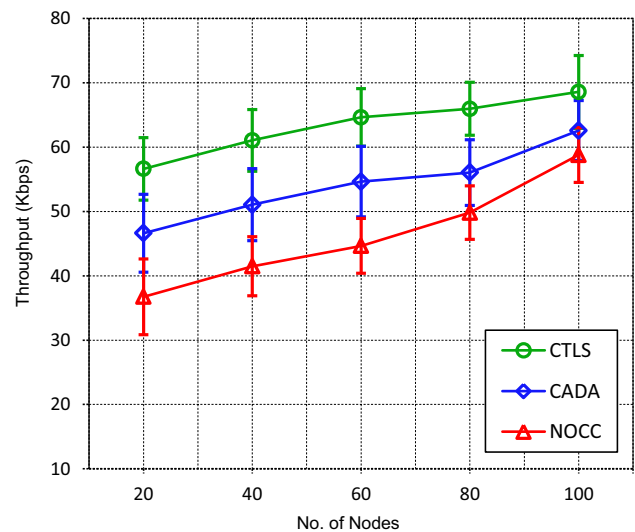


Fig. 22 Throughput versus variable number of nodes simulation time 100 s, packet inter-arrival time 0.033 s (30 packets/s) for 7 different sensing nodes

trol overhead is high because of the periodic messages. This decreases the wireless channel occupancy for the data packets in the network that leads to a smaller rate of increase in PDR in the case of CADA as compared to CTLS. In contrast to this, NOCC shows a smaller rate of increase in PDR as compared to CADA and CTLS; this is because it forwards the traffic over the shortest route and there is no mechanism to avoid and detect congestion. The packets in NOCC always try to use the same shortest path to the sink that increases the packet loss and decreases the rate of increase in PDR.

Figure 22 shows the throughput behaviour of all three protocols with respect to the number of nodes in the network.

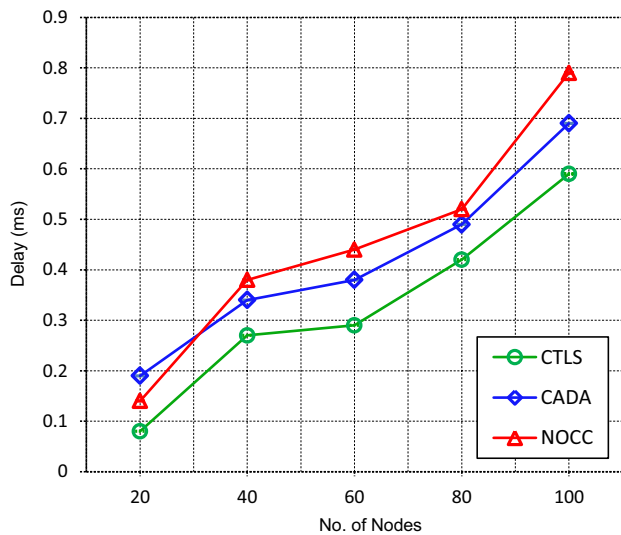


Fig. 23 ETE delay versus variable number of nodes with simulation time 100 s and packet inter-arrival time as 0.033 s for 7 different number of sensing nodes

The throughput increases with the increase in the number of nodes because of more even distribution of data traffic in the network. Throughput in CTLS is high as compared to CADA and NOCC; this is because it timely detects congestion and tries to alleviate it locally by using the ripple-based search. Based on this, the congested nodes are bypassed that increases the sensor nodes to forward the traffic to the next hop node without dropping it. However, like CTLS, CADA also alleviates congestion by sending a message to the predecessor node to discover the new detour path. However, if it cannot find the path within a particular time, then the predecessor node sends a message to the source to decrease the traffic rate. This behaviour results in reducing the rate of increase in the throughput in CADA. Unlike CADA, NOCC does not have any local repair mechanism so it always tries to use the same route for the whole communication, which leads to increase in the packet loss due to congestion that decreases the rate of increase in the throughput.

The ETE delay with respect to the number of sensor nodes in the network is shown in Fig. 23. Initially CTLS, experiences very less delay as compared to CADA and NOCC. This is because of the shortest route with the least cost in terms of consumed energy and participation level and signal strength. Additionally, there are no periodic messages in CTLS and the number of relay nodes are also minimum in the path, however, in CADA and NOCC the periodic messages decreases the chance of channel occupancy for the data packets that lead the data packets to exhibit more delay. Another reason of having a high ETE delay in CADA as compared CTLS is because of the number of relay nodes in a path. Due to this, the control traffic generated by CADA also increases. This control traffic reduces the flow of data packets that lead

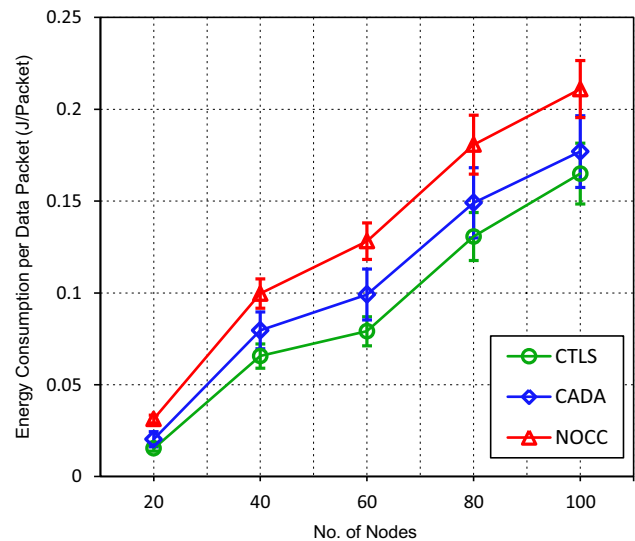


Fig. 24 Energy consumption versus variable number of nodes with simulation time 100 s and packet inter-arrival time as 0.033 s for 7 different traffic flows

to high ETE delay. In the case of NOCC, the delay is very high as compared to CTLS and CADA; this is because the data path contains common sensor nodes (bottleneck nodes) for multiple flows, which increases the packet loss and the queuing delay so the average ETE delay increases.

Figure 24 shows the behaviour of energy consumption per data packet of all three protocols with respect to the number of nodes. It can be seen that the energy consumption increases with the increase in the number of nodes. Here, CTLS shows a smaller rate of increase in the energy consumption as compared to CADA and NOCC. This is because CTLS tries to balance the traffic load over the network in order to make the traffic flow smooth. However, the other two protocols do not consider the state of the nodes with respect to the traffic while selecting the nodes. This behaviour leads the network in the state of congestion. Due to this, the packet loss increases that increases the rate of increase in the energy consumption per data packet in the case of CADA and NOCC.

Network lifetime is an important parameter to envision the durability of the network. Figure 25 shows the impact of the number of nodes on the network lifetime. The network lifetime decreases with the increase in the number of nodes. This is because of the rise in the overall energy consumption because of the idle/listening and reception states of the nodes. CTLS selects the nodes with the least energy consumption and participation level. Due to this, there is an even distribution of power consumption among the nodes. Whereas, CADA and NOCC periodically generate the messages, which increases the energy consumption. Additionally, NOCC always selects the shortest path that results in packet loss due to congestion and hence the network lifetime decreases in CADA and NOCC as compared to CTLS.

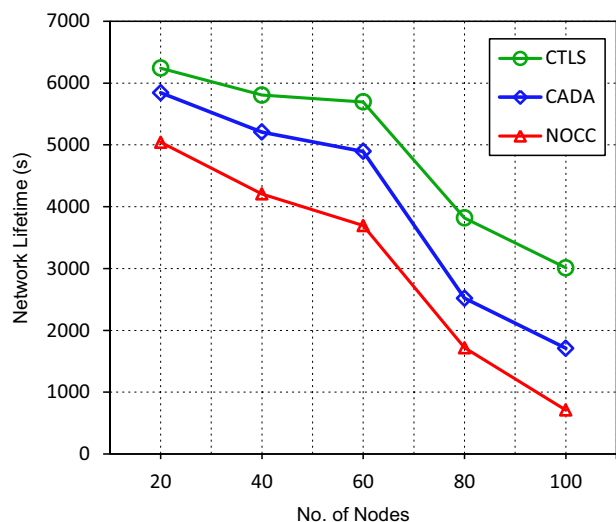


Fig. 25 Network lifetime versus variable number of nodes with simulation time 100 s and packet inter-arrival 0.5 s for 3 random traffic sources in a WSN

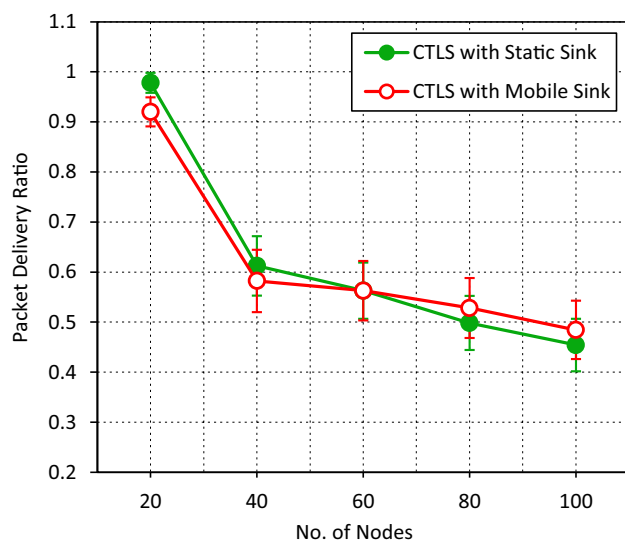


Fig. 26 PDR versus variable number of nodes with static and mobile sink over a simulation time of 100 s and inter-arrival time as 0.033 s for 7 traffic flows

5.1.4 Impact of static and mobile sink in CTLS

The impact of the number of nodes on PDR in a proposed CTLS with static sink and mobile sink is shown in Fig. 26. In this scenario, the sink broadcasts the beacon message after every 1 s to show its existence to its neighbourhood and it moves with a speed of 1 ft/s. CTLS with static sink shows better performance as compared to the mobile sink when the number of nodes in the network is less. This is because of no topological change in the network, whereas, in the case of CTLS with mobile sink, the frequent change in the topology increases the packet loss due to the mobility that leads to decrease in PDR. CTLS with mobile sink behaves well with

the increase in the number of nodes. This is because of the fact that the mobile sink is used to disperse the data traffic flows that decreases the effect of congestion, however, at the cost of control overhead.

6 Conclusion

Conventional WSN algorithms try to avoid congestion by selecting the next hop node with maximum buffer capacity, detect it by monitoring the buffer occupancy and alleviate it by adjusting the source traffic rate. The proposed CTLS protocol avoids congestion proactively by modifying the traditional route discovery mechanism in order to select the best node during the forward route formation. It detects congestion in a timely manner by monitoring either the remaining space of the buffer, the interval between the consecutive packets and the link utilization based on the number of times a node goes into the Backoff stage of CSMA/CA. The CTLS either bypasses the congested node/link through a local repair technique or deviates the traffic to the detour path in order to alleviate congestion. The performance evaluation performed in ns-2 has confirmed the usability of CTLS over existing schemes. The CTLS achieved 37 % improvement in the ETE throughput as compared to CADA and 45 % as compared to NOCC. Similarly, the ETE delay in CTLS has been decreased by 37 % as compared to CADA and 50 % as compared to NOCC. Moreover, the PDR in the case of CTLS has been increased by 24 % as compared to CADA and 45 % as compared to NOCC. Likewise, the energy consumption per data packet in CTLS has been decreased by 25 % as compared to CADA and 47 % as compared to NOCC. For future directions, efforts can be made to use different routing metrics based on the application requirements and to extend the analysis for delay sensitive applications to prioritize the packets in the case of multi-sink network. It can also be extended to mitigate congestion by using single or multiple mobile sinks.

Acknowledgments The authors would like to acknowledge the Ministry of Higher Education (MOHE) Malaysia, under the Fundamental Research Grant Scheme (FRGS/1/2012/TK02/UTP/03/02) fund and to the CISIR Laboratory, Universiti Teknologi PETRONAS for providing the research facility.

References

1. Agrawal, D., & Zeng, Q.-A. (2010). *Introduction to wireless and mobile systems*. Cengage Learning.
2. Han, H., Zhang, B., & Li, Q. (2013). The improved routing protocol of wireless sensor network based on aodv. In *Control conference (CCC), 2013 32nd Chinese* (pp. 7410–7415). IEEE.
3. Heo, J., Hong, J., & Cho, Y. (2009). Earq: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics*, 5(1), 3–11.

4. Kumar, M., Gupta, I., Tiwari, S., & Tripathi, R. (2013). A comparative study of reactive routing protocols for industrial wireless sensor networks. In *Quality, reliability, security and robustness in heterogeneous networks* (pp. 248–260). Berlin: Springer.
5. Zhang, Z., & Zhou, H. (2009). Empirical examination of mobile ad hoc routing protocols on wireless sensor networks. *International Journal of Computer Networks and Communications*, 1(1), 75–87.
6. Booranawong, A., Teerapabkajorndet, W., & Limsakul, C. (2013). Energy consumption and control response evaluations of aodv routing in wsns for building-temperature control. *Sensors*, 13(7), 8303–8330.
7. Kulik, J., Heinzelman, W., & Balakrishnan, H. (2002). Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, 8(2/3), 169–185.
8. Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 56–67). ACM.
9. Schurgers, C., & Srivastava, M.B. (2001). Energy efficient routing in wireless sensor networks. In *Military communications conference, 2001. MILCOM 2001. Communications for network-centric operations: Creating the information force. IEEE* (vol. 1, pp. 357–361). IEEE.
10. Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences* (pp. 10). IEEE.
11. Johnson, D. B., Maltz, D. A., & Broch, J. (2001). *Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks*. Pittsburgh, PA: Computer Science Department Carnegie Mellon University.
12. Perkins, C.E., & Royer, E.M. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings. WMCSA'99. Second IEEE workshop on mobile computing systems and applications* (pp. 90–100). IEEE.
13. Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6), 6–28.
14. Ee, C.T., & Bajcsy, R. (2004). Congestion control and fairness for many-to-one routing in sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 148–161). ACM.
15. AlAmri, H., Abolhasan, M., Franklin, D. R., & Lipman, J. (2013). Optimised relay selection for route discovery in reactive routing. *Ad Hoc Networks*, 11(1), 70–88.
16. Camara, D., & Loureiro, A.A.F. (2000). A gps/ant-like routing algorithm for ad hoc networks. In *IEEE wireless communications and networking conference WCNC, 2000* (vol. 3, pp. 1232–1236). IEEE.
17. Azlan, A., Lagrange, X., & Ros, D. (2009). A cross-layer medium access control and routing protocol for wireless sensor networks. *Proceedings of 10emes Journées Doctorales en Informatique et Réseaux (JDIR 2009)*.
18. Fang, W., Chen, J., Shu, L., Chu, T., & Qian, D. (2010). Congestion avoidance, detection and alleviation in wireless sensor networks. *Journal of Zhejiang University SCIENCE C*, 11(1), 63–73.
19. Wan, C.-Y., Eisenman, S. B., & Campbell, A. T. (2003). Coda: Congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on embedded networked sensor systems* (pp. 266–279). ACM.
20. Tao, L. Q., & Yu, F. Q. (2010). Ecoda: Enhanced congestion detection and avoidance for multiple class of traffic in sensor networks. *IEEE Transactions on Consumer Electronics*, 56(3), 1387–1394.
21. Kumar, R., Crepaldi, R., Rowaihy, H., Harris, A. F., Cao, Guohong, Zorzi, Michele, et al. (2008). Mitigating performance degradation in congested sensor networks. *IEEE Transactions on Mobile Computing*, 7(6), 682–697.
22. Ahmad, M. Z., & Turgut, D.O. (2008). Congestion avoidance and fairness in wireless sensor networks. In *Global telecommunications conference, 2008. IEEE GLOBECOM 2008* (pp. 1–6). IEEE.
23. Fang, W.-W., Qian, D. P., & Liu, Y. (2008). Transmission control protocols for wireless sensor networks [j]. *Journal of Software*, 6, 020.
24. Akan, O. B., & Akyildiz, I. F. (2005). Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 13(5), 1003–1016.
25. Lee, J.-H., & Jung, I.-B. (2010). Adaptive-compression based congestion control technique for wireless sensor networks. *Sensors*, 10(4), 2919–2945.
26. Kang, J., Zhang, Y., & Nath, B. (2007). Tara: Topology-aware resource adaptation to alleviate congestion in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(7), 919–931.
27. Puccinelli, D., & Haenggi, M. (2008). Arbutus: Network-layer load balancing for wireless sensor networks. In *Wireless communications and networking conference, WCNC 2008, IEEE* (pp. 2063–2068). IEEE.
28. Ren, F., He, T., Das, S., & Lin, C. (2011). Traffic-aware dynamic routing to alleviate congestion in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(9), 1585–1599.
29. Jea, D., Somasundara, A., & Srivastava, M. (2005). Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Distributed computing in sensor systems* (pp. 244–257). Berlin: Springer.
30. Ye, F., Zhong, G., Lu, S., & Zhang, L. (2005). Gradient broadcast: A robust data delivery protocol for large scale sensor networks. *Wireless Networks*, 11(3), 285–298.
31. Costa, L. H. M. K., Fdida, Serge, & Duarte, O. C. M. B. (2000). Distance-vector qos-based routing with three metrics. In *Networking 2000 broadband communications, High Performance networking, and performance of communication networks* (pp. 847–858). Berlin: Springer.
32. Chughtai, O., Badruddin, N., & Awang, A. (2014). A congestion-aware and energy efficient traffic load balancing scheme for routing in wsns. In *TENCON 2014-2014 IEEE region 10 conference* (pp. 1–6). IEEE.
33. Chughtai, O., Badruddin, N., & Awang, A. (2014). Distributed on-demand multi-optional routing protocol in multi-hop wireless networks. In *TENCON 2014-2014 IEEE Region 10 conference* (pp. 1–6). IEEE.
34. Gupta, N., & Das, S. R. (2002). Energy-aware on-demand routing for mobile ad hoc networks. In *Distributed computing* (pp. 164–173). Berlin: Springer.
35. Gouda, M. G., & Schneider, M. (2003). Maximizable routing metrics. *IEEE/ACM Transactions on Networking (TON)*, 11(4), 663–675.
36. Zhao, L., & Fan, C. (2004). Enhancement of qos differentiation over ieee 802.11 wlan. *IEEE Communications Letters*, 8(9), 494–496.
37. Issariyakul, T., & Hossain, E. (2011). *Introduction to network simulator NS2*. Springer Science & Business Media.
38. GS1011-datasheet. (2014). Low-power wireless system-on-chip wi-fi module data sheet, [online] available: Grainspan, <http://www.datasheetarchive.com/gs1011-datasheet.html>. accessed, May 2014.



Omer Chughtai is a Research Scholar in the Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Malaysia. He received his B.Eng. degree in IT from University of Engineering and Technology Taxila, Pakistan. He later received his MS degree in Computer Engineering from COMSATS Islamabad, Pakistan. He worked as a Research Engineer in a sponsored R&D project titled as “End-to-End Mobility Management Framework (EMF)

for Multihomed Mobile Devices”. He received an IETF Fellowship to participate in 91st IETF Meeting, Honolulu, Hawaii, USA from ISOC in November 2014. He has several national and international research publications. His research interests include cross layer protocol design, routing in low power wireless ad hoc networks, traffic load balancing, congestion avoidance, detection & alleviation in multi-hop wireless ad hoc networks, and mobility management in heterogeneous networks.



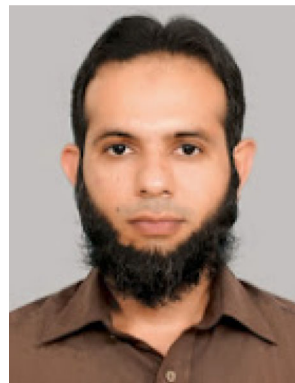
Nasreen Badruddin graduated with a first class honours B.Eng. degree in Electronic Engineering from RMIT University, Australia, in 2000. She later received an M.Sc. in Electrical & Computer Engineering from Carnegie-Mellon University, USA in 2002, and a Ph.D. in Electrical & Electronic Engineering from the University of Melbourne, Australia, in 2011. Her Masters thesis was on the potential capacity benefits of utilising fixed relay stations between base stations

and mobile users in a CDMA cellular network, while her Ph.D. thesis focused on finding optimal power allocation and scheduling schemes which would maximise the overall throughput in interference networks. Her research interests are primarily in the area of wireless communications and networks as well as biomedical engineering, particularly in neuro-signal processing and wireless body area networks (WBAN). Nasreen was a recipient of a PETRONAS overseas scholarship for her undergraduate studies and an Endeavour Postgraduate Award from the Australian government for her Ph.D. studies. She is currently an Associate Professor at Universiti Teknologi PETRONAS, and attached to the Biomedical Technology research group as well as the Centre of Intelligent Signal and Imaging Research (CISIR).



Azlan Awang received his Ph.D. degree from TÉLÉCOM Bretagne and University Rennes 1, Rennes, France in 2011. He received his B.Sc. and M.Sc. degrees in Electrical Engineering from Polytechnic Institute of NYU, Brooklyn, New York, in 1989 and 1990, respectively. He was with Motorola (M) Sdn Bhd (1991), Schlumberger Overseas S.A. (1992–1993), Alcatel Networks Systems (M) Sdn Bhd (1994–2001) and Universiti Teknologi MARA (2002–2004).

Since 2004, he has been with Universiti Teknologi PETRONAS, Malaysia in the Department of Electrical and Electronic Engineering. His research interests include cross-layer MAC/Routing protocols for Wireless Sensor Networks, Wireless Body Area Networks and Vehicular Ad Hoc Networks. He is a Chartered Engineer (CEng) and a member of IEEE, Eta Kappa Nu and Tau Beta Pi engineering honor societies.



Maaz Rehan has done MSc (Computer Science) from Quaid-I-Azam University Islamabad Pakistan in 2002 and PhD in the department of Computer Science from Universiti Teknologi PETRONAS, Malaysia in 2015. He served as Lecturer for 4 years in Qurtuba University and taught CS Students of BS(Hons). Currently he is an Assistant Professor in COMSATS Wah Cantt, Pakistan. His research interests include cross-layer designs & optimizations; IEEE 802.21 based vertical handover decision algorithms; Load Balancing and Multipath QoS Routing in Wireless Multihop Networks (especially MANETs/VANETs) using ant-colony optimization.