IET Wireless Sensor Systems

Research Article

Congestion control and energy-balanced scheme based on the hierarchy for WSNs

ISSN 2043-6386
Received on 24th July 2015
Revised 10th May 2016
Accepted on 13th August 2016
E-First on 18th October 2016
doi: 10.1049/iet-wss.2015.0097
www.ietdl.org

Wenguang Chen[1], Yugang Niu[1] ✉, Yuanyuan Zou[1]

[1]*Key Laboratory of Advanced Control and Optimization for Chemical Process of Ministry of Education, East China University of Science and Technology, Shanghai 200237, People's Republic of China*

✉ *E-mail: acniuyg@ecust.edu.cn*

Abstract: The problem of congestion control with balanced-energy is important for application of WSNs, since the limited resources and many-to-one communication model often result in congestion and unbalanced energy consumption. In this study, a hierarchy-based congestion control and energy-balanced scheme is proposed. The network model is firstly initialised into a hierarchical topology, by which these neighbour nodes of a node will be explicitly divided into three kinds, i.e., the same hierarchical nodes, the upstream nodes, and the downstream nodes. Then, in the proposed congestion avoidance method, the node will use other lower hierarchy neighbour nodes to forward data when its downstream node will be congested. After that, the congestion control mechanism will detect the congestion via the queue length, forwarding and receiving rate, and inform its upstream nodes to find other next hop to release the congestion. The balanced energy consumption strategy will balance the energy consumption of lower hierarchy nodes by using the node with the most remaining energy. Meanwhile, by using the same hierarchy nodes, the remaining energy of all the nodes in the same hierarchy is balanced. Simulation results show that the proposed algorithm can effectively deal with the network congestion and unbalanced energy consumption.

1 Introduction

In recent years, wireless sensor networks (WSNs) have attracted a great deal of attention with the development of the electronics and wireless communications technologies. A WSN is composed of hundreds or thousands of sensor nodes equipped with various components, e.g. sensing, energy supplying, data processing, and communicating units. Those nodes work together to detect the event of predetermined nature or obtain data about the environment, and transmit the sensed data to the sink or base station through wireless links [1]. The sensor nodes with restricted power are often deployed in the areas where it is difficult to replace their power. This brings many constraints for designing protocols on WSNs.

It is well known that the WSNs have shown great potential in many practical applications, such as military surveillance, target tracking and natural disaster relief and so on [2]. Generally, there are large amounts of data to be transmitted from the source to the sink, which may frequently result in congestion due to the many-to-one data flow. Congestion is an undesired phenomenon due to serious reducing the network performance and the network lifetime, and causing wastage of the limited energy. So, the problem of congestion control in WSNs has been receiving more attentions in the past decades and many congestion control protocols have been proposed for WSNs, see the survey papers [3–6]. The key feature of these protocols is to decrease the data rate of the source or forward to the data flow to other spare neighbour nodes so as to attain releasing congestion [7]. Thus, they usually use the shortest path to forward data, which may make the nodes run out of energy quicker than other nodes. Moreover, when they find the spare neighbour nodes, they just consider the hop count and the buffer occupancy of neighbour nodes.

Apparently, the energy balancing of sensor nodes is also quite important for the practical application of WSNs [8]. It is known that the many-to-one communication model can cause funnelling effect and the energy-hole problem in the sensor network [9]. It was shown in [10] that the energy-hole problem cannot be eliminated completely and the neighbour nodes of the sink always run out of their energy quicker than other nodes. In order to prolong the network lifetime, some load-balancing routing protocols were proposed in [11–14], in which these nodes communicating with the sink node directly would deplete their energy at roughly same time. It is assumed in those protocols that the nodes produce packets regularly, i.e. they do not consider the burst of data flow which often results in congestion in the network.

As mentioned above, both the congestion and unbalanced energy consumption can decrease the network lifetime. However, the existing congestion control mechanisms cannot balance the energy consumption. Meanwhile, the existing load-balancing routing protocols also do not consider the problem of network congestion when ensuring the balancing of energy consumption.

In order to cope with those problems simultaneously, this paper proposes the congestion control and energy-balanced schemes based on the hierarchy (CcEbH). At the initialisation phase of network model, a hierarchical topology is constructed from the sink node, by which every node will know its own hierarchy. The sensed data is transmitted from the high hierarchy nodes to the low or same hierarchy nodes. By checking the queue length of the next hop node, the forwarding node will decide whether or not to continue to send packets to this next hop node at the congestion avoidance phase. In order to detect the congestion accurately, the CcEbH utilise a novel congestion detection method. The upstream node will employ other unloaded parent nodes or the same hierarchy neighbouring nodes to receive packets after it knows the occurrence of congestion at the next hop node. In order to balance the energy consumption, the upstream node will take the lower-hierarchy node with the most residual energy as the receiving node. In CcEbH, the node also uses the same hierarchy neighbouring nodes to balance the energy consumption of the nodes located at the same level.

The remaining of this paper is organised as follows. In Section 2, the analysis will be made on the related works of the congestion control mechanism and the balanced energy consumption scheme. In Section 3, the hierarchical topology will be constructed. Section 4 gives the CcEbH algorithm. The simulation results will be provided in Section 5, which is followed by the conclusions in Section 6.

2 Related work

Congestion control has been a hot area of research in WSNs. Congestion will occur when the traffic load exceeds the available capacity on node level (buffer overflow) or link level (interference or contention) [3–6]. The existing congestion schemes are roughly divided into traffic control [15–18] and resource control [19–21]. The former is suitable for transient congestion situation as in [7], while the latter is more effective in cases of persistent congestion circumstances.

The main feature of the traffic control methods as in [15–18] is to send a back-pressure message to the source nodes when the occurrence of congestion is detected. Then, the source nodes will decrease the rate so as to release congestion. However, the traffic control methods are not suitable for real-time applications, such as patient monitoring, security detection and so on. Therefore, there are many efforts to focus on the resource control methods as in [19–21]. The resource control method will not decrease the data rate after the happening of network congestion. Instead, it will make use of the spare nodes to forward the redundant data so as to reduce the load of the congested nodes. Hence, for the resource control method, it is important to choose new next hop or construct alternate path. Both HTAP [19] and DAlPaS [21] will choose new next hop for any node when congestion occurs at its original next hop.

In order to find all possible routes from the sources to the sink when an event occurs, the HTAP algorithm in [19] builds a source-based hierarchical tree after the end of topology control scheme. When congestion is about to occur at a specific node, its upstream nodes will be informed to stop transmitting packets. Then, those upstream nodes will choose the available node with the same level (in comparison with the congestion node) and the least buffer occupancy as their next hop. After that, the upstream node will forward the excess packets through new next hop. Through the above algorithm, the load at the congestion node is reduced and the congestion is released.

Similar to the HTAP, the DAlPaS algorithm in [21] bases on a hierarchical tree which begins from the sink. When congestion occurs at an overloading node, the DAlPaS algorithm adopts a ‘flag decision’ algorithm which will take into account several factors, e.g. availability (flag), the number of hops to the sink, the remaining power, remaining buffer occupancy and the node one-dimensional (ID), to choose the most appropriate path to transmit packets. By employing this algorithm, the packets are forced to change their path in order to not congest the receiving node. However, both of the above algorithms not consider the problem of energy balancing consumption, when they choose available path for congestion release.

As well known, the balanced energy consumption is an important factor in the application of WSNs since it can influence the network lifetime. The energy consumption of sensor nodes involves the power for sensing, processing, receiving and transmitting data [9]. The receiving and transmitting data will deplete most of node energy. Therefore, a number of load-balancing routing algorithms as in [11–14] have been proposed to balance the energy consumption.

Kleerekoper and Filer in [11] proposed a distributed construction method based on load balanced routing trees, DECOR, for many to one sensor networks. It places limits on the number of children nodes and attempts to ensure that all the nodes at the same level have the same number of children. This can obtain a global load balanced routing trees. Gagarin *et al.* [13] proposed a modified Kruskal's minimum spanning tree search algorithm. By hierarchical clusters, the distributed search can balance the energy consumption routing.

However, it should be pointed out that some of aforementioned works can release the congestion, while others can balance the energy consumption. However, there are few works considering both the congestion control and balanced energy consumptions simultaneously. This motivates this present research. In the following, we firstly propose a construction method of a hierarchical topology as the initialisation phase of the proposed algorithm. Then, we shall present the algorithm of CcEbH.

3 Hierarchical topology

In this section, the hierarchical topology will be constructed. Suppose that the network model has one sink node and a lot of homogenous sensor nodes deployed uniformly over the target filed. The sink node has infinite energy and other nodes have the same limited initial energy without renewable energy budget. Besides, each node has a particular ID and all nodes have same communication distance.

3.1 Construction stages of hierarchical topology

The hierarchical topology formation can be divided into two stages: topology discovery and hierarchy updating. The former will form an initial hierarchical topology after the nodes deployment. The latter will update the hierarchy value of nodes so as to have less hop and delay to the sink. The above topology construction procedure is only made once at the initialisation of network model. The node will also update its hierarchy and neighbour table when its neighbour node will run out of their energy or its neighbour node has moved out its radio range

Topology discovery stage: The phase starts from the sink node. The sink node broadcasts a *hierarchy_discovery* message with its hierarchy value 0 and ID. The hierarchy of node represents the hop count from itself to the sink node. The node receiving the *hierarchy_discovery* message will consider the sink node as its neighbour node and set its hierarchy as 1 which is got by increasing the hierarchy value in the *hierarchy_discovery* message by 1. After that, this node also broadcasts a *hierarchy_discovery* message with its hierarchy 1 and ID. The above procedure will be made until each node has its own hierarchy.

Hierarchy updating stage: This stage will be carried out when any node receives a *hierarchy_discovery* message from neighbour nodes again. The node records the node ID in the *hierarchy_discovery* message into its neighbour table firstly. Then, it will compare its hierarchy with the hierarchy value in the *hierarchy_discovery* message. The node will change its hierarchy if its hierarchy value is greater than the value in the *hierarchy_discovery* by 2 or more. The node updates its hierarchy to the value which is higher than the hierarchy value contained in the *hierarchy_discovery* message by 1. Then, the node will broadcast a new *hierarchy_discovery* with its hierarchy and the node ID. In other circumstances, the nodes drop the *hierarchy_discovery* message and just record the node ID into its neighbour table.

In the following, we take the network model in Figs. 1*a–c* as an example to illustrate the above two stages.

In Fig. 1*a*, the double arrowed lines represent that the nodes at the both sides of the line can communicate with each other after the completeness of node deployment. For example, node 5 can communicate with nodes 2, 4, 6, 8, 9 and 12.

At the topology discovery stage in Fig. 1*b*, nodes 1, 2 and 3 receive *hierarchy_discovery* message at first from the sink, so their hierarchy value is 1. After setting their hierarchy, nodes 1, 2 and 3 also broadcast a *hierarchy_discovery* message with their hierarchy 1. Then, nodes 5 and 6 will receive *hierarchy_discovery* message from 2 firstly, meanwhile nodes 4 and 7 also receive *hierarchy_discovery* message from nodes 1 and 3, respectively. So their hierarchy is 2. This procedure is iterated until the fringe nodes get their hierarchy, i.e. node 16 has its hierarchy 5, and node 22 has its hierarchy 7 and so on.

At the hierarchy update stage in Fig. 1*c*, when node 12 receives a new *hierarchy_discovery* message from node 5, it will compare its hierarchy 4 with the hierarchy value 2 in the *hierarchy_discovery* message. After that, node 12 will update its hierarchy to 3 and send a new hierarchy_discovery message with its hierarchy 3 and ID to its neighbour nodes. This procedure will be executed at the nodes which receive at a new *hierarchy_discovery* message with a lower hierarchy value compared with the first *hierarchy_discovery* message which the node has received.

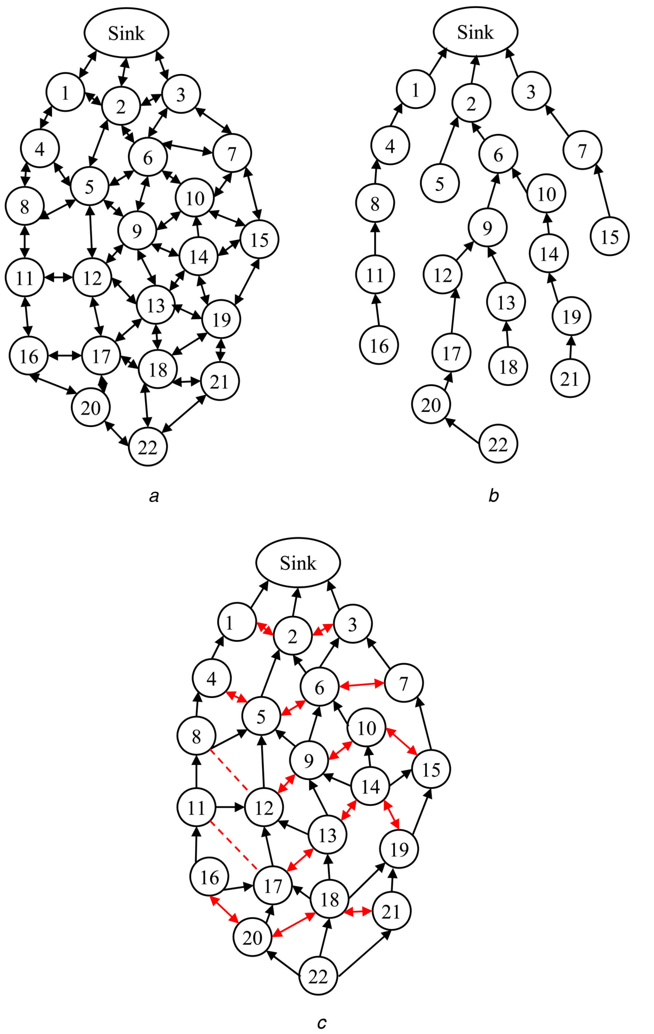

Fig. 1 *Topology discovery stage and hierarchy updating stage*
(a) Initial network connectivity, ***(b)*** Network connectivity after the topology discovery stage, ***(c)*** Network connectivity after the hierarchy update phase

In order to obtain the information of neighbour nodes, such as the queue length and the remaining energy, each node will send a *hello* message to its neighbouring nodes at specified intervals. The *hello* message is presented in Table 1. Every node retains a more detail neighbouring table with the help of *hello* message.

As an example, Table 2 shows the information of the neighbour table of node 9. The queue length is used to detect the congestion. With the help of remaining energy, the network may balance the energy consumption of nodes at the same hierarchy.

3.2 Hierarchy updating for low power and mobile nodes

The powerless nodes and the mobile nodes may bring about the change of topology structure. In this work, we propose a method to deal with the powerless nodes and mobile nodes. When a node does not receive *hello* message from its neighbour node for a period of time, it will delete this neighbour node from its neighbour table. Similar deleting procedure will be also made when the residual energy of its neighbour node is less than 20% of total energy.

The node checks its neighbour table after deleting a neighbour node with low power or moving out of its ratio range. If it has not lower-hierarchy nodes, it will increase its hierarchy by 1. Then, it will broadcast a *hierarchy_discovery* message with its new hierarchy value and ID. The nodes that receive the *hierarchy_discovery* message will update its neighbour table and execute the above process. When the node still has lower-hierarchy nodes, it will do nothing.

In order to join the hierarchical topology again, the mobile node will renew its neighbour table when it receives *hello* messages from its new neighbour nodes. It sets its hierarchy to the value which is greater than the minimum hierarchy value in its neighbour table by 1. Meanwhile, it also broadcasts a *hierarchy_discovery* message with its new hierarchy value.

Table 1 The *hello* message

Node ID	Remaining energy	Hierarchy value	Queue length

Table 2 The neighbour table of node 9

Node ID	Remaining energy, J	Hierarchy value	Queue length
5	0.62	3	40%
6	0.63	3	0
10	0.76	4	20%
12	0.79	4	0
13	0.84	5	0
14	0.82	5	15%

4 Congestion control and energy-balanced scheme

In this section, based on the constructed hierarchical topology in Section 3, we present the congestion control and energy-balanced algorithm (CcEbH), which includes three parts: congestion avoidance method, congestion control mechanism, and balancing energy consumption strategy. In the following, the whole procedure of CcEbH algorithm will be discussed in detail

4.1 Congestion avoidance method

In a particular area, when many nodes attempt to transmit data to one node simultaneously, the congestion may happen at this node. Thus, in order to reduce this kind of congestion at the downstream nodes, the following congestion avoidance method is proposed in CcEbH for the upstream nodes.

Before an upstream node sends data, it will check the buffer occupancy of the next hop node. When this buffer occupancy is greater than the queue length of the upstream node by 20% of total buffer size, the upstream node will choose other suitable downstream node to receive data. By means of the neighbour table, the node with the most remaining energy in the lower-hierarchy neighbour nodes is prior. For the case that this upstream node has only one lower-hierarchy neighbour node, the congestion avoidance method will not work.

For example, node 5 in Fig. 1*c* has three high hierarchy neighbour nodes. So, it can receive data packets from nodes 8, 9 and 12. Suppose that every node has a buffer size of 100 packets. If node 5 receives data packets from nodes 12 and 9 simultaneously with high data rate, its queue length will increase quickly. When node 5 has 46 packets in its buffer and nodes 12 and 9 have 22 and 23 packets, respectively, the proposed congestion avoidance method will be adopted. Thus, node 9 will choose node 6 from its low hierarchy neighbour nodes as its next hop node. However, node 12 will had to continue transmitting data to node 5 because it has only node 5 in its lower-hierarchy nodes. By this method, the network congestion may be avoided to some extent.

4.2 Congestion control mechanism

Although the above congestion avoidance method can decrease the probability of congestion to some content, it cannot eliminate network congestion completely. Hence, in CcEbH, the following congestion control mechanism is proposed, which includes congestion detect and congestion release. This mechanism is a resource control method and can release persistent network congestion effectively.

Congestion detect phase: When the buffer occupancy of a particular node reaches congestion detect threshold, e.g. 50% of the buffer size, the following mechanism will be adopted to detect the network congestion accurately:

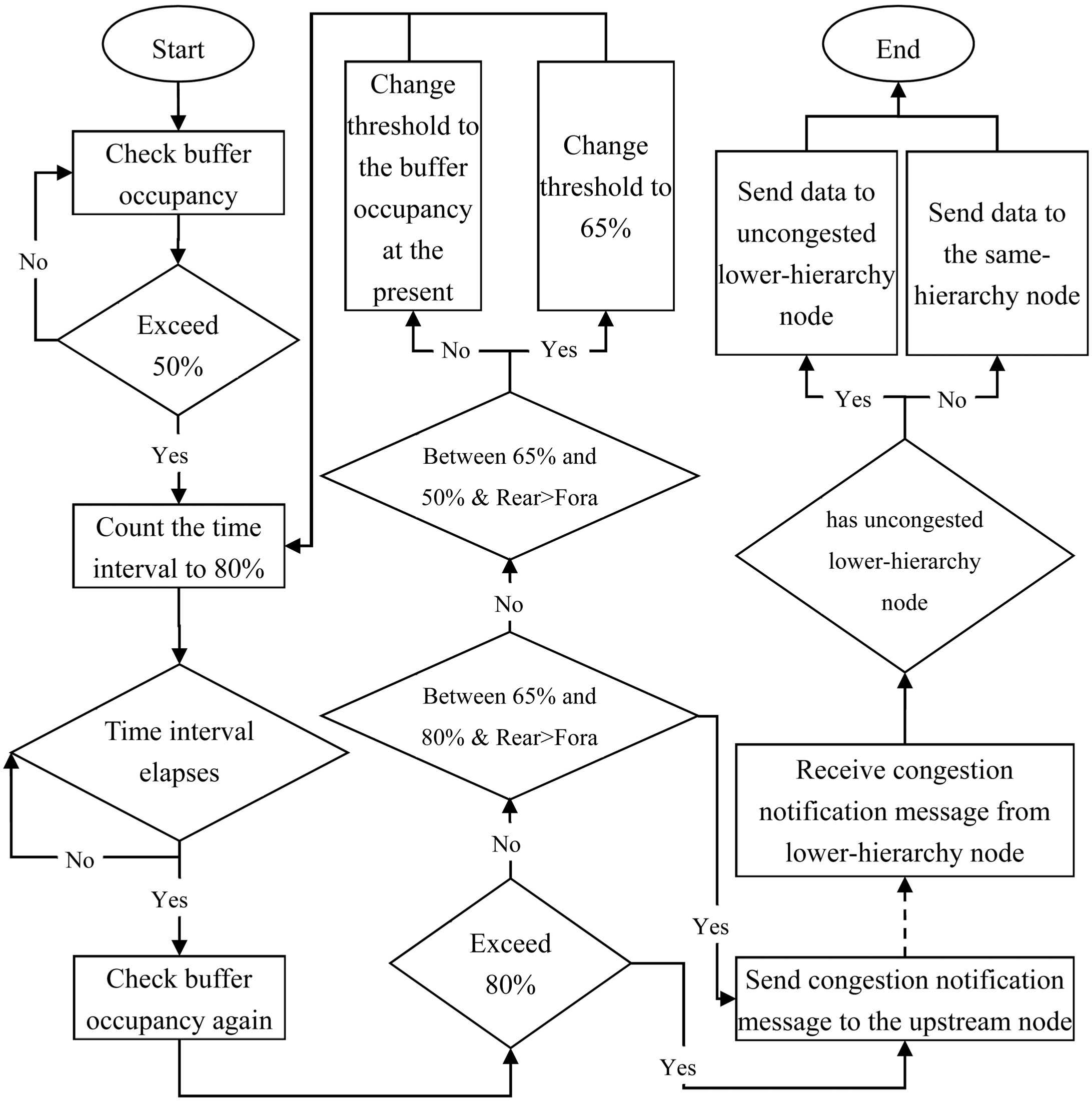

Fig. 2 *Flowchart of congestion control mechanism*

Step 1: This node counts the interval time that the buffer occupancy reaches the 80% of its buffer size as follows:

$$Tiit = \frac{(80\% - 50\%) * Qule}{Rera - Fora} \quad (1)$$

where *Rera* donates the receiving rate and *Fora* donates the forwarding rate, *Qule* is the buffer size of node, and *Tiit* is the time interval that the queue length increases from 50 to 80%.
Step 2: After this time interval elapses, we will judge the happening of the congestion according to the following conditions:

- The queue length of this node exceeds 80% of the total buffer.
- The queue length is between 65 and 80%, meanwhile the receiving rate is greater than the forwarding rate.

For the case that the queue length is less than 65% and the receiving rate is greater than the forwarding rate, the node will change the threshold from 50 to 65% and re-executes step 1.

When the queue length is between 50 and 80% and the forwarding rate is greater than the receiving rate, the congestion detect threshold will be set to the queue length of the node at present. For example, if the queue length is 68% after the time interval (*Tiit*) and the forwarding rate is greater than the receiving rate, i.e. the queue length is decreasing, and the congestion detect threshold is set to 68%. After the congestion is released, the congestion detect threshold is set to 50% again.

Fig. 2 shows the operation process of congestion detect phase and congestion release phase.

In the congestion detect phase, the CcEbH begins to detect the network congestion, when the queue length reach 50% of the total buffer, and uses the 80% of the total buffer as the index of congestion judge. By this means, the CcEbH can avoid to employ the congestion control frequently. Meanwhile, It can prevent the occurrence of packet loss before executing of congestion control.

Congestion release phase: After detecting the happening of congestion, the congested node will send a congestion notification message to its upstream nodes. This message includes the time interval of congestion removing (*Tiiocr*) which is given as:

$$Tiiocr = \frac{Buoc}{Frea} \quad (2)$$

where the *Buoc* indicates the buffer occupancy of the congested node and the *Frea* represents the forwarding rate opf the congested node. After the time interval *Tiiocr*, the congested node checks the queue length again. If it is still greater than 50%, the congested node sends the congestion notification message again.

When the upstream nodes receive the above congestion notification message, it will stop using this node to forward data to the sink before the interval time *Tiiocr* elapses. Instead, it will find other available lower-hierarchy nodes from its neighbour nodes. If this upstream node finds multiple available lower-hierarchy neighbour nodes, it will transmit packets through one of them. However, if this upstream node does not find available lower-hierarchy neighbour node, it will use the same-hierarchy neighbour nodes to

Table 3 Simulation parameters

Parameters	Values	Parameters	Values
traffic type	CBR	sleep power	0.6 mW
transmission range	250 m	idle power	0.01 W
data packet size	512 bytes	number of sinks	1
initial battery power	100 J	number of sources	10, 20
transmission power	2.5 W	bandwidth	2 M/s
receive power	0.02 W	simulation time	300 s

receive packets. To avoid routing loops, the upstream node will not choose the same-hierarchy node which also sends data to it unless this same-hierarchy node stops transmitting.

4.3 Balanced energy consumption strategy

It is well known that the unbalanced energy consumption is an inherent problem for WSNs with multiple hop routing and many-to-one traffic flow. Moreover, the sensor nodes closest to the sink tend to deplete their energy faster than other sensor nodes, which lead to the energy-hole problem and decrease the network lifetime. In order to maximize the network lifetime, the nodes within the radio range of sink node should deplete their energy roughly at the same time. In CcEbH, we use the neighbour nodes in the same hierarchy or lower hierarchy to balance the energy consumption.

The balanced energy consumption strategy not only concentrates on the balancing energy dissipation of inner-most hierarchy nodes, but also contributes to balance the energy consumption of intra-hierarchy node. The inner-most hierarchy nodes are the type of nodes that the hierarchy value is 1, while the intra-hierarchy nodes is the nodes located at the same hierarchy. The balanced energy consumption strategy is divided into two parts.

Part I: This part is to ensure the even energy dissipation of the lower-hierarchy nodes of a particular upstream node. At the beginning of data transmission, the upstream node transmits data to the node with the most remaining energy in its lower-hierarchy neighbour nodes. When the upstream node is forced to find a new lower-hierarchy neighbour node to receive data by the congestion avoidance method or congestion control mechanism, it also takes the node with the most remaining energy as its next hop node.

Part II: The part is to balance the energy consumption of all the nodes located at the same level. First, we compute the average remaining energy of the lower-hierarchy nodes (ARELH which get from dividing the total energy of lower-hierarchy nodes by the number of lower-hierarchy nodes) and the average remaining energy of the same-hierarchy (ARESH which get from dividing the total energy of same-hierarchy nodes by the number of same-hierarchy nodes) nodes of the forwarding node. This forwarding node updates the values, ARELH and ARESH, when it has packets in its buffer.

The forwarding node compares the values of ARELH with ARESH at a fixed period. It selects the lower-hierarchy nodes to receive data if the lower-hierarchy nodes have enough energy compared with the same-hierarchy nodes. When the value of the ARELH is lower than the value of the ARESH by 5% of total energy, the forwarding node will send the data to the same-hierarchy nodes with the most remaining energy. In order to avoid routing loops, the upstream node will not choose a same-hierarchy node which also sends data to it as its next hop unless this same-hierarchy node stop transmit.

5 Experimental setup and performance analysis

In this section, several experimental results are given to evaluate the effectiveness of CcEbH via some performance indices. To estimate efficiency of CcEbH in releasing congestion, we introduce the packet drop rate, average end-to-end delay, and throughput. Meanwhile, the average residual energy of nodes located at the same level and the variance of the residual energy are utilised to evaluate the effective of balanced energy consumption strategy.

5.1 Simulation environment

We present simulation environment using network simulator-2 version 2.35. We simulate the network model which consisted of 49 nodes deployed in the 800 m* 800 m uniformly and the network model that consisted of 100 nodes deployed in the 1400 m * 1400 m. The nodes are equipped with omnidirectional antennas and a buffer of 50 data packets. The 802.11 MAC protocol is used to share the wireless channel. The other simulation parameters used are shown in Table 3.

5.2 Simulation results

We compared the proposed CcEbH with DAlPaS [21], which is a dynamic alternative path selection based congestion control scheme. In DAlPaS, the network is divided into different levels from sink.

We evaluate the performance of the CcEbH at different interval time of sending packets. It is shown from Figs. 3*a* and *b* that the source nodes send the same number of packets in DAlPaS and CcEbH in different network models. In the network with 49 nodes, the sink node receives the same amount of packets when the send rate is low. However, the amount of packets received by the sink is different from the number of packets sent by the source nodes with increasing of send rate. That is because that the amount of packets injected into the network has exceeded the service capability of WSNs and leads to the occurrence of congestion. The CcEbH can deal with the network congestion well. Figs. 3*a* and *b* also show that although the number of received packets is less than the amount of sent packets for CcEbH and DAlPaS in different network model, the CcEbH receives more packets than the DAlPaS while the interval time is less than 0.18 s in the network model with 49 nodes. Meanwhile, the CcEbH can receive more packets less than the DAlPaS in the network model with 100 nodes, which proves that the CcEbH can deal with the network congestion well.

It is seen from Fig. 4 that the CcEbH decreases the packet drop rate compared with the DAlPaS in different network model. This result shows that the CcEbH is efficient and can improve the performance of network. To avoid packet drops, the CcEbH executes the congestion avoidance method at first. The CcEbH can transmit the data packets to the same hierarchy nodes when all the lower-hierarchy nodes is unavailable, which will also decrease the probability of packet drops.

To further analyse the performance of CcEbH, we calculate the average end-to-end latency of data packets and the throughput. Fig. 5 shows that the average end-to-end latency is same approximately in the network model with 49 nodes, and in the network model with 100 nodes, the CcEbH can decrease the average end-to-end latency which can improve the network performance. The CcEbH can send the data to the same hierarchy nodes when the congestion occurs. The memory of the same hierarchy nodes can be used to store more packets, which will increase the throughput of network. Fig. 6 illustrates that the maximal throughput of the CcEbH is 137 and 72 kbit/s, whereas the largest throughput of the DAlPaS is 128 and 60 kbit/s. The CcEbH increase the throughput of network by 7–20%, compared with the DAlPaS.

In order to evaluate the efficiency of the CcEbH in balancing the energy consumption, we count the average residual energy of all the nodes located at the same level after 300 s in the network with 49 nodes. In this network, the hierarchical topology divides the network into six hierarchies. From the high hierarchy to the low hierarchy, the average residual energy decreases in Fig. 7, and the network with 100 nodes has the same characteristic. This condition conforms with the energy consumption of traditional sensor network [10]. The nodes located at the low hierarchy exhaust their energy quickly. So in the balanced energy consumption scheme, the network lifetime is maximised by balancing the energy consumption of the low hierarchy.

We also compute the variance of the remaining energy of all the nodes located at the same hierarchy. Results in Figs. 8 and 9 are the

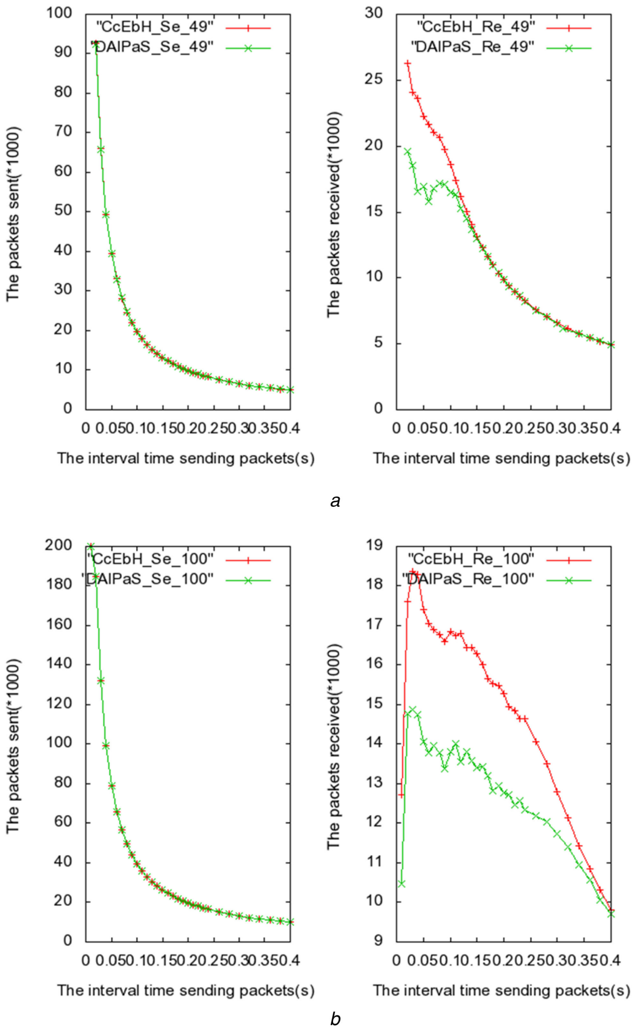

Fig. 3 *Number of packets received and sent by the sink and source nodes in the network*
(a) Case with 49 nodes, ***(b)*** Case with 100 nodes

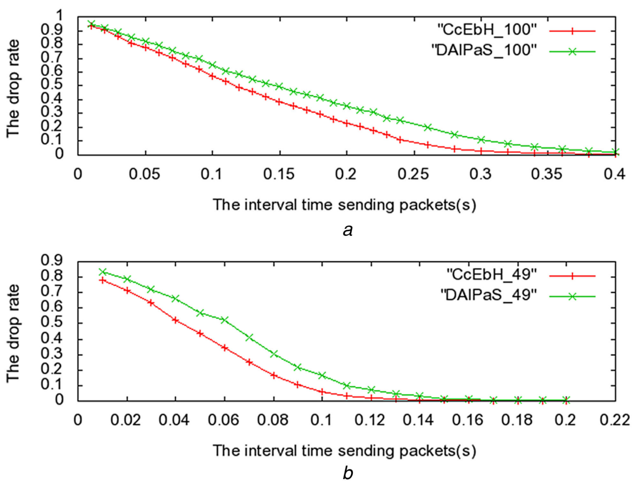

Fig. 4 *Packet drop rate*
(a) Case with 100 nodes, ***(b)*** Case with 49 nodes

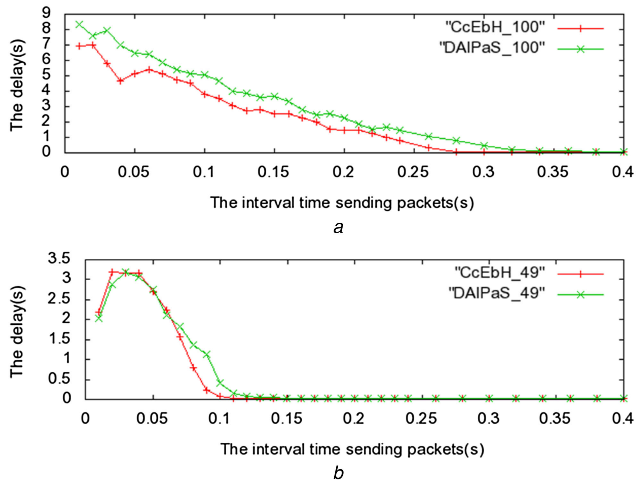

Fig. 5 *Average end-to-end latency of data packets*
(a) Case with 100 nodes, ***(b)*** Case with 49 nodes

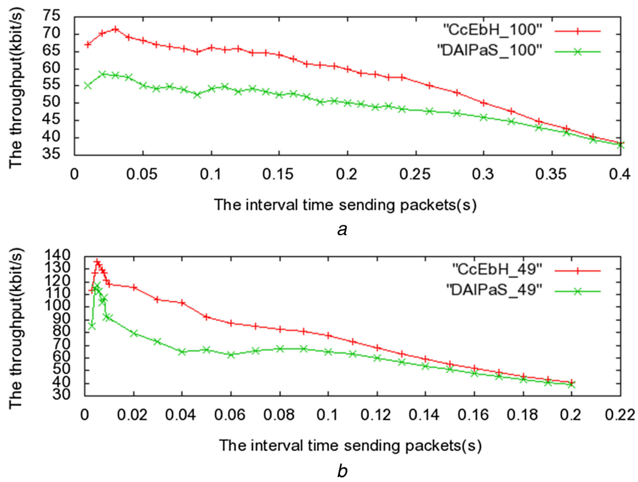

Fig. 6 *Network throughput*
(a) Case with 100 nodes, ***(b)*** Case with 49 nodes

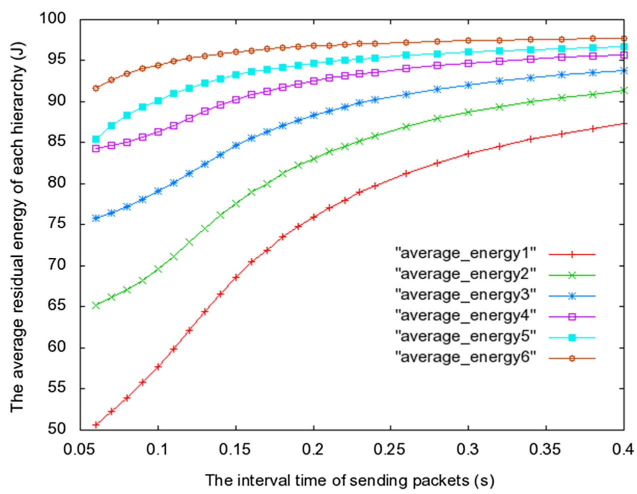

Fig. 7 *Average residual energy of each hierarchy in the network with 49 nodes*

variance of residual energy in the network models with 49 nodes and 100 nodes, respectively. It is noted that the variance of the remaining energy of the inner-most hierarchy nodes is close to 0, in other words, the inner-most hierarchy nodes dissipate the same energy. That is because the node chooses the lower-hierarchy node with the most remaining energy to receive data packets, which ensure the even energy dissipation of the lower-hierarchy nodes of a certain node. In the second part of balancing energy consumption strategy, It utilises the same hierarchy nodes to guarantee that the even dissipation of all the nodes have the same hierarchy value. It is also discovered that the variance increases with the improvement of the hierarchy shown in Figs. 8 and 9.

To further illustrate the situation of energy consumption of the network, we present the remaining energy of different networks shown in Figs. 10 and 11. The nodes are deployed uniformly over $800 * 800\ \mathrm{m}^2$ area and $1400 * 1400\ \mathrm{m}^2$ area. The interval time of sending packets is 0.05 s. The base station is located at (800, 0) and

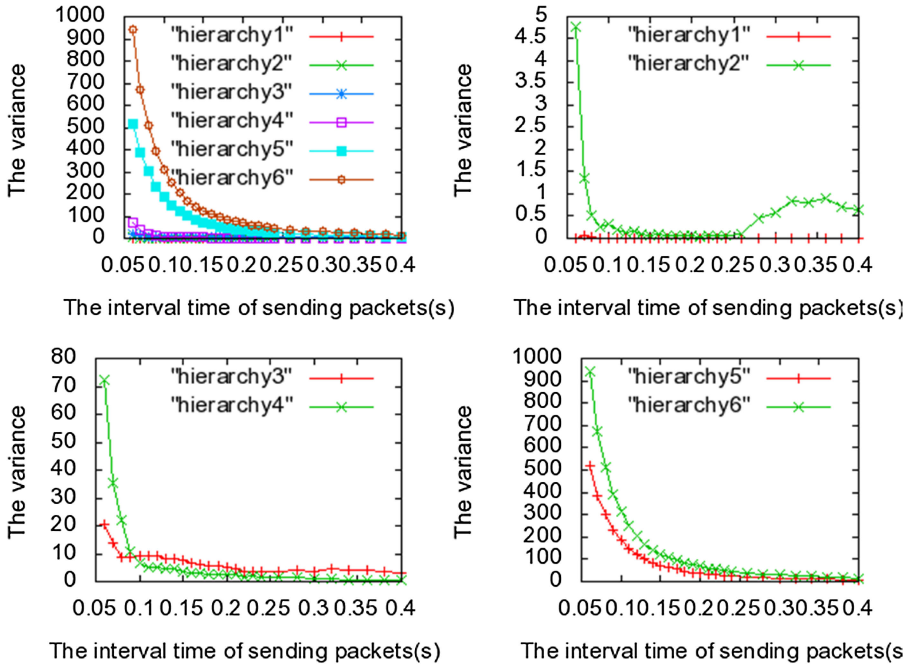

Fig. 8 *Variance of residual energy of each hierarchy in the network with 100 nodes*

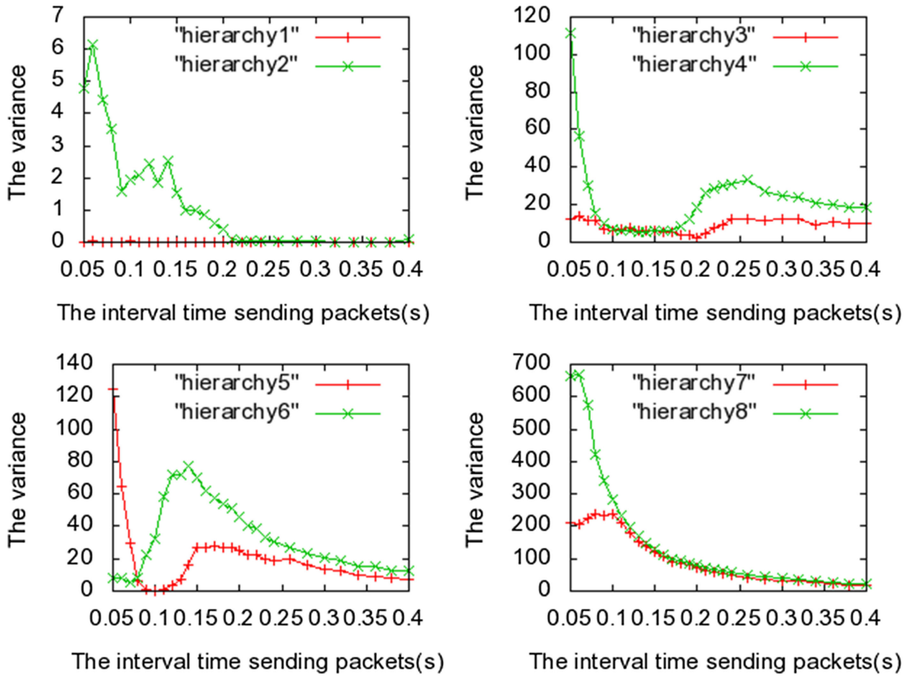

Fig. 9 *Variance of residual energy of each hierarchy in the network with 100 nodes*

(1400, 0). Figs. 10 and 11 show that the nodes expend identical amount of energy, when the nodes around the sink node have same hierarchy value. This mode of power dissipation can prolong the lifetime of sensor network extremely.

Figs. 10 and 11 also demonstrate that the fringe nodes dissipate uneven power, which is caused by the balanced energy dissipation method. The method only balances the energy consumption of the lower-hierarchy nodes of the forward nodes. However, the fringe nodes are not responsible for forwarding the data packets. They just send the data packets that they have sensed. So, the fringe source nodes consume more energy than other fringe nodes. This circumstance will not reduce the lifetime of sensor network because the remaining energy of fringe nodes is greater than the remaining power of inner-most hierarchy nodes.

6 Conclusions

In this work, we have proposed a congestion control and energy-balanced scheme based on the hierarchy topology for WSNs (CcEbH), in which the congestion avoidance method and the congestion control mechanism have been also given to decrease the probability of congestion occurrence. The proposed strategy can balance the power consumption of the low hierarchy nodes by compelling the node to forward the packets to the lower-hierarchy node with the most residual power firstly. This strategy also takes advantage of the neighbouring nodes that have the same hierarchy value to balance the energy dissipation of all the downstream nodes at same level.

7 Acknowledgment

This work was supported by National Natural Science Foundation (NNSF) from China (61273073, 61374107, 61304044).

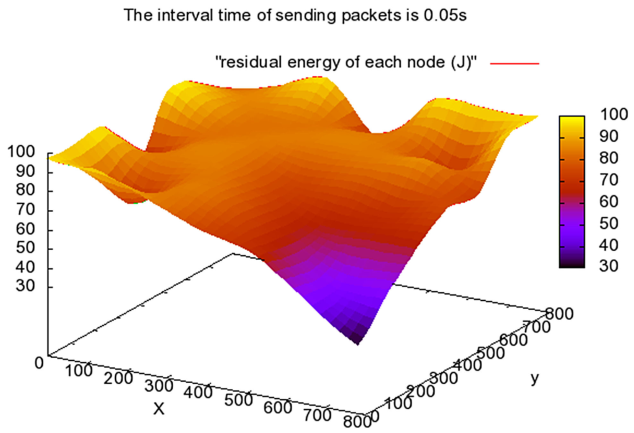

Fig. 10 *Distribution diagram of residual energy in the network with 49 nodes*

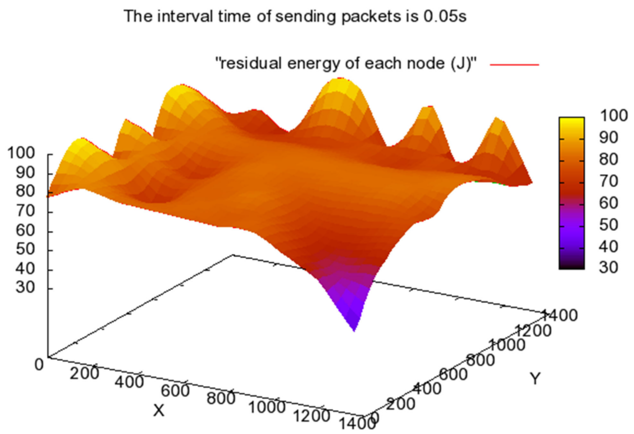

Fig. 11 *Distribution diagram of residual energy in the network with 100 nodes*

8 References

[1] Rahman, K.C.: 'A survey on sensor network', *J. Comput. Inf. Technol.*, 2010, **1**, (1), pp. 76–87

[2] Yick, J., Mukherjee, B., Ghosal, D.: 'Wireless sensor network survey', *Comput. Netw.*, 2008, **52**, (12), pp. 2292–2330

[3] Wang, C., Sohraby, K., Li, B., *et al.*: 'A survey of transport protocols for wireless sensor networks', *IEEE Netw.*, 2006, **20**, (3), pp. 34–40

[4] Ghaffari, A.: 'Congestion control mechanisms in wireless sensor networks: a survey', *J. Netw. Comput. Appl.*, 2015, **52**, pp. 101–115

[5] Flora, J.: 'A survey on congestion control techniques in wireless sensor networks'. Int. Conf. on Emerging Trends in Electrical and Computer Technology, 2011

[6] Kafi, M.A., Djenouri, D., Ben-Othman, J., *et al.*: 'Congestion control protocols in wireless sensor networks: a survey', *IEEE Commun. Surv. Tutor.*, 2014, **16**, (3), pp. 1369–1390

[7] Sergiou, C., Vassiliou, V.: 'Alternative path creation vs data rate reduction for congestion mitigation in wireless sensor networks'. Proc. of the 9th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks, 2010

[8] Ansari, M.S., Mahani, A., Kavian, Y.S.: 'Energy-efficient network design via modelling: optimal designing point for energy, reliability, coverage and end-to-end delay', *Netw. IET*, 2013, **2**, (1), pp. 11–18

[9] Li, J., Mohapatra, P.: 'Analytical modeling and mitigation techniques for the energy hole problem in sensor networks', *Pervasive Mob. Comput.*, 2007, **3**, (3), pp. 233–254

[10] Stojmenovic, I., Olariu, S.: 'Data-centric protocols for wireless sensor networks'. *Handbook of Sensor networks*, John Wiley & Sons, NJ, USA, 2005, pp. 417–456

[11] Kleerekoper, A., Filer, N.P.: 'DECOR: distributed construction of load balanced routing trees for many to one sensor networks', *Ad Hoc Netw.*, 2014, **16**, pp. 225–236

[12] Karim, L., Nasser, N.: 'Reliable location-aware routing protocol for mobile wireless sensor networks', *IET Commun.*, 2012, **6**, (14), pp. 2149–2158

[13] Gagarin, A., Hussain, S., Yang, L.T.: 'Distributed hierarchical search for balanced energy consumption routing spanning trees in wireless sensor networks', *J. Parallel Distrib. Comput.*, 2010, **70**, (9), pp. 975–982

[14] Khedr, M.A, Omar, D.M.: 'SEP-CS: effective routing protocol for heterogeneous wireless sensor networks', *Ad Hoc Sensor Wirel. Netw.*, 2015, **26**, pp. 211–234

[15] Wang, G., Liu, K.: 'Upstream hop-by-hop congestion control in wireless sensor networks'. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications, Tokyo, Japan, 2009, pp. 1406–1410

[16] Annie, U.R., Kasmir Raja, S.V., Antony, J., *et al.*: 'Energy-efficient predictive congestion control for wireless sensor networks', *IET Wirel. Sens. Syst.*, 2015, **5**, (3), pp. 115–123

[17] Antoniou, P., Pitsillides, A.: 'A bio-inspired approach for streaming applications in wireless sensor networks based on the Lotka–Volterra competition model', *Comput. Commun.*, 2010, **33**, (17), pp. 2039–2047

[18] Yin, X., Zhou, X., Huang, R., *et al.*: 'A fairness-aware congestion control scheme in wireless sensor networks', *IEEE Trans. Veh. Technol.*, 2009, **58**, (9), pp. 5225–5234

[19] Sergiou, C., Vassiliou, V., Paphitis, A.: 'Hierarchical tree alternative path (HTAP) algorithm for congestion control in wireless sensor networks', *Ad Hoc Netw.*, 2013, **11**, (1), pp. 257–272

[20] Antoniou, P., Pitsillides, A., Blackwell, T., *et al.*: 'Congestion control in wireless sensor networks based on bird flocking behavior', *Comput. Netw.*, 2013, **57**, (5), pp. 1167–1191

[21] Sergiou, C., Vassiliou, V., Paphitis, A.: 'Congestion control in wireless sensor networks through dynamic alternative path selection', *Comput. Netw.*, 2014, **75**, pp. 226–238