# Alternative Path-based Congestion Control in Many-To-One Sensor Networks

Kunliang Zhao, Weidong Liu, Ming Sze Wong, Jiaxing Song
Department of Computer Science and Technology
Tsinghua University
Beijing, China

*Abstract*—**Compared with traditional wireless networks, the congestion control design of Wireless Sensor Networks (WSN) is more significant to the network performance due to its own characteristics. In this paper, we design a mechanism using spare resources to mitigate congestion in a many to one network. We classify the node congestion into two categories according to the reasons and propose two distributed algorithms which can find an alternative path using the neighbor table efficiently. At the end of this paper, we simulate the drop rate of WSN to evaluate our mechanism. The simulation shows that using our mechanism improves network transmission quality significantly.**

***Keywords-wireless sensor network; congestion control***

## I. INTRODUCTION

Wireless Sensor Network (WSN) is composed of a large number of autonomous sensor nodes which are capable of gathering phenomenon, performing processing and communicating with other nodes [1]. The sensor detects sensory information and sends the data to sink node through short-range, a multi-hop transmission. Compared with the traditional wireless ad-hoc networks, WSNs get their unique features such as the limit power and memory of sensor nodes, densely deployed in the phenomenon, the multi-hop tree topology of many to one characteristics and the mass burst data when sudden changes happened in the environment [2]. These characteristics of sensor nodes and WSN cause localized and transient congestion, which easily lead to a packet loss, reducing the link utilization and other issues to reduce the reliable of network [2], [3], [4].

This paper focuses on the congestion of WSNs occurred by sudden data. We design a scheme of using redundancy resources to generated diversion paths dynamically in tree topology. The paths will distribute the sudden traffic to reduce the congestion rapidly and keep the sink node collecting information accurately. Current mechanism of congestion control introduces some approaches to relieve congestion, and lots of them focus on how to restrict the traffic rate to alleviate congestion. Although some researches design multiplexing paths [4] or alternate paths to increase the reliability of networks, these mechanisms not only use few extra nodes and paths resources effectively which may easily cause power wasting, but also ignore the many-to-one characteristic of WSN.

Consider of the different causes， we investigate a congestion control mechanism of using extra network resources to eliminate the congestion. The following WSN features are considered; first, as the energy limitation of sensor node, we should use the existing route path as the alternative path instead of generating a new path. Second, consider in tree topology networks, we design a distributed algorithm to select the appropriate path quickly at the same time to avoid new congestion.

The rest paper is divided into three sections; the first section introduces related work and defines the problems, the next part describes the details of the alternative path congestion control mechanism, the final part simulates the scheme and concludes the paper.

## II. RELATED WORKS

Some researches have proposed some reliable multi-path routing protocols for wireless Ad-hoc networks [4], [10]. This paper focuses on using alternative method to mitigate the WSN congestion. Comparing with multi-path protocols, we only establish alternative paths when detect the congestion and we will use redundancy path to forward traffic to save power.

Recently, the congestion control is a hot topic of WSN research. CODA [3] is one of the typical congestion control mechanisms of wireless sensor networks, which uses restricting source rate to mitigate the congestion. CODA introduced a open-loop and closed-loop technique. IFRC [5] proposed one fair mechanism work on tree topology. It defined the interference node of WSN. When the congestion occurred in the tree topology, IFRC designed that nodes share congestion indicators instead of hop-by-hop back-pressure to reduce source rate as well as keep fire of transmit.

Gansesan [9] introduced the idea of multi-path scheme to ensure transmission quality in WSN. ARC [4] proposed using adaptive resource control to mitigate the congestion. ARC introduced how to use adaptive resources in WSN to set up a backup path based on neighbor table to deliver congestion traffic. But ARC ignored lots of idle path in WSN and only generated new path when detected the congestion and released the path after the alleviate congestion, which would consume valuable energy of sensor nodes.

Base on the impact, WSN congestions can be classified into local congestion and global congestion. Local congestion which occurs in part region is usually caused by link or node failure. Global congestion is the type occurs in the entire network and due to the amount of data need transmit excess the

CHINACOM 2010, August 25-27, Beijing, China

DOI 10.4108/iwiot.2010.2

network capacity, while it reduces the data collection rate. Since wireless sensor networks have lots of available resources as densely distributed nodes, the global congestion is rare while the local congestion will often be generated by the environment of monitor region changes and emergent data. This paper focuses on discovering and solving local congestion in time.

Recent researches on WSN congestion propose some effective congestion detection and reporting methods. Congestion detection is based on buffer queue length [2], [3], [4], [6] or channel condition [3], [7], [8]. Congestion notification may be sent by notice packets [3], [6] or set the congestion bit of the sending packet [4], [5], and the neighbor nodes will listen to the packets to update congestion status.

As the congestion will bring packets drop to WSN, the sink node cannot receive valid data that sensor nodes detect and the environment would be failed to monitor. Congestion can avoid losing packet and improve network reliability. We will evaluate our mechanism performance by calculate packets congestion nodes' drop rate and total packets delivery.

## III. Considerations and Descriptions

### A. Design Considerations

As described before, we use redundancy resources to mitigate local congestion, and the existing idle path is priority selected to save energy [4]. We divide the local congestion into two categories according to causes. First, due to the node energy depletion or the link failure in hotspot, the packets cannot be transmitted cause congestion. On the other hand, when many source nodes attempt to send data to the sink node and several flows will aggregation to a node, large amounts of data may exceeding the node processing capacity results in a congestion.

When a sensor node losses power because of energy consumption or mutations, it will fail to send data to the next hop and the packets will be dropped. The mechanism should find an alternative path to transmit the data, as well as making sure that the failure node is not on the path and the path will not interfere with the failure node. Also while several flows aggregation to one node, these flows need be transmitted immediately. If the data excess the node processing capacity, the design should use the free path to forward some of these flows which should be determined efficiently to avoid new congestions.

### B. Description and Neighbor Table

Consider a WSN with densely dispersed sensor nodes, and the set of the sink nodes is $S=\{s_i\}$. It's easy to notice that $s_i$ is the root of route tree.

Denote node's max forwarding rate is $F^0$, which is related to the network environment. As soon as the sensor nodes self-organize a network, the $F^0$ of each node can be set up. For fairness and efficiency of the transmission, we keep $F_i^0 \leq F_j^0$, in which node $j$ is one of node $i$'s child nodes [5]. The node's available forwarding rate $F_t$ at time $t$ can be determined by the total transmit rate. We can dynamic adjust $F_{t,i}=F_i^0-f_t$, in which $f_t$ indicates the sum rate of all data flow through node $i$. Take node $i$ for instance, it's easy to conclude that to keep $i$ away

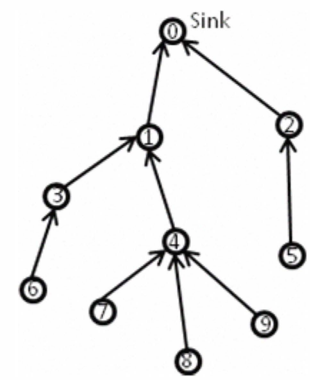


Figure 1. Sub-net of WSN.

from congestion, the sum of its child nodes' rate can't excess $F_t$. Accordingly, if node $i$ or its ancestor node is congestion , we can set its $F_t$ zero to indicate this node is over capacity. As a node's child node starts or stops sending data or the network topology changes, its $F_t$ should be adjusted, and the new value of $F_t$ should be forwarding to neighbor nodes to help them to update neighbor table.

Because of the dense nodes, the WSN get a large size, we can use the redundancy resources to build an alternative path. Because it is too expensive to save entire network status, we design the neighbor table only to store necessary information saving sensor nodes' buffer and energy. The neighbor table of node 7 in figure 1 is shown in table 1.

In the table, ID indicate the node's universal ID, $H$ is the node's height in the stable route and the sink nodes' height is 0, $N$ is the node's next hop node. Every node should record neighbors in the table as well as its own height and $F_t$. After sensor nodes deploy in monitor area, they will establish tree topology, and then nodes get there height and father nodes. These information need to be send to neighbor nodes through broadcasting, after a while, every node would set its neighbor table.

## IV. Mechanism Design

In this section we will describe the design's details of the congestion detection and how to establish an alternative path.

### A. Congestion Detection

Once the congestion occurs, how to detect the congestion and report its status is the key to rapidly mitigate congestion.

Considering two different kinds of local congestion as we described before, if congestion is caused by node failure, the failure node will set its $F_t$ to 0. As discussed before if any child

TABLE I. Example of Neighbor Table

| ID | Height(H) | Father node(N) | Available forwarding rate(Ft) |
|---|---|---|---|
| 3 | 2 | 1 | 5 |
| 4 | 2 | 1 | 3 |
| 6 | 3 | 3 | 0 |
| 8 | 3 | 4 | 8 |

nodes try to send data to the failure node, it will notice failure node's $F_t$ is 0 which indicates it's a congestion node. The child nodes will change its neighbor table and start to find alternative path. On the other hand, when several nodes send packets to one node, the buffer occupation can represent the congestion state [2], [3], [5], [10]. When the buffer queue crosses threshold $U$ [5], the node is regarded to be congestion. In our design we use diversion path to mitigate congestion instead of adjusting data rate frequently, so we set threshold $U$ to be a constant. Generally the threshold may set to be 0.9 times of queue length [4]. We set the congestion bit of forwarding packets to notify congestion condition to other nodes. Any node listened the congestion information will update its neighbor table of setting congestion node's $F_t$ to 0. The child nodes listened the congestion information will start to find a new forwarding path according to the mechanism.

### B. *Alternative Path Searching Algorithm of Failure Node Type Congestion*

Consider a failure node $a$, any source node trying to send data to sink node through node $a$ will bring congestion. All the child nodes will start find another forwarding path using the algorithm shown in figure 2.

Since sink node is the root of the route tree, the node with a smaller height is closer to the sink node, so that it will cost less energy for transmitting. For nodes at the same level, the algorithm should guarantee it's not a child node of node $a$. In case the diversion path is congestion too, the $F_k$ should be bigger than zero. Once node i find the candidate diversion node $k$, the request packet will be sent by node $a$ to node $k$. After $k$ response the request, the alternative path will be established and node i can forward data through the new path. If node $i$ doesn't receive a response it will look for other neighbor nodes.

If a node doesn't find a candidate node in step 3 or 4, instead it will send a finding diversion path request to the node with minimum $H_k$ and non-zero $F_k$. The request contains the request node ID, the congestion node ID, and the height of the congestion, the node receiving this request will find an alternative path in its neighbor table recursively. Take an example, the node $j$ receives the request ($i$, $a$, $H_a$), and it will execute the following steps in figure 3.

As the algorithm shown in figure 3, if the node $j$'s height is lower than the congestion node, the diversion path can be set up immediately. In step 4, 5 and 6, the node should check whether it uses the same path with node $i$, if not, this node can be used to transmit the congestion flow. In other condition, it will find a path recursively as realized in step 7. In case of circle request, once a node start finding a alternative path, it will set a flag. Any nodes receive finding path request should check the flag first, it will run the algorithm if only the flag is not set. Because there are a large number of idle nodes, shortly we can find a diversion path to transmit the packets from congestion area.

As part of WSN shown in figure 4, node 4 can't forward packets from node 6 and 7 anymore, and then node 6 and 7 find new paths to send data through node 3 and node 8 instead.

1. sort all nodes in the neighbor table except father node according to ascending order of H
2. for each node $k$ in nodes array:
3. if $H_i > H_k$ and $F_k > 0$ send forwarding request and wait for a response
4. else if $H_i$–$H_k$ = 1 and $N_k$ != $N_i$ and $F_k$ > 0 send forwarding request and wait for a response
5. else in $F_k$ > 0 nodes select the node with minimum $H_k$, send finding path request wait for a response
6. receive response, start forwarding packets using diversion path

Figure 2. Searching algorithm of node i.

1. $\Delta h = H_j - H_a$
2. switch($\Delta h$):
3. case <0: return the request, find path
4. case 0: if $j$!=$a$ return the request, find path
5. case 1: if $N_j$!=$a$ return the request, find path
6. case >1: check if the ancestor node of $\Delta h$ hops is node a, if not find path
7. select a node m with the minimum height except node $i$ or $a$, send ($i$, $a$, $H_a$)request
8. response from m, find path

Figure 3. Recursive search algorithm of node j.

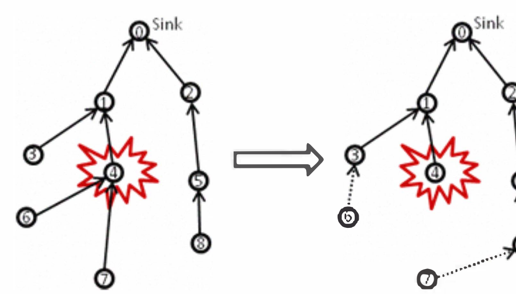


Figure 4. Diversion Paths Scheme on Failure Node.

### C. *Alternative Path Searching Algorithm of Aggregate Flows Type Congestion*

As described before, several flows aggregating to one node will cause local congestion. Take figure 5 for instance, node 7, 8 and 9 send data to node 4 at the same time, the forwarding rate excess the node's capability, and the buffer of node 4 is cram, so node 7, 8 and 9 should find their new paths to send packets.

When occupation of node $a$'s queue excess threshold U, node $a$ is regarded as a congestion. The node set its $F_a$ to 0 and sends the congestion information to child nodes. The child node $i$ will check its buffer after receiving the congestion information, if in buffer there is any packet which is waiting to be sent to node $a$, the node $i$ should start finding $a$ diversion path. Set the forwarding rate of node $i$ be $r_i$, the algorithm is described in figure 6.

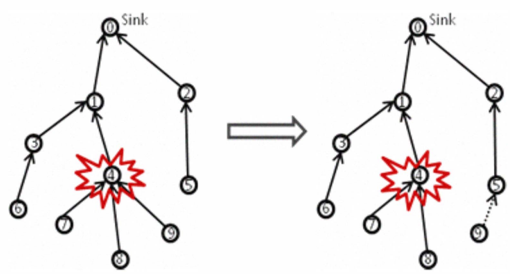


Figure 5. Diversion Paths Scheme on Aggregate Flows.

| |
|---|
| 1. sort all nodes in the neighbor table except father node according to ascending order of H |
| 2. for each node k in nodes array: |
| 3. if $H_i > H_k$ and $F_k > r_i$ send forwarding request and wait for a response |
| 4. else if $H_i–H_k = 1$ and $N_K \,!= N_i$ and $F_k > r_i$ send request and wait for a response |
| 5. else if $N_i$ is node $k$'s ancestor node: |
| 6. continue |
| 7. else send forwarding request and wait for a response |
| 8. return congestion node, report if find a diversion path |

Figure 6. Searching Algorithm

When congestion node receives its child nodes' returns, it will decide which child node will use diversion path to forward data. Let $S=\{i\}$ represents child node set of the congestion node $a$, $r_i$ indicates the data rate of $i$ forward to a, $D=\{d\}$ denotes the set of nodes find alternative path and the size of set $D$ is $k$. According to max forwarding rate's definition, cause of node $a$ is congestion, so we can know,

$$F_{a,t} = F_a^0 - \sum_{i}^{i\in S} r_i < 0 \qquad (1)$$

Descending sort nodes in set D by their rates. For the ordered set $D=\{ d_i \mid r_{di}>r_{di+1}\}$, find a integer $m <=k$, let

$$F_{a,t} + \sum_{j}^{j\in D, j<m-1} r_j \le 0 \le F_{a,t} + \sum_{j}^{j\in D, j<m} r_j \qquad (2)$$

Then the congestion node sends their response to $\{d_1, d_2 \ldots d_m\}$, the nodes receive the response starting to forward packets through diversion paths. If node $a$ can't find integer $m$, all nodes in set $D$ begin to forward traffic through the alternative path and other nodes in set $S$ may slow down their rates to mitigate congestion.

TABLE II. SIMULATION SETTING

| Parameter | Value |
|---|---|
| Area of simulation field | 500m * 500m |
| MAC | 802.11 |
| IFQ length | 50 |
| RXThresh | 2.81838e-9 |
| Packet Size | 128 |

Take figure 5 for example, node 4 is a congestion, node 7, 8 and 9 start to find an alternative path using the algorithm described in figure 6. Assumed node 7, 9 find diversion paths and node 4 notices that only need node 9 to forward flow to alleviate congestion, so node 9 sent packets through node 5, and node 6, 8 still use node 4 to forward the traffic.

## V. SIMULATIONS

We simulate the diversion path on NS2 to evaluate our mechanism. We design two different scenarios to simulate two categories of local congestion. The detailed NS2 setting is shown in table 2.

### A. Congestion Caused by Node Failure

For node failure type congestion we set a sub-network as shown in figure 7, in which node 0 is the sink node, node 6 is the source node and node 4 is out of power bringing congestion. According the algorithm describes in section 4.2, the source node 6 will send data through node 5. We will evaluate the transmission result by using or not using an alternative path.

We add up the total number of packets sink we received and calculate the loss rate of congestion node to evaluate the influence of the congestion. The simulation results is as shown in figure 8, the red curve indicates the results of using diversion path comparing with the blue curve without the diversion path, and the x-axis is the different source rate of the traffic.

As shown from the left figure, when there is congestion in the network, by using an alternative path, we can significantly increase the number of received packets. Also from the right figure, we can see using a diversion path will decrease the drop rate from 0.7 to 0.1.

### B. Congestion Caused by Flow Aggregation

We evaluate the second algorithm using the sub-net as shown in figure 5. Assume that node 7, 8 and node 9 try to send data to sink through node 4, as we discussed before, the node 4 will congest. We simulate different drop rates in different source rates to evaluate the mechanism.

First we increase the source rate of all 3 nodes, and the result is shown in the left of figure 9. The blue line indicates the drop rate of the entire net without flow diversion, while the red curve shows the result of using alterative path. We can notice that the green drop rate is obviously higher than the blue one.

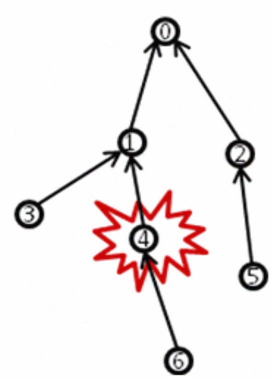

Figure 7. Ccenario of Sub-network.

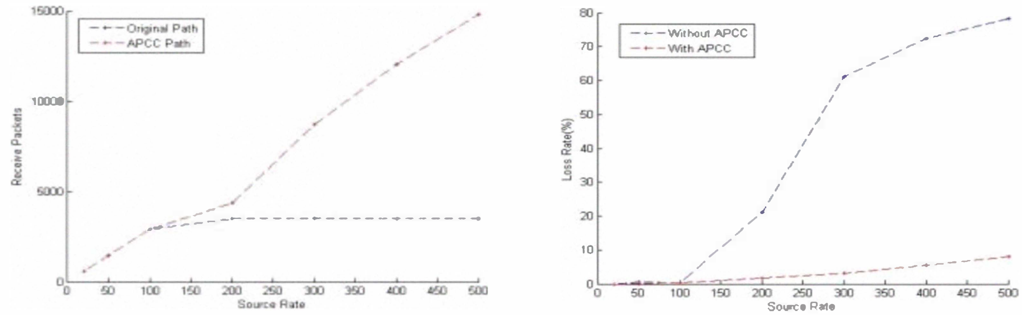


Figure 8. Results of Failure Node Congestion.

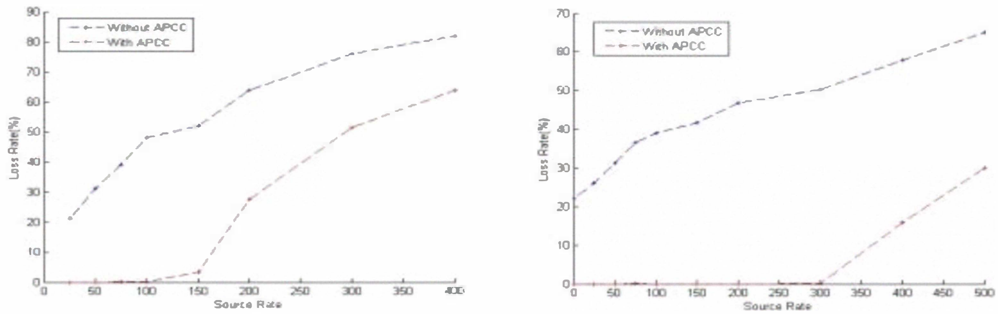


Figure 9. Result of Flow Aggregation Congestion.

Also we lock up the data rate of node 7 and node 8, and dynamically enhance node 9's sending rate. The result is shown in right of figure 9. Clearly without the transmitting flow through alternative path there is notable congestion when the rate of source node just excess 50kbps, compared using alternative the congestion drop appears when the source rate excess 400kbps.

According to the simulate result, for both node failure and flow aggregation congestion, using the diversion path to forward traffic will mitigate the congestion and increase the network reality.

## VI. CONCLUSIONS AND FUTURE WORK

This paper represents the categories of local congestion of wireless sensor networks. Based on using spare resources idea we proposed a new mechanism to mitigate the congestion using the vacant path in the network instead of restricting the transmit rate. We describe the detail algorithm of finding alternative path rapidly and choosing the proper diverting flow. We also simulate to evaluate our design in NS2.

In this paper we define the $F^0$ to indicate the capacity of a single node. But in the real WSN environment the node's capacity is a more complex parameter, which related to environment, topology, MAC protocol and node's energy. To determine node's capacity is a complicated task, we may investigate in the future work.

## REFERENCES

[1] F. A. Ian, S. Weilian, S. Yogesh and C. Erdal, "A Survey on Sensor Networks," IEEE Communications Magazine (2002), pp. 102-114

[2] Y. Sankarasubramania, O. B. Akan and I. F. Akyidiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," In: Proc of the 4th ACM international symposium on Mobile Ad Hoc Networking and Computing. New York : ACM Press , vol. 20031, pp. 177 -188

[3] Y. C. Wan, S. B. Eisenman and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," In: Proc of the 1st ACM Conf on Embedded Networked Sensor Systems. New York : ACM Press , vol. 20031, pp. 266 – 279

[4] J. Kang, B. Nath and Y. Zhang, "Adaptive resource control scheme to alleviate congestion in sensor networks," In: Proc of the 3st IEEE Workshop on Broadband Advanced Sensor Networks (BASENETS). San Jose , USA : IEEE Communications Society , 2004

[5] S. Rangwala, R. Gummadi, R. Govindan and K. Psounis, "Interference-Aware Fair Rate Control in Wireless Sensor Networks," In: Proc. of SIGCOMM'06, September 11–15, 2006, Pisa, Italy.

[6] C. T. Ee, R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," In: Proc of the 2nd ACM Conf on Embedded Networked Sensor Systems. New York : ACM Press , vol. 20041 pp. 148 - 161

[7] L. Popa, C. Raiciu, I. Stoica and D. S. Rosenblum, "Reducing congestion effects in wireless networks by multipath routing," In: Proc. of the 14th IEEE Int'l Conf. on Network Protocols (ICNP). Santa Barbara: IEEE Press, 2006. pp. 96−105.

[8] Y. C. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: Overload traffic management using multi-radio virtual sinks in sensor networks," In: Proc. of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys). San Diego: ACM Press, 2005. pp. 116−129.

[9] D. Gansesan, R. Govindan, S. Shenker and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," Mobile Computing and Communications Review. 2002, 1(2).

[10] S. De, C. Qiao, and H. Wu, "Meshed multipath routing with selective forwarding: an efficient strategy in wireless sensor networks," Elsevier Computer Networks, Vol. 43, pp. 481-497, 2003.