International Journal of Smart Home
Vol. 10, No. 9 (2016), pp. 179-196
http://dx.doi.org/10.14257/ijsh.2016.10.9.17

2C-CTP: A Centralized Clustering Data Collection Protocol based on CTP

Lingyun Xu[1,2], Tao Li[1,2], Baowei Wang[1,2], Xingming Sun[1,2], Xingang You[2,3], Yanhua Zhu[4]

[1]Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, China
[2]College of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China
[3]Beijing Institute of Electronic Technology Application, Beijing 100091, China
[4]College of Languages & Cultures, Nanjing University of Information Science & Technology, Nanjing 210044, China
18512503721@163.com, nuistmail@163.com, wbw.first@163.com, sunnudt@163.com

Abstract

WSN (wireless sensor network) is widely applied in environmental monitoring, assets tracking, battlefield monitoring and smart building. Since nodes in WSN are constrained by finite battery power and operational capability, it will have the direct effects on the lifetime of whole network. Well-designed routing protocols could make network succinct and efficient. In order to prolong the overall lifetime of WSN, lots of clustering algorithms have been proposed. Most of them are based on LEACH and proved energy-efficient, but few are implemented in real systems. CTP (collection tree protocol) has been extensively used in many systems and provides a reliable protocol for data collection. However, it neither considers energetic balance nor follows clustering feature. In this paper, we propose an efficient routing protocol by improving the CTP, adding clustering and centrally controlled features. Furthermore, we introduce a lightweight method of command distribution and some implementation technologies of clustering based on CTP. The related experiments that carried on TelosB demonstrate that our scheme is outstanding in lifetime and efficiency.

Keywords: *Wireless sensor networks, CTP, clustering routing, centrally controlled*

1. Introduction

Wireless sensor network (WSN) is one of supporting technologies for Internet. It has prospects for broad application with characteristics which are different from the traditional wireless networks. In the wireless sensor network, routing protocol is responsible for the forwarding data packet from the source nodes to the destination nodes through the network [1-5]. It includes two main functions: searching for the optimal approach from the source nodes to the destination nodes; making the data packet forward along the optimal approach exactly. In the view of the hierarchical structure of the sensor nodes and the different functions in the routing process, the researchers put the routing protocol into two types: planar routing protocol and clustering routing protocol (hierarchical routing protocol) [6]. Clustering routing protocol has gradually been the focus of research, because its real effects outweigh the planar routing protocol. There are generally two strategies in the process of routing building: the distributing strategy and the centralized controlling strategy. However, we notice that among most of existing network protocols, whether is planar or clustering, sensor network protocol is mostly designed for distributing strategy. It is generally believed that the effects of using network information collected by centralization to calculate the best routing path is extremely inefficient. According to the study, it needs to spend one third of energy collecting information [7]. In addition, the distributing solutions have been proved that it has had ideal effects. This doesn't mean that the centralized controlling strategy makes no sense. The biggest advantage is that once the network configuration is complete, the required overheads for control operation is very small [8]. In the distributing solution, each node needs exchanging the information constantly to ensure the effective organization of the network. Moreover, we find that there are no distributing solutions that can compete with a well-designed centralized solution. For reasons above, we present a protocol based on the CTP (Collection Tree Protocol) [9-11], supporting clustering and being controlled centrally.

2. Related Work

CTP is one of the most promising data collection protocol for WSN that offers data delivery at high ratio with reducing costs. It is designed to collect information from many nodes into the sinking nodes. In general, it is composed of three parts: a link estimator, a routing engine and a forwarding engine. The link estimator evaluates the single-hop communications. Then, the routing engine uses this information to select the next hop. Finally, the forwarding engine is responsible for sending the traffic generated by the node itself as well as other nodes that need to be relayed.

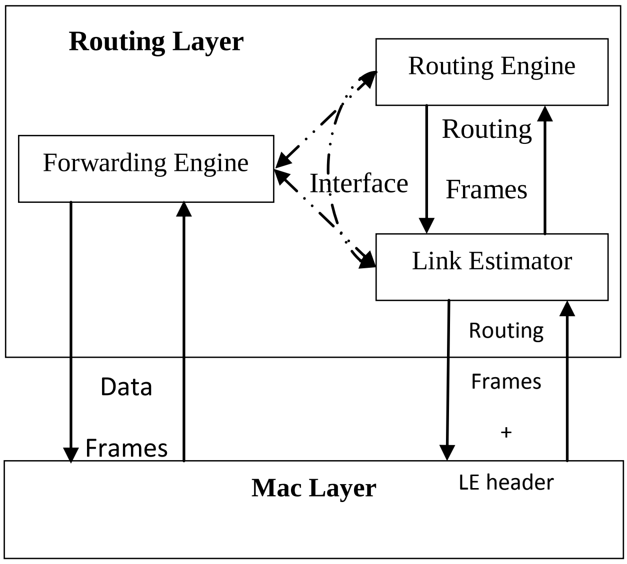


Figure 1. Module Interaction and Message Flow in CTP

The link estimation metric of CTP is based on the number of expected transmissions (ETX). This metric indicates how many times a message should be transmitted to reach its destination. In particular, the sink node has an ETX of 0, and each sensing node computes its own ETX as the ETX of its parent plus the ETX of its link to its parent. Finally, the parent selected is the node that provides the lowest ETX value.

CTP has been widely used in the recent years for both research and commercial products. In fact, it has been included in TinyOS 2.x [11]. Therefore, many projects are using CTP nowadays. For instance, CitySee [12] use this protocol for proposes of ecological surveillance and Powernet [13] for energy consumption monitoring.

As the discussion above, CTP is a tree-based routing protocol, and it contributes to the planar routing protocol. In addition, though, the CTP contains the key principles of the collection protocol, it doesn't take energy of nodes into account in the network. When the nodes are deployed in the network in different positions, the task of forwarding packets from those positions would be different from each other, which results in some nodes' energy consuming faster than others. In recent years, many researchers put forward improved CTP protocols to balance the energy consumption, extending the lifetime of the network.

EARBB [14] is an energy-aware routing protocol based on pointing the way of improvement of CTP. By modifying the beaconing scheme, it can provide a reliable and energy-efficient routing scheme for both collection and dissemination of information with beaconing packets exchanged between nodes and their neighbors. Its average lifetime is at least 20% longer than that of CTP. But the researchers make experimental tests with TOSSIM [15].

BCTP is a balanced version of CTP. It enhances CTP by enabling the network to migrate the load of the nodes under heavy traffic, avoiding some nodes draining their energy for all traffic passes through them. It uses a strategy to coordinate with the loads through the network and then balance energy consumption by nodes [16].

ICTP is a load balanced version of CTP. It uses both longer length with good link quality path and shorter length with bad link quality. It may decrease the reliability but improve efficiency to avoid congestion. The authors have shown that the energy consumption in ICTP is less than CTP in same scenarios for reducing possibility of congestion [17].

O-CTP is based on the investigation of WSN routing protocols behaviors in networks that are affected by interference. O-CTP is a hybrid routing protocol that uses the packet delivery at high ratio of opportunistic routing in error-prone networks and it is also a energy-efficient routing protocol [18].

However, it still has a lot to be improved and developed. In EARBB, minimum-cost routing tree will be established as beacon packets exchanged between nodes and their neighbors. But the number of beacon packets increases with the increasing numbers of neighbors. The energy consumption caused by information exchanges is considerable with a large-scale and larger-density network. In addition, the opportunity of re-network is also uncontrollable. In BCTP and ICTP the way to balancing traffic and avoiding hotspot may decrease the reliability and the factor of residual energy is not considered.

In this paper, we propose an idea to add features of clustering and centralized controlling to CTP. Aimed at balancing energy consumption by clustering algorithm with the global information, we design a command distribution technique with utilizing routing table established by CTP adequately. In addition we also modify the format of the beacon for clustering feature. As a result, additional energy consumption is kept within due bounds because of the good interaction with CTP. The opportunity of re-network and command sending is controllable, and the command packet is very simple. The hotspot is avoided effectively due to the centralized clustering. The experiment shows that the network lifetime can be prolonged without decreasing the total packets receiving rates of the original CTP.

3. Design of 2C-CTP

2C-CTP is based on the CTP protocol. The operational principles of the protocol are as following: At the beginning of the sensor node deployment, all the nodes transmit packets and organize the network in the manner of CTP. These packets contain the data what the nodes need to monitor and the state of nodes themselves like the electric information. After a certain period, the network has enough information to give back to the base station. Meanwhile we have already collected the global information of entire network. We make a solution: The solution utilizes the global information which has been retrieved to calculate a global optimal clustering result in PC. And then PC sends the results of clustering to each node in a network through ROOT (In fact, the results just need to send to the nodes which has become a cluster head or will become a cluster head). In the next time, PC will keep calculating clustering result according to the collection of the global information and distributing the results to optimize the life of the entire network. The protocol also supports the extension network during the network lifetime. And the new joined nodes have the corresponding strategy to form the network organization too.

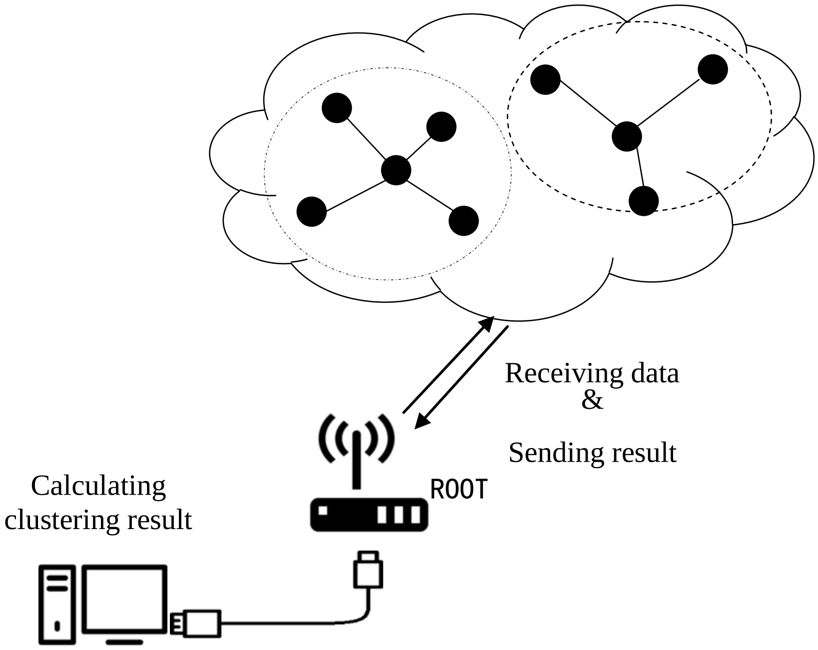


Figure 2. The Framework of our Solution

We make the following contributions to achieve the goals:

(1) We implement a reliable command-controlling distribution scheme based on CTP to ensure that ROOT could reliably send the controlling commands to any specific nodes in the network. Even though some distribution protocols can be used, considering some reasons (which has been explained in section 3.1), we decide to implement a data distribution method that adapts to our protocol better.

(2) We improve the process mode of neighbor table based on the basic CTP to obtain a handling mechanism which has a better design to meets the requirements.

(3) We also design a clustering mechanism to make the basic CTP routing mechanism to adapt to our solution. It can mainly choose a new routing approach according to the nodes information and the neighbor table information, etc.

(4) We implement a simple clustering algorithm on PC to generate the clustering solution of energetic balance by using the gathered information from all nodes in the network to achieve the goal of prolonging network lifetime. It is worth mentioning that clustering algorithm is independent of the network protocol. The network simply supports the clustering feature, and it can actually change the clustering algorithm to adapt to the application scenarios in any time without redeploying the network. This feature makes the whole system more flexible and widespread. In the paper, we implement an analogous LEACH-like [19] algorithm to test our proposed protocol.

3.1. The Command Dispatch

Because 2C-CTP is a centralized controlling network protocol, we need to make the network support the data distribution (the root can distribute the data to any node in the network. However, the CTP protocol does not support the data distribution, so we should explore a better solution of data distribution.

Although some distribution protocols have been used, such as Deluge [20], it's the standard code distribution protocol of Tinyos system. The protocol uses the ADV-REQ-DATA, three-way handshake and NACK (negatrue acknowledge) mechanism to ensure the reliability. The protocol proposes the paging mechanism making the network use the Pipelining to distribute the data in the space multi-channel. Thus the speed of the distribution has been significantly improved.

Moreover, the Typhoon [21] protocol uses the snoop mechanism to sniff the data. When the data is re-transmitted at high speed, the mechanism reduces collision probabilities of the data and uses the channel conversion technology to solve the problem of the hidden station, achieving the goal of spatially multi-channel transmission at high speed.

ECD [22] protocol proposes a selection algorithm of sending node based on the link quality. The algorithm chooses the node which has the best quality link as the sending node to improve the accuracy of sending node selection. Furthermore, the protocol dynamically configures the packet size to enhance the efficiency of code distribution.

Those three algorithms mentioned above are designed for the code distribution. They are meant to distribute the code blocks to the nodes better. The code blocks are always the big data, but 2C-CTP just needs to send the control signal. So we decide to implement our distribution protocol. To quickly response the network, the protocol firstly thinks about sending the control command to the target node by a speedy and reliable way instead of sending the big data block with high efficiency. Moreover, due to the capacity of limited processing and storage for the nodes, we hope our distribution protocol can be integrated with the other modules of the node to save the chip resource. According to these objectives, we try to utilize the CTP on our distribution protocol.

Here we improve the original CTP protocol. CTP is an address-less rendezvous protocol. It can only collect the data from every node in the network through the base station.

To implement the data distribution protocol based on CTP, we designed a table which called the "backtracking routing table". We mean to realize a command sending solution with lightweight and quick response through this table. The table is saved and maintained by the data from forwarding layer. The "backtracking routing table" record each forwarding data is forwarded from which node. The specific design follows as Table 1.

"Origin" records the original node ID of the data which is received from the forwarding layer, and "back" records the last hop's node ID of the received data. In the forwarding layer we use the following methods to maintain this form: The forwarding layer can receive the data from the child nodes which need to be adapted. When we can get the packet source node ID from the packet, viz the ORIGIN in the table, and the child node ID is BACK. These two ID respectively are saved in CTP protocol's packet header and transport layer's packet header. To get these two values will not increase the amount of data transmission.

Table 1. An Example of the Backtracking Routing Table

Origin	Back
12	5
23	13
41	6
4	3
…	…

When we need to send the data to a node in the network, the ROOT node only needs to query the carried table and sends the data to the previous target node. Hereafter, each node will be able to send the data to the destination node only by querying this table. As shown in Figure 3, the root node receives a command from PC which needs to be sent to the node12. Firstly, root queries the table and knows that the command is need to be forwarded to the node 5. Node 5 also queries the table and sends command to the node6, then node 6 sends the command to the destination node according to the same rules.

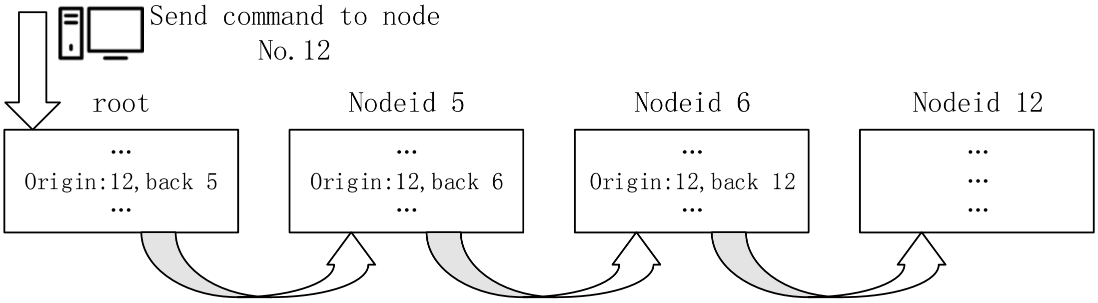


Figure 3. The Usage of Backtracking Routing Table

Different from the neighbor table, the updating of backtracking routing table is completely concerned with the normal data receiving and forwarding. Except for running few codes in the node, it hardly brings any performance loss to the node, and its maintenance is not like a neighbor table that is supported by the link estimation. We just utilize the formed route to avoid the performance burden brought by the routing recalculation.

The implementation not only makes the maintenance convenient but also has a good flexibility. Backtracking routing is changed with the collection routing. As long as a routing path can receive the data, we can send commands to the node on the routing path quickly and reliably.

3.2. The Design and Forwarding of the Control Packet

We use the method of forwarding engine in CTP to forward and execute the controlling command. When a node receives or generates a data, we just need to simply modify or fill the CTP protocol packet head and send the packet to the parent node. We decide to adopt a method. It not only avoids the misjudgment of routing loops but also adds the characteristic on control packets to try to make the signal packet carry the minimum data, reducing the control network energy consumption. We modify the packet head of CTP. If a node receives a data packet which the source value is 0 (root), the node responds that it's a signal control packet and then the node will change the ETX into MAX instead of processing the packet as the common CTP packet. So that we can achieve our goals without increasing the energy burden. Our control packet structure is shown as Figure 4.

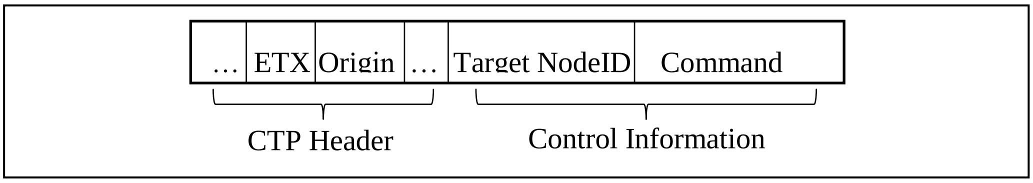


Figure 4. The Control Packet

When the forwarding engine receives a controlling command, it immediately detects whether the controlling command is a command which is sent to it or a broadcast command. If it is the situation mentioned above, the forwarding engine can process accordingly. If it is not, the forwarding engine will check the backtracking routing table and forward the command. The broadcast command we mentioned here is the command which is sent to all nodes on the link. In some cases, we need the whole network to execute the command. It will inevitably cause energy waste if we use the unicast form. So we introduced the broadcast command. Even if we send the command to the fringe node, the whole network will execute the command. We add a bit to the packet to denote that as broadcast command, viz all nodes on link should execute and forward the signal.

Algorithm 1. Receiving and forwarding packets

Input: Received Packet P, current NodeID

Output: N/A

```
function REC_AND_FORWARD_PKTS (P,NodeID)
    if P.origin_field == 0 then
        if P.TargetID == NodeID then
            DO_EXEC_COMMAND(P)
        else
            DO_QUERY_BACKTRACKINGTABLE(P)
            DO_FORWARD_PKT(P)
        end if
    else
        DO_FORWARD_PKT(P)
    end if
    return
```


13. **end function**

3.3. The Clustering Supported Routing Engine

In the CTP protocol, routing engine maintains the neighbor table according to the information provided by link layer, and processes some events when the routing engine needs to update the routing in some cases, such as the missing of parent node, the arrival of the routing updating time, etc. When executing the routing updating, routing engine firstly calculates the ETX value from every node itself in the neighbor table and adds the ETX value which is carried in the neighbor table. During this period, routing engine will ignore the neighbor nodes which cannot be connected. Then it will take the node with the smallest ETX value as the candidate parent node, and it will determine whether to really update routing according to the ETX updating threshold.

Algorithm 2.The Routing update theory in CTP

Input: N/A

Output: The best candidate NODE

1. **function** *DO_UPDATEROUTING*
2. **for all** Node' ∈ neighbor_Table **do**
3. ETX' ←*GETETX*(Node')+Node'.ETX
4. **if** ETX' < MinETX **then**
5. MinETX ←ETX'
6. NODE ←Node'
7. **end if**
8. **end for**
9. **if** CurrentETX −MinETX > 20 **then**
10. UPDATEROUTING(NODE)
11. **return** NODE
12. **end if**
13. **return** NULL
14. **end function**

The solution we proposed only has one type of control signal which can turn the node into cluster head. When a node receives a control signal whose target is the node itself, it will inevitably make itself become a cluster head or the normal node. In the implementation of clustering strategy, we make significant improvements on the origin CTP routing engine.

First is the routing layer processing of the cluster head nodes. We introduce "IsCluster" to denote whether the current node is the cluster head node. When a node receives the control signal which turns the node into cluster head, it will immediately set IsCluster value to 1 and it will be like a new node rejoining the network. The node will clear all the current routing information, including the parent node, neighbor table, etc. After the clean-up, the node which has become the cluster head will try to rejoin the network. Here we set cluster head nodes what we only choose the root or other cluster head nodes as its parent node. After rejoining the network successfully, the node will broadcast the beacon packet to notify that it has become a new cluster. The nodes which are able to receive the beacon packet will form a potential clustering region.

Conversely, if a cluster head node receives the control signal which turns the cluster head node into a normal node, it will immediately set "IsCluster" to 0, empty the routing information and be a normal node to join the network. During this period, the nodes in the cluster will capture this change and take corresponding strategies.

Algorithm 3.The Command Processing

Input: Command

Output: N/A

1. **function** PROC_COMMAND(Command)
2. **if** Command == beCluster **then**
3. **if** isCluster == 0 **then**
4. CLEAR_ALL_ROUTING_INFO()
5. RENETWORKING_AS_CLUSTER()
6. SEND_BEACON()
7. isCluster ←1
8. **end if**
9. **else**
10. **if** isCluster == 1 **then**
11. CLEAR_ALL_ROUTING_INFO()
12. RENETWORKING_AS_CTP()

```
            isCluster ←0
        end if
    end if
    return
end function
```

In order to adapt to our solution, we make a tiny change on original beacon packet of CTP, the original beacon packet format is shown as Figure 5.

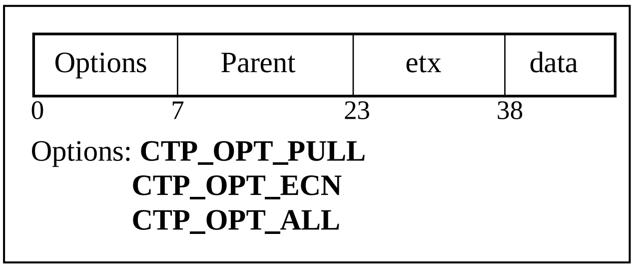


Figure 5. Original Beacon

We introduce the variable "clusterNum" to show how many cluster head nodes in the current node communication range. When the node just starts, the node will set the options as CTP_OPT_PULL beacon packet to broadcast. The nodes which receive the broadcast will send the beacon packets to this node. If there is no cluster head node around the node, the variables "clusterNum" value will be 0, then the node will construct the network according to the standard CTP protocol. That is choosing the node which has the smallest etx value as the parent node from the neighbor table which consists of previously received beacon packets. If a node has cluster around it, it will receive beacon packet from the clusters where the options is CTP_OPT_ISC. We do not use the way we mentioned earlier to deal with the neighbor table for this kind of beacon packet. We ignore its etx size and mandate to add the beacon packet to the neighbor table, (if it does not exist in the neighbor table), add 1 on clusterNum value. If it already exists in the neighbor table, then the table will refresh the information and decide whether it has changed the type according to the options. If it has been changed, clusternum value will increase or decrease according to the situation.

When a node's clusterNum value is changed from 0 to 1, it denotes that the node has been in a cluster network. At this time, the node needs to follow the new rules to join the network. Firstly it will remove the routing information (but not including the neighbor table), and then choose the cluster head node from the neighbor table as its parent. If there are multiple cluster head nodes, the node will chose the cluster head node which has the minimum etx value as its parent node.

Conversely, when a node's clusterNum value is changed from 1 to 0, it denotes that the node has been away from the cluster network. Here, the node also needs to remove the routing information and rejoin to the network. The node will follow the original CTP protocol.

Algorithm 4.The Beacon Processing

Input: Beacon From node X, Beacon

Output: N/A

```
function PROC_BEACON
    UPDATE_NEIGHBOUR_TBL(X)
    if Beacon.Options | CTP_OPT_ISC then
        clusterNum ← clusterNum+1
        if clusterNum == 1 then
            CLEAR_ALL_ROUTING_INFO()
            RENETWORKING_AS_2CCTP()
        end if
    else
        if X ∈ neighbor_Table then
            clusterNum ← clusterNum−1
            if clusterNum == 0 then
                CLEAR_ALL_ROUTING_INFO()
                RENETWORKING_AS_CTP()
            end if
        end if
    end if
end function
```

3.3.1. Clustering Strategy: Through the statement of the above chapters, we can sum up the clustering strategy in 2C-CTP:

The cluster head node can only choose the root node or other cluster heads as its parent node.

A normal node can only choose a cluster head node as its parent node, when the variable "clusterNum" is greater than 0.

A normal node executes standard rules of CTP, when the variable "clusterNum" equals 0.

4. Performance Evaluation

4.1. Experiment Setup

To evaluate the performance of 2C-CTP, we have performed real experiments. Twenty-five TelosB nodes are deployed in a real wireless sensor network environment, while a sink node is connected to a computer, which is used to receive data and control the nodes by sending commands. The packet payloads every node gathered contains data of node-ID, node-voltage and some junk data. The junk data is used to fill the payload to 28bytes, which is a size approaching to real application scenarios.

Figure 6. Sensors Used in Our Experiments

4.2. Experiment Setup

4.2.1. Controllability: To evaluate the effectiveness of the command sent from PC, we select a routing path from the formed network as shown in Figure 7. On the PC which connects to the sink, it generates 5 commands every 5 seconds and sends them to the nodes in path respectively, while the nodes were sending one packet (28bytes) per second. The function of the command was triggering LED on node. So the blinking of LEDs could express that the command should be effective or not. Table2 contains the theoretical number of commands which node should be received, and command execution times of nodes in the path. The duration of experiment was 1 hour.

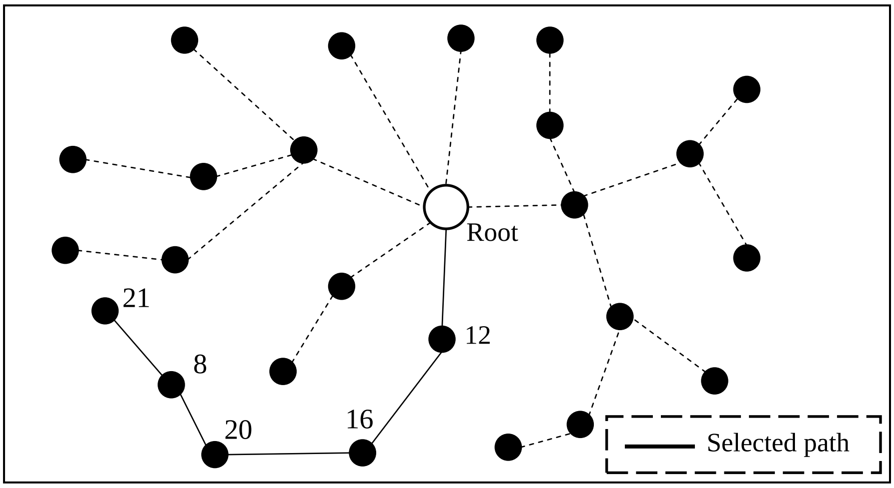


Figure 7. The Selected Path

Table 2. The Success Rates of Command Distribution (1)

Node id	theoretical	execution times	success rate
12	3600	3504	97.3%
16	3600	3497	97.1%
20	3600	3510	97.5%
8	3600	3423	95.1%
21	3600	3400	94.4%

The experimental result indicates that the commands could be received and execute responsibly under a large flow circumstance. However, with increasing hop count of the path, the success rate decreases slightly. When we only lower the packet sending frequency to 2 times per min, the result could be found in Table 3. The result shows that under a relatively smaller flow network, command could be executed more effectively.

Table 3. The Success Rates of Command Distribution (2)

Node id	theoretical	execution times	success rate
12	3600	3560	98.8%
16	3600	3557	98.8%
20	3600	3541	98.3%
8	3600	3523	97.8%
21	3600	3532	98.1%

4.2.2. Network-LifeTime and Data Collection: Another experiment has been carried out to analyze the clustering period impacts of data traffic and evaluate the energy consumption of 2C-CTP. In this experiment, a LEACH-like clustering algorithm was adopted on the PC. The clustering period was 5, 15, 30, 60 mins as the number of cluster head was 3. Every nodes generate 60 packets per min with two AA batteries. In this experiment. There was a total of 4 cases in this experiment, Table 4 presents the data records received from nodes in different situations adopted 2C-CTP, and Figure 8 presents the number of surviving node as time went on.

Table 4. The Numbers of Packet Received

Case	Clustering period	Sending frequency	Packet received
1	5mins	60/min	2106924
2	15mins	60/min	2108645
3	30mins	60/min	2507931
4	60mins	60/min	2366426
CTP	N/A	60/min	2367260

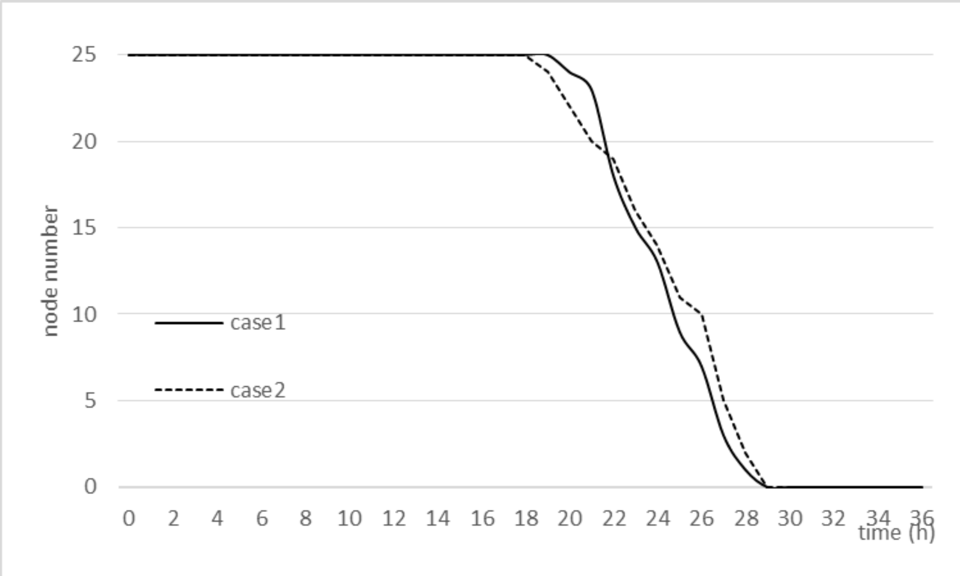


Figure 8. Lifetime of Network (1)

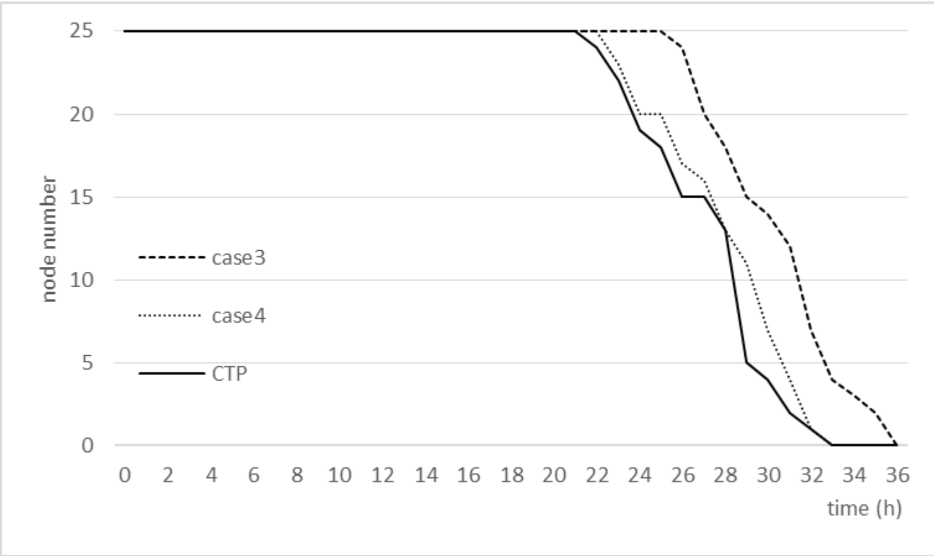


Figure 9. Lifetime of Network (2)

When traditional CTP is adopted in the experiment, the results are also shown in Figure 8, where all the nodes were given the same batteries as the previous experiment, and the sending frequency of the data is also 60/min. Comparing case1 and case2 with CTP, the results show that the lifetime of network decreases obviously with a short clustering period, as clustering frequently brings great overhead in both performance and power dissipation. However, an overlength period may fail to prolong the lifetime, like case4. In summary, to further extend the network lifetime furthest with 2C-CTP, a rational clustering should be adopted, like case3.

4.2.3. Energy Balance: As one of our goals is to enhance the lifetime of the network, the voltage histogram of each node at different times is shown in Figure 10 and 11, from which we can analyze the specific changes more clearly.

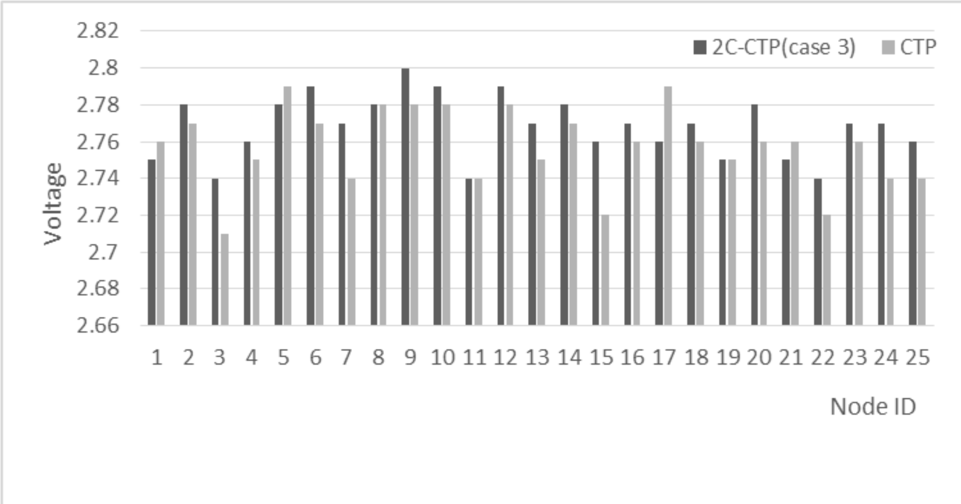


Figure 10. Voltages of every Nodes at 6h

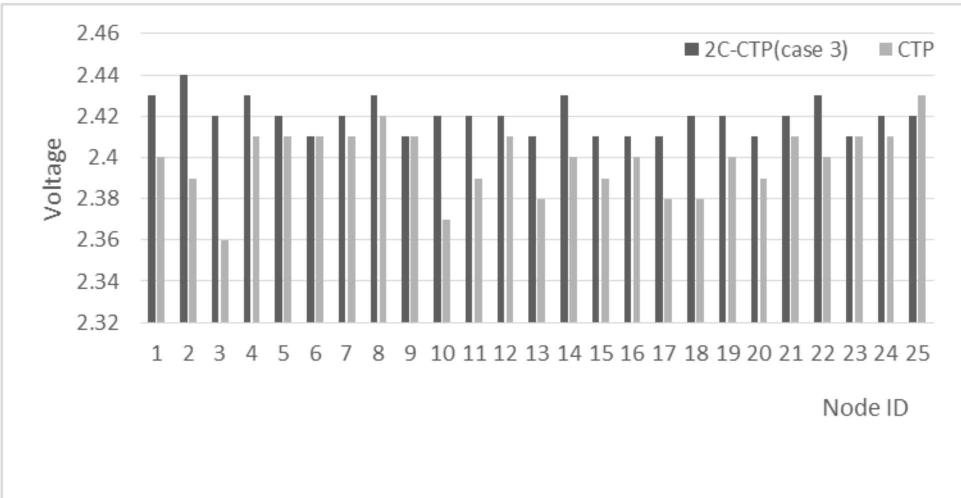


Figure 11. Voltages of every Nodes at 20h

From Figure 10 and 11, it could be easily summarized that the hotspots were avoided effectively in 2C-CTP. In CTP, the topology was relatively stable in the lifecycle of the network, the nodes forwarding many packets would bring premature death like Node3. Compared to other nodes, Node3 can only survive for 20 hours, 80 mins lower than the average lifetime in CTP, which could easily lead to route hole when deployed in large scale. The voltage comparison of Node3 in different protocols can be found in Figure 12.

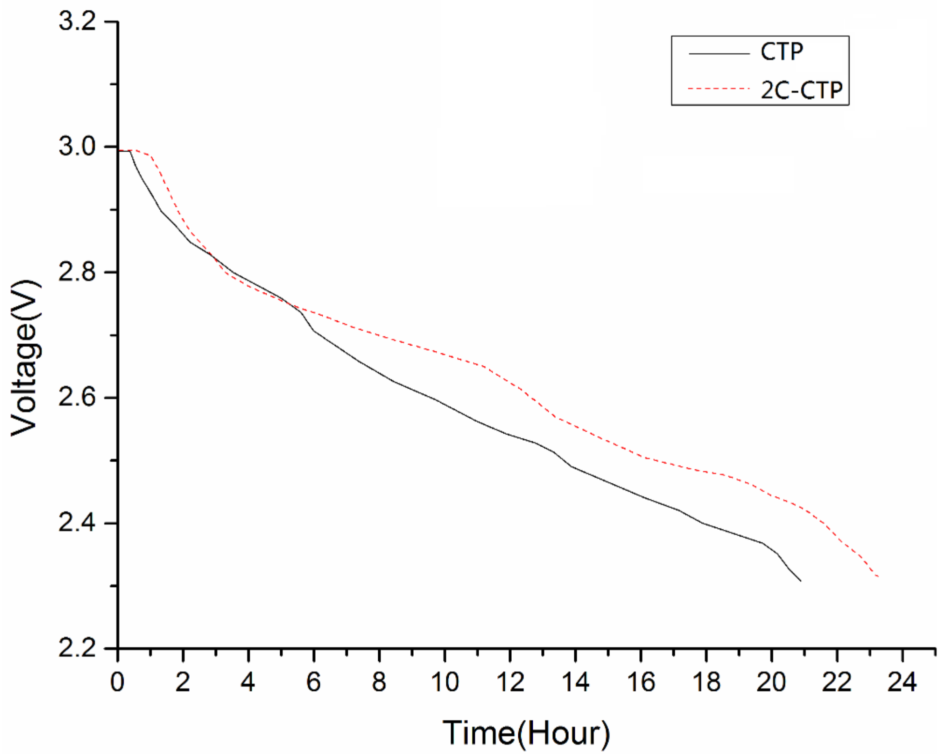


Figure 12. Comparison of Node 3

Once the clusters are changed regularly and the voltage of each node is taken into consideration in clustering algorithm, 2C-CTP would have the ability to prevent the excessive energy consumption of any node and keep the balance as far as possible.

5. Conclusions and Future Work

We propose a protocol based on CTP, supporting clustering and centrally being controlled for WSN that has been proposed. The experimental results show that 2C-CTP has high reliability of communication in both sending command and receiving data. Furthermore, if an appropriate clustering algorithm is adopted, 2C-CTP can balance the energy consumption. As a result, the network has a prolonged lifetime to avoid the route holes. In fact, the improvement of the network lifetime has reached at least 15.2%.

Considering that the experiment was a practice in a relative small scale with 30 nodes, 2C-CTP would play a greater role in energy balance and high-quality communication when it is deployed in a larger scale. In the future, we will focus on the clustering algorithm and control strategy in different application scenarios. For instance, if a node's sensor was out of order, the control system may let the node take on more tasks in forwarding to save other nodes' energy.

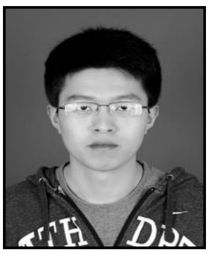
Acknowledgments

This work is supported by the NSFC(61232016, U1405254, 61173136,61173141, 61173142, 61373133,61502242), 201301030, 2013DFG12860, BC2013012, BY2013095-4-11 ,the CICAEET fund and the PAPD fund.

References

[1] P. Rawat, K.D. Singh, H. Chaouchi, "Wireless sensor networks: a survey on recent developments and potential synergies", J. Supercomput **(2014)**, Vol.68, pp.1-48.
[2] J. Yick, B. Mukherjee, D. Ghosal, "Wireless sensor network survey", Comput. Netw **(2008)**, Vol.52, pp.2292-2330.
[3] J. Shen, H. Tan, J. Wang, "A Novel Routing Protocol Providing Good Transmission Reliability in Underwater Sensor Networks", J. Internet. Technol **(2015)**, Vol.16, pp.171-178.
[4] P. Gou, J. Wang, X. H. Geng, "A Variable Threshold-Value Authentication Architecture for Wireless Mesh Networks", J. Internet. Technol **(2014)**, Vol.15, pp.929-935.
[5] X. M. Sun, S. S. Yan, B. W. Wang, L. Xia, Q. Liu, H. Zhang, "Air Temperature Error Correction Based on Solar Radiation in an Economical MeteorologicalWireless Sensor Network", Sensors **(2015)**, Vol.15, pp.18114-18139.
[6] Y. Tang, M. T. Zhou, X. Zhang, "Overview of routing protocols in wireless sensor networks", J. Softw **(2006)**, Vol.17, pp.410-421.
[7] A. Bachir, M. Dohler, "MAC essentials for wireless sensor networks", IEEE. Commun. Surv. Tut **(2010)**, Vol.12, pp.222-248.
[8] U. Hunkeler, C. Lombriser, H. L. Truong, "A case for centrally controlled wireless sensor networks", Comput. Netw **(2013)**, Vol.57, pp.1425-1442.
[9] O. Gnawali, R. Fonseca, K. Jamieson, "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks", ACM. T. Sensor. Network **(2013)**, Vol.10, pp.16.
[10] O. Gnawali, R. Fonseca, K. Jamieson, "Collection tree protocol", Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, **(2009)** November 4-6; NY, USA, pp.1-14.
[11] P. Levis, S. Madden, J. Polastre, "TinyOS: An operating system for sensor networks", J. Amb. Intel **(2004)**, Vol.35.
[12] Y. Liu, X. Mao, Y. He, "CitySee: Not only a wireless sensor network", IEEE. Network **(2013)**, Vol.27, pp.42-47.
[13] Available online: http://powernet.stanford.edu/ (Accessed on 15 April 2009).
[14] L. Ya, W. Pengjun, L. Rong, "Reliable energy-aware routing protocol for heterogeneous WSN based on beaconing", Proceedings of the 2014 International Advanced Communication Technology (ICACT), **(2014)** February 16-19; Pyeongchang, ROK, pp.109-112.
[15] Available online: http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM (Accessed on September 2013).
[16] J. Zhao, L. Wang, W. Yue, "Load migrating for the hot spots in wireless sensor networks using CTP", Proceedings of the 2011 Seventh International Mobile Ad-hoc and Sensor Networks (MSN), **(2011)** December 16-18; Beijing, China, pp.167-173.
[17] Y. Li, H. Chen, R. He, "ICTP: An improved data collection protocol based On CTP", Proceedings of the 2010 International Conference Wireless Communications and Signal Processing (WCSP), **(2010)** October 21-23; Suzhou, China, pp.1-5.
[18] J. Flathagen, E. Larsen, P. E. Engelstad, "O-ctp: hybrid opportunistic collection tree protocol for wireless sensor networks", Proceedings of the 2012 IEEE 37th Conference Local Computer Networks Workshops (LCN Workshops), **(2012)** October 22-25; Clearwater, FL, pp.943-951.
[19] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", Proceedings of the 33rd Annual Hawaii International System Sciences, **(2000)** January 4-7; IEEE.
[20] J. W. Hui, D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale", Proceedings of the 2nd international Embedded networked sensor systems, **(2004)** November; NY, USA, pp.81-94.
[21] C. J. M. Liang, R. Musăloiu-e, "Terzis, A. Typhoon: A reliable data dissemination protocol for wireless sensor networks", Proceedings of the 5th European Wireless Sensor Networks, **(2008)** January 30-February 1; Bologna, Italy, pp.268-285.
[22] W. Dong, Y. Liu, C. Wang, "Link quality aware code dissemination in wireless sensor networks", Proceedings of the 2011 19th IEEE International Network Protocols (ICNP), **(2011)** October 17-20; Vancouver, BC, pp.89-98.

Authors



Lingyun Xu, He received his B.S. degree in Information Science and Technology from Nanjing University of Information Science and Technology, China in 2013. Currently he is studying for his M.S. degree in Software Engineering at the same university. His research interests include wireless networks and network routing.

Tao Li, He received his M.S. degree in Computer Application from Nanjing University of Technology in 2004, and his Ph.D. in Signal and Information Processing from Southeast University. He is currently working as a lecturer in School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include wireless sensor network and embedded system.

Baowei Wang, He received his B.S. and Ph.D. degrees in Computer Science from Hunan University in 2005 and 2011, respectively. He is currently working as a lecturer in School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include steganography, wireless networks and securing ad hoc networks.

Xingming Sun, He is a professor in the School of Computer and Software, Nanjing University of Information Science and Technology, China from 2011. He received the B.S. degree in Mathematical Science from Hunan Normal University and M.S. degree in Mathematical Science from Dalian University of Technology in 1984 and 1988, respectively. Then, he received the Ph.D. degree in Computer Engineering from Fudan University in 2001. His research interests include information security, network security, cryptography and ubiquitous computing security.

Xingang You. He is a professor from Nanjing University of Information Science & Technology, China. He received his Bachelor degree in Radio Engineering from Southeast University in 1984. Besides, he is also one of the directors of Chinese Computer Institute. His research area mainly covers steganography and Digital Watermarking technology signal processing and transmission technology and multimedia information security technology. He is now working as a doctoral supervisor in Beijing Institute of Electronic Technology. During his working time, he have undertaken many significant scientific researches, e.g. the national 863 project, and attained a lot of achievements. Plus, he also participated in the national 973 project and some natural science fund projects. In terms of honors, he attained multiple scientific researching achievements and was granted the national technical invention award and the scientific progress prize.

Yanhua Zhu, She received her B.A. degree in Business English from Binjiang College, China in 2015. Currently she is studying for her M.A. degree in Master of Translation and Interpreting at Nanjing University of Information Science and Technology.