

The research on a Multipath Forwarding Strategy in Wireless Sensor Networks

Fan Jingjing, Zhao Jianguang*, Liu xiaoqun

Information Engineering Institute, Hebei Institute of Architectural and Civil Engineering, Zhangjiakou
075000, China

*Corresponding Author.

Abstract:

Wireless communication signals are unstable, WSN network topology changes frequently, network dynamics are strong, and single path forwarding is difficult to provide high quality of service. A vector address-based end-to-end multi-path forwarding mechanism and path recovery algorithm are proposed. Although a single path cannot continuously guarantee network connection, multiple paths together can provide network connection with high reliability. In this method, the caller applies for and stores multiple relatively independent communication paths, and ranks them for use according to the path cost. The experimental results show that the performance of the above method is significantly improved.

Keywords: Software-defined wireless sensor network; Virtual wireless sensor network; multipath forwarding.

I. INTRODUCTION

In recent years, wireless sensor network (Wireless Sensor Network, WSN), as the core technology of the Internet of Things, has been widely used in aviation, military, counter-terrorism, disaster relief, environment, industry, medical treatment, and home furnishing. However, most WSN private networks are dedicated and universal. The performance is poor, and there is much room for improvement in WSN architecture, efficiency, and energy consumption [1].

The constructed network is mostly used to complete a specific task. If another task needs to be completed, a new network needs to be re-deployed, resulting in a low reuse rate of the physical equipment at the bottom of the network, a waste of WSN resources, and poor network versatility and multitasking. On the other hand, for a single WSN node, energy supply, computing and

storage resources are limited, and it is necessary to increase the service life of the node and reduce the power consumption of the node. Due to the resource limitations of a single WSN node, the network must balance the load to extend the life of a single node, thereby extending the life of the entire network[2-5]. In this paper, software-defined network (SDN) and vector network (Vector Network, VN) related technologies are introduced into the WSN field. Based on the vector network architecture, an SDN-based WSN network (VN-based WSN, VN-WSN) is designed.) Architecture, separating the data plane, control plane and management plane. A VN-WSN construction algorithm based on SDN and VN technology and a source-end data forwarding strategy based on vector forwarding are proposed, and the constructed model is analyzed, and the proposed algorithm is simulated[6-8].

II. ALGORITHM MODEL

2.1 Vector address

This section introduces vector network related technologies, including vector address, vector forwarding, vector network control plane and data plane architecture, and proposes a network model, and conducts an in-depth analysis of the model's data plane and control plane.

After the network devices in the network form a network, it is necessary to establish a set of encoding methods to achieve the purpose of communication. Each network device is assigned an encoding identifier. This type of identifier is called the identifier address of the network device. The vector address encoding diagram is shown in Fig.1 Shown.

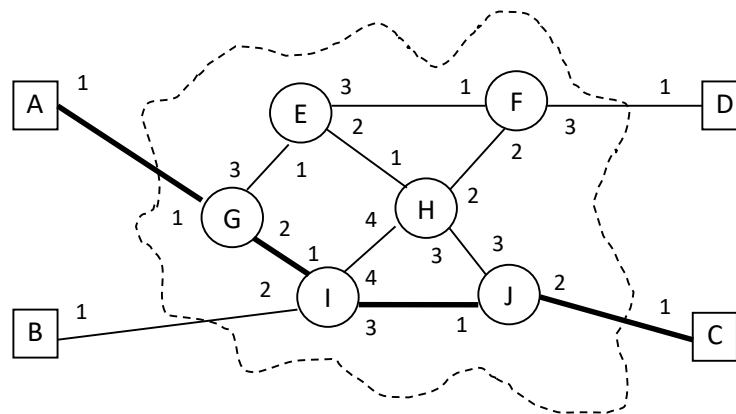


Fig. 1 The Address Encoding Method of VN

The box and circle both represent network nodes, the box node represents the sender or receiver of information, called terminal equipment, and the circle node represents the data forwarding node. The dotted line represents the boundary of the network. Inside the boundary is the forwarding node, and outside the boundary is the terminal node. In the figure, the network

nodes are respectively identified as A, B, C, D, E, F, G, H, I, J, each network The port of the node is also marked with a port name and marked with Arabic numerals. For example, the four ports of H are marked as ports 1, 2, 3, and 4.

According to the definition of vector address, the path from the source node to the destination node can be identified by a vector address. For example, the thick-lined path in the figure is A->G->I->J->C, the source node is A, and the destination The node is C, Data packets are sent out through port 1 of the source node, flow in through port 1 of G node, first flow out through port 2 of G node, then flow in through port 1 of I node and out through port 3, and then flow into port 1 of node J, and It flows out from port 2 of the J node, and finally flows into the destination node from port 1. Finally reaches the destination node C, the port sequence that is the vector address is {1,2,3,2 } Is the vector address from the source node A to the destination node C. These serial numbers are like direction signs, gradually guiding the data packet to the destination node. With this address, there is a path from node A to node C. The data packet Can be forwarded from A to C.

Compared with the address in the traditional network, the vector address is more direct. The IP address is the device code, the ATM address is the link code, and the vector address is the port code. The most direct channel for data forwarding is the port. Therefore, the vector address is better than the IP address and ATM address. Vector addresses have infinite versatility, address indefinite length, uninterpretability, encryption, address relativity and path definition.

2.2 Vector forwarding

According to the vector address, the process of guiding the data packet to be transmitted step by step from the source node to the destination node is called vector forwarding. In the forwarding process, data packets receive data packets from the input port and send data packets from the output port. Each time the input port receives a data packet, the forwarding algorithm is called. The basic operations of vector forwarding are as follows:

- (1) Receive data packets;
- (2) Separate the first component V_i of V_A in the received data packet, and assign V_i to the mark T_o ;
- (3) Delete the current component address V_i of V_A in the data packet;
- (4) Send the data packet to the output port T_o .

The vector forwarding flowchart is shown in Figure 2.

Assuming that the vector address is in binary form, in the example shown in Figure 3, there is a data packet sent from node A, and the sent data can be expressed as follows:

[Head 11001110 Data]

The forwarding process is shown in TABLE 1.

The data requests the path address from the control plane. After the controller receives the address request, it feeds back the path information to the source node, and the source node obtains the path information from the source node to the destination node. The path information is

essentially a sequence set of forwarding ports. And append the path information to the data packet. The forwarding node realizes the forwarding of data packets from the source node to the destination node by analyzing the port information of the node [9-12].

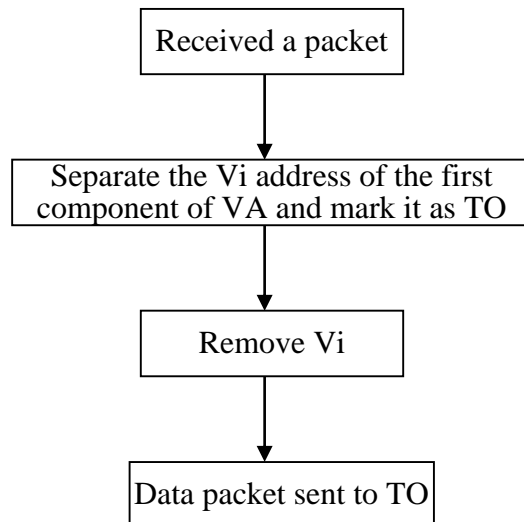


Fig.2 The Basic Process of Vector Forwarding

The process of sending data packets from source node A to destination node B is as follows:

Step 1: Node A requests to send data to B

Step 2: A requests the path from the controller

Step 3: The controller sends the vector address (the path to B) to node A

Step 4: Choose the path

Step5: Vector forwarding

Step6: Check whether it reaches B correctly and perform the corresponding operation.

TABLE I. The Vector Forwarding Process of VN

Packet before analysis	Path node	Separate component address	Forwarding port	Packet after separation
[Head 11001110 Data]	A	1	1	[Head 1001110 Data]
[Head 1001110 Data]	G	10	2	[Head 01110 Data]

[Head 01110 Data]	I	011	3	[Head 10 Data]
[Head 10 Data]	J	10	2	[Head {} Data]

III. EXPERIMENTAL SIMULATION AND ANALYSIS

When selecting the best path, in addition to considering the number of hops and buffer occupancy, the load and bandwidth of the nodes in the path must also be considered. In this experiment, multi-path forwarding first selects the path with the smallest number of hops for data forwarding, and in the case of the same number of hops, preferentially selects the path with the least congestion. In order to select the path with the least congestion, it is necessary to comprehensively consider the number of hops and the buffer load of each node on the path, as shown in equation (1).

$$\min \left\{ \frac{1}{n_j} \sum_{i=1}^{n_j} (buffer\ size(i)) \right\} \quad (1)$$

Among them, n_j is the route hop count of path j , and $buffer\ size(i)$ is the occupied size of the buffer on path i . By measuring the buffer occupancy of all paths and the number of hops of each path, the optimal path is selected.

The optimal path bandwidth is calculated by formula (2), as shown in formula (2).

$$BW_a = BW_{max} \times (T_{idle}/Int_t) \quad (2)$$

Among them, $[BW]_a$ is the available bandwidth, $[BW]_{max}$ is the maximum bandwidth of the path, T_{idle} is the time interval of channel idle time, and $[Int]_t$ is the total channel communication time.

The optimal path load calculation in the experiment is measured by the average channel competition of a single node, and the average channel competition of a single node in the network is measured by the contention window (CW) and queue length (QL). The calculations of CW and QL adopt formula (3) and formula (4) respectively.

$$CW = \alpha \cdot CW_{old} + (1 - \alpha) \times CW_{cur} \quad (3)$$

$$QL = \beta \cdot QL_{old} + (1 - \beta) \times QL_{cur} \quad (4)$$

Among them, CW_{old} and CW_{cur} respectively represent the CW value at the previous time and the current time CW value, QL_{old} and QL_{cur} respectively represent the queue length at the previous time and the current queue length, $\beta \in [0,1]$

Through formula (3) and formula (4), the load information of node n in the network can be

obtained, as shown in formula (5).

$$L_n = \gamma \times CW / CW_{max} + (1 - \gamma) \times QL / QL_{max} \quad (5)$$

Considering the characteristics of this algorithm, the value of γ is selected as 0.45 in the experiment.

The vector network multi-path forwarding algorithm includes the data plane algorithm as shown in Algorithm 1, and the control plane algorithm as shown in Algorithm 2. The detailed implementation is as follows pseudo code:

Algorithm 1 Multi-path vector connection data plane construction algorithm:

```
1  while(1)
2  for i = 1:N
3  If(Node.DatatoSent)
4  this=pointer to current node
5  this.sent(VA_REQUEST)
6  VA_REQUEST_TTL =VA_REQUEST_TTL-1
7  If(this.receive)
8  VA[]=this.recurve()
9  Store VA[]
10 If(VA_REQUEST_TTL=0)
11 goto 3
12 for i = 1:i
13 VA=VA[i]
14 this.sent Hello packet byVA[i]
15 If(this.receive)
16 Insert VA[i] into VAs[ ]
17 End
```

In the experiment, the delay index (Dalay, D for short) adopts the end-to-end average delay, which refers to the arithmetic mean of the time difference between the receiving node receiving data packets and the sending node sending data packets, as shown in equation (6).

$$D = \frac{1}{i} \sum_{n=0}^i (Rt_n - St_n) \quad (6)$$

In the formula, i represents the total number of data packets successfully transmitted by the network, Rt_n represents the time to receive the n th data packet, and St_n represents the time to send the n th data packet.

The calculation of routing cost (Routing Cost, RC) is shown in formula (7).

$$RC = (SC_i + FC_i)/RD_i \quad (7)$$

In the formula, SC_i represents the total number of control signaling sent by the network, FC_i represents the total number of control signaling forwarded by the network, and RD_i represents the total number of data packets received by the network.

The control surface algorithm is shown in Algorithm 2.

Algorithm 2 Multi-path vector connection control surface construction algorithm:

```
1   while(1)
2     for i = 1:M
3       while(Control.Listen)
4         this=pointer to current control
5         VA[]=this.creatVA()
6         this.sent(VA_Reply)
7         VA_REQUEST_TTL =VA_REQUEST_TTL-1
8         If(this.receive)
9           END
10        If(VA_REQUEST_TTL=0)
11          goto 6
```

Figure 3 compares the delay characteristics of the three algorithms. Through experiments, it can be seen that the VN delay is relatively high in the initial stage, and then gradually lower than the other two algorithms, mainly because the network node needs to request a path from the controller. As the path is established, the subsequent delay gradually decreases.

Among them, FLOODING's total energy consumption is 377nj, GOSSIP's total energy consumption is 76.1nj, and VN's total energy consumption is 40.9nj. VN is 366.1nj lower than FLOODING's total energy consumption, and 35.2nj lower than GOSSIP's total energy consumption, respectively 89.15% and 46.25%.

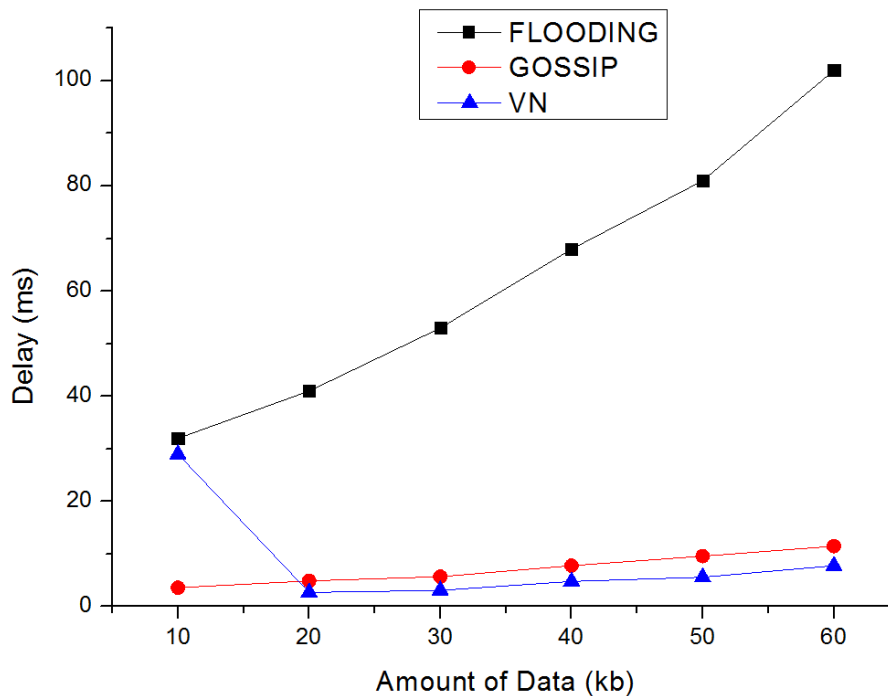


Fig.3 Transmission Delay vs. Data Volume Curve

IV. CONCLUSION

FLOODING is generally used for topology establishment and update, not for data forwarding. However, GOSSIP is a commonly used routing mechanism in sensor networks, which has certain reference significance when compared with VN data forwarding. In applications where WSN nodes only need data forwarding, VN data forwarding has obvious advantages. As the amount of data to be transmitted increases, the advantages of vector forwarding become more obvious.

ACKNOWLEDGEMENTS: This work was financially supported by Zhangjiakou City's 2018 municipal-level science and technology plan financially-funded project. Project name: Research on Cloud Computing Data Center Network Architecture Optimization of Olympic Stadiums. Project Number: 1811009B-19, and the "Three, Three, Three Talents Project" of Hebei Province in 2018, the project name: Software-Defined Wireless Sensor Network Data Forwarding Research Project award number: A201803005. Hebei Technological Innovation Guidance Plan Project: Project Title: Research on High-precision Location Technology of Snow and Ice Accidents in VR Scene Based on 5G. Project Number: 20470302D.

REFERENCES

- [1] Jiang Zhongyuan, Liang Mangui, Zhang Shuai, Wang Shujuan, Guo Dongchao. AN EFFICIENT BANDWIDTH ALLOCATION STRATEGY FOR SCALE-FREE NETWORKS[J]. International Journal of Modern Physics C, 2012-1:440-450.

- [2]Jiang Zhongyuan, Liang Mangui, Guo Dongchao.IMPROVED EFFICIENT ROUTING STRATEGY ON SCALE-FREE NETWORKS[J].International Journal of Modern Physics C,2012-2:1250016-1250040.
- [3]Jinxin Zhang,Mangui Liang.A Credibility-based Congestion Control Scheme and its Performance Evaluation[J].International Journal of Computational Intelligence Systems,2012-9:851-859.
- [4]Jiang Zhongyuan, Liang Mangui, Huang Jianling, Li Gan. AN EFFICIENT WEIGHTED ROUTING STRATEGY FOR SCALE-FREE NETWORKS[J].Modern Physics Letters B,2012-11:440-448.
- [5]Zhao Aqun, Liang Mangui, Lian Songhai, Guo Xiaoming. Research on the Average Length of Vector Addresses[J]. High-Tech News, 2011-12:89-95
- [6]Zhao Aqun, Liang Mangui. A new forwarding address for next generation networks[J].Journal of Zhejiang University Science C,2012-1:1-10.Lucian Popa,Praveen Yalagandula,Sujata Banerjee,Jeffrey C. Mogul,Yoshio Turner,Jose Renato Santos.Elastic Switch[J]. ACM SIGCOMM Computer Communication Review . 2013 ,43(4):351-362.
- [7]Seung Yeob Nam,Sirojiddin Djuraev,Minho Park.Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks[J]. Computer Networks,2013,57(18):3866-3884.
- [8]Zhang Chaokun, Cui Yong, Tang Zhuo, Wu Jianping. Research progress in software-defined networking (SDN) [J]. Journal of Software, 2015(01): 62-81.
- [9]Huang Yun. Consumption tax management innovation in the Internet of Things era: international experience and application prospects [J]. International Taxation, 2014 (09): 60-63
- [10]Huang Tao, Liu Jiang, Huo Ru, Wei Liang, Liu Yunjie. Overview of research on future network architecture [J]. Journal of Communications, 2014(08): 184-197.
- [11]Wang Wendong, Qi Qinglei, Gong Xiangyang, Hu Yannan, Que Xirong. An autonomous service quality management mechanism based on software-defined networks (English) [J]. China Communications, 2014(07):1-7.
- [12]Qiao Xiuquan, Zhang Yang, Wu Budan, Cheng Bo, Zhao Shuai, Ma Huadong, Chen Junliang. Event-driven, service-oriented Internet of Things service provision method[J]. Chinese Science: Information Science, 2013(10):1219- 1243.
- [13]Hu Guangchang, Wu Budan, Chen Junliang. Business process adaptation based on context awareness and rule orientation [J]. Computer Integrated Manufacturing System, 2013(08): 1898-1905.
- [14]Zhang Shunmiao, Zou Fumin. Overview of Software-Defined Network Research[J]. Application Research of Computers, 2013(08): 2246-2251.
- [15]Lin Chuang, Jia Zixiao, Meng Kun. Adaptive future network architecture [J]. Chinese Journal of Computers, 2012(06):1077-1093.