The Test of Contiki-based Protocol Stack of 6LoWPAN

Liu GuangDi *
Library of Chengdu University
Chengdu, China
E-mail: liuguangdi1103@126.com
* Corresponding Author

Tie Ling
School of Information Science and Technology
Chengdu, China
E-mail: 2396565376@qq.com

Abstract—Propose a 6LoWPAN protocol stack test scheme based on the Contiki operating system [1]. Use the simulation tool cooja to test the performance of 6LoWPAN protocol [2]. Analyze the network connectivity, energy consumption, timeline when the header compression algorithm, packet fragmentation and reconstruction were introduced into the scheme.

Keywords-Component; 6LoWPAN Protocol Stack; Contiki Operate Systerm; Cooja; Wireless Sensor Networks; IPv6 Stack Protocol;

I. WIRELESS SENSOR NETWORK BASED ON 6LoWPAN

This article performs the test of 6LoWPAN protocol stack based on contiki system. The simulation results show that the protocol stack can achieve normal transmission of the packet in wireless sensor networks.

II. THE SIMULATION OF THE 6LOWPAN STACK

6LoWPAN technology is actually an IPv6 protocol stack of wireless sensor network. It constructs an adaptation layer to compress header. IPv6Dispatch field is added to IPv6 header. The compress_hdr_ipv6 () function is main processing function [4].

1280 bytes are minimum MTU in IPv6 link layer; however, the link layer provides only 127 bytes in IEEE802.15.4 standard, so adaptation layer must be added into 6LoWPAN stack. It uses runtime packet fragmentation and reassembly to solve the problem of large packet transmitted.

III. THE SIMULATION PLATFORM OF 6LOWPAN

A. Cooja Simulation Platform

Cooja is the simulator in the Contiki operation system. It supports cross-layer simulation based on instruction level, system level and application level. In this paper, the test environment topology is shown in Fig. 1. The completed 6LoWPAN protocol stack is installed in each node. The node 1 and 2 communicate with node 3. The communication mode is based on Client/server. udp-client.c program is running in the client; udp-server.c program is running in the server [5].

B. Packet fragmentation and reassembly

The data packets transmitted among the nodes 1, 2 and node 3 may be larger than the maximum transmission unit. The fragmentation will be built in the sender and reconstruction is performed in the received part. The test result is shown in Fig. 2.

C. Energy Consumption Test

The power consumption results are shown in Fig. 4-7. The highest level of power consumption is highlighted based on Power Tracker [6].

Fig. 3-6 shows that the energy consumption of node 2 is the highest each time. The answer can be found in Fig. 1, since the distance between the node 2 and node 3 is the longest, it requires higher power consumption to transmit...
the data packets. Further we know that the energy consumption is high when the node has just started, the energy consumption decreased when the connection is established.

![Figure 3. Node Energy State When Time= 5.446s](image)

![Figure 4. Node Energy State When Time= 12.385s](image)

![Figure 5. Node Energy State When Time= 24.412s](image)

![Figure 6. Node Energy State When Time= 41.316s](image)

### D. TimeLine Test

Cooja TimeLine shows a time line for each node in the simulation in Fig. 7. On the time line, the power state of the radio transceiver of each node is shown in a color code: It is white if the transceiver is off, gray if it is on. Radio transmissions and receptions are shown in the same time line: transmissions are blue and receptions are green. When two simultaneous transmissions are reaching to a node, it results in radio interference, red is shown. Transmissions and receptions are shown with bit granularity. See Fig. 7 for a Cooja screenshot with a TimeLine. [7]

![Figure 7. Timeline When Node Just Started](image)

### E. UDP Packet Transmission Test

Experiments are performed to show the UDP datagram transfer with the IPv6 address [8], in Fig. 8. Circular area represents the coverage of node 3 network. Nodes 1 and 2 are clients, node 3 is server. After the two nodes establish socket connection, they continuous send "Hello" string to node 3 by UDP protocol. Test results show that the wireless sensor network node can transfer the UDP datagram among them using the 6LoWPAN protocol stack. Fig. 8 shows the process that 3 nodes send message to each other and the process of the acknowledgment are covered by the node 1, 2 and node 3.

![Figure 8. Process of Sending and Receiving Data](image)

![Figure 9. UDP Packet Transmission Test.](image)
IV. CONCLUSIONS

This paper simulates the basic functions of 6LoWPAN protocol stack. [10] We test functions of datagram fragmentation and reassembly, energy consumption, timeline and UDP data transmission by cooja tools, and so on. Experimental results show that the proposed scheme is able to complete the normal process of networking and accurate transmission of data, with some research and application value.

ACKNOWLEDGMENT

This research work is supported by the Chengdu University foundation project (No.2012XJZ07); Sichuan Province Science and technology support program (No.2013GZ0016); the open foundation of Key Laboratory about Pattern recognition and intelligent information processing.

REFERENCES