

Improvements on Proxying and Cache Scheme of CoAP Protocol for IPv6-Based Wireless Sensor Networks

Ping Wang^{1,a*}, Xin Li^{1,b}, Heng Wang^{1,c} and Na Bao^{1,d}

¹Key Laboratory of Industrial Internet of Things and Networked Control, Chongqing University of Posts and Telecommunications, Chongqing, China

^awangping@cqupt.edu.cn, ^bleeandxin187@163.com, ^cwangheng@cqupt.edu.cn, ^d172105839@qq.com

Keywords: CoAP, WSN, proxying and cache scheme.

Abstract. For the IPv6-based Wireless Sensor Networks (WSN), there exists a difficulty in applying Internet protocol into Internet of Things (IoT) due to the constrained resources. Constrained Application Protocol (CoAP) is a lightweight protocol which is proposed and standardized by Internet Engineering Task Force (IETF). And it is designed for constrained environments, such as sensor nodes and sensor networks. However, there are some shortcomings for the CoAP proxying and cache scheme, e.g., without considering the topology changes adequately, and requiring a large cache space at proxy endpoints, etc. In order to address these issues, some improvement schemes on proxying and cache are proposed in this paper. The results verify that, through reducing the router throughput and time delay, the enhanced proxying and cache scheme can improve the network performance effectively.

Introduction

In recent years, Internet of Things (IoT) as one of the strategic emerging industries of China has undergone a rapid development. And Wireless Sensor Network (WSN) has attracted a lot of attention as the key technology of IoT perceptual layer. WSN possesses the ability to collaborative monitoring and sensing through ad-hoc deployment of numerous tiny and intelligent wireless network sensor nodes [1,2].

Due to the 32-bit address of IPv4 has been exhausted [3], IPv6 becomes the next generation Internet protocol with the advantage of vast amount of IP address, variety kinds of functions and services, as well as a higher safety coefficient. Since combining WSN and IPv6 technology, the research on IPv6-based WSN shares more and more concern [4].

For meeting the demand of Representational State Transfer (REST) web services architecture in constrained environments (e.g., WSN), Internet Engineering Task Force (IETF) proposed the Constrained Application Protocol (CoAP). Similar to HTTP, CoAP supports several key items of Web services, such as method definitions, code description for responses and Internet media types, etc. Therefore, CoAP is born to easily translate to HTTP with the Web. User can access the abstract resources of the server through explorer and CoAP public client. By parsing the Uniform Resource Identifier (URI) carried by request packets, the server can locate the target resource easily [5,6].

Proxying and Cache Mechanism

In WSN, nodes are mostly powered by battery, and dormancy scheme decreases much cost and security risks on the battery replacement. However, the client, which is the originating endpoint of a request, may get no response when the resource server is sleeping. Thus, CoAP use proxying and cache mechanism to improve this issue.

Proxying in CoAP are divided into forward proxying and backward proxying [6]. The CoAP client usually selects forward-proxies itself and entitles them the authority of clients. The backward proxying is set in WSN and stands for servers. Their message interactions are shown in Fig. 1 and Fig. 2. And the freshness model [7] is as follows:

$$\text{Proxy_max_age} = \text{Original_max_age} - \text{Cache_age} \quad (1)$$

It is used for judging the reliability of storage response with “freshness”, which is up to the response storage time in cache after data update. If the storage time is no more than maximum life cycle, response is determined as fresh and reliable, otherwise, response is considered not fresh.

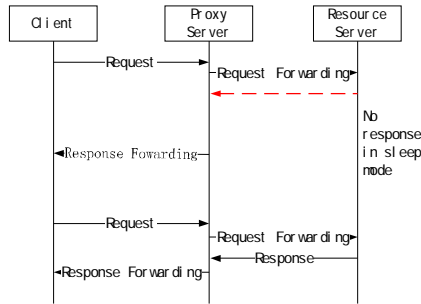


Fig.1 Forward proxying

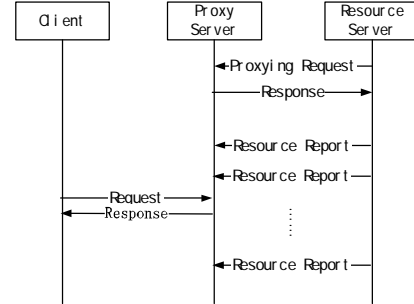


Fig.2 Backward proxying

In practical scene, the proxying and cache mechanism of CoAP has some shortcomings. Firstly, taking little consideration of the topology changes, lacking of maintain scheme may lead to the increasment of router throughput and time delay after network topology changes. Secondly, CoAP protocol has not given fully consideration to the time that a user costs to operate explorer, which may lead to complicate operations when access directly to sleepy servers. Thirdly, it may lead to more cache consumption to show the resource variation tendency.

Optimization

Proxying Mechanism Optimization. To reduce router throughput and shorten packet round-trip time, proxying are optimized to let node keep proxying relation with the optimal parent node. The trigger conditions can be characterized as follows:

1) After router maintenance, the optimal parent node of resource server is changed, and the proxy must be maintained to ensure a reasonable throughput and delay.

2) After sending the update message, if the response from proxy server shows that there is no storage space in resource cache library for resources, the proxying relationship must be maintained.

Once the condition occurs, a maintenance process will start. Before the proxying relationship maintaining, the resource server unbind the proxy server bound previously, then, it resends request to the optimal parent node to establish proxying relation. Proxying is built while a response codenamed “2.01 Created” is received, and it will fail with a “4.01 Unauthorized” response. Before receives the response, the control function will keep nodes out of sleep mode. Resource server will send constantly request until proxying establishes successfully or runs out of attempts.

Cache Mechanism Optimization. By timely deletion of the space, where involves error proxy and unbind proxy, user can conveniently manipulate resource in sleepy server. Following the deletion, the resource cache library starts a maintenance. The trigger conditions are as follows:

1) When proxy server can not locate the machine resource storage area after receiving update request, resource cache library maintenance is needed, proxy server responded to build proxy again.

2) When proxy server can not create storage space after receiving request due to the full cache, the resource cache library will be maintained, then proxy server returns response to build proxy again.

Reference to the cross layer optimization, the resource cache library maintenance scans each storage space orderly, and the IPv6 address can be calculated with the server number read from storage space. According to the address, proxy server will check whether the child node is in router table, and proxy server keeps the node if its item is targeted, conversely, proxy server abandons it. If the triggering conditions are satisfied, the maintenance runs.

Approaching Freshness Modem. Learning from the freshness model showed in Eq. 1, approaching freshness modem can correctly show the changing tendency of resource with a rational format of resource cache library, whose working principal is formulated as follows:

$$R = \text{Max_age} + \text{Update_timer} - \text{Setup_wait} - \text{Request_timer} \quad (2)$$

There involves four data, Max_age defines option value of data lifecycle described in update request; Update_timer records the time when proxy server receives the update message; Setup_wait represents average time that manipulate the explorer once; Request_timer reveals the moment proxy server gets the request from explorer; and R shows the changing tendency of resource server in advance. If R is greater than zero, there is no change of target resource before next explorer operation stops; or some changes may be produced.

Testing and Verification

The CoAP scheme is designed and implemented in IPv6 over Low power WSNs (6LoWPAN) testbed with optimized proxying and cache mechanism. The 6LoWPAN platform is shown in Fig. 3. And the test environment is shown in Fig. 4.



Fig.3 6LoWAN testbed

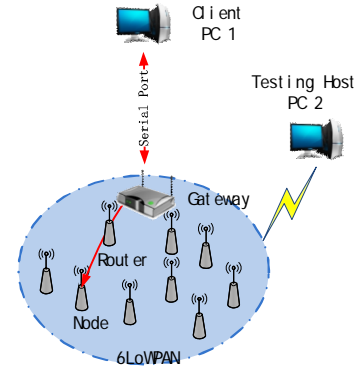


Fig.4 Test environment

For performance, the proposed proxying and cache mechanism is lightweight, low-throughput and short delay, thus the test mainly focus on resource occupation, router throughput, and Round-Trip Time (RTT).

Resource Occupation Test. Contiki is an open source operating system for networked, memory-constrained systems, it provided powerful technical support for WSN together with 6LoWPAN. Based on this, we choose Contiki and 6LoWPAN as testbed and separately name them A and B, they are both configured with IEEE 802.15.4, IPv6, RPL, 6LoWPAN, UDP, CoAP. Applications in testbed add same function support (discover, created, read, update, delete, observe and backward proxying) for CoAP. We debug A and B with IAR Embedded Workbench, the test result is shown in Fig. 5.



Fig.5 Storage resource usage

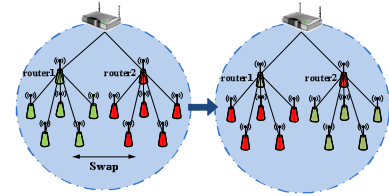


Fig.6 Test environment setup

As we can see from Fig. 5, A occupies less FLASH, and 6LoWPAN is more lightweight.

Router Throughput Test. We pick the 6LoWPAN as testbed and divide it into group A and group B, which are equipped with original and optimized proxying and cache mechanism, respectively. The sleeping period of nodes in A and B is set to 5 minutes, test goes on according to the following procedures:

- 1) Start packet sniffer on PC 2 and capture the data frames;
- 2) After the network established and topology stabilized, swap the child nodes in A and B as Fig. 6 shows (topology and proxy expressed in form of line and color);
- 3) Stop the packet sniffer at 30 minutes after A's topology is stable;

4) Count the length of data frame from router 1 in every 5 minutes, the throughput of router 1 comes from A and B is calculated and presented in Fig. 7, which changes with time.

Note that most nodes in WSN are fixed, without fault router and environment disturbances, the router maintenance is only related to a little changes in topology. To make the phenomenon more obvious, we excute step 2.

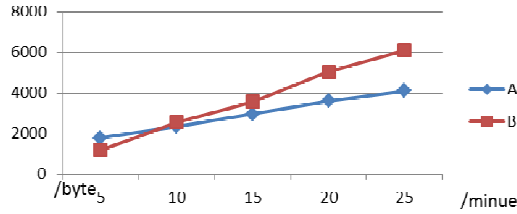


Fig.7 Test results of router throughput

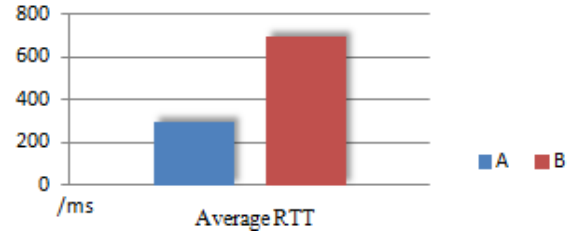


Fig.8 Test results of RTT

As shown in Fig. 6, although some message interacted when the proxy is maintained in network A, router throughput in A is still healthier than in network B. In addition, with the increasing node and maintenance frequency, the advantage of router throughput is more obvious. The enhanced proxying and cache mechanism not only effectively reduces router throughput but also improves the network performance.

Round-Trip Time Test. RTT indicates the interval time from data sending to acknowledgment receiving (default is to send acknowledgment while data is receiving). Delay includes transmission delay, propagation delay, processing delay and queuing delay [8].

To compare the packet RTT of update and response, we employ two sets of 6LoWPAN armed with original and optimized proxying and cache mechanism, one is defined as A and another is defined as B. And the test steps are as follows:

- 1) Reverse child nodes of router 1 and router 2 after network shaped and topology fixed, as shown in Fig. 6;
- 2) Run serial debugging assistant on PC 2 and print the debug information of router 1's child nodes in A and B;
- 3) Calculat the RTT of update and response packet in A and B with the printed information.

The test results of RTT are shown in Fig. 8, where shows the optimized proxying and cache mechanism make the network more excellent with less RTT.

Summary

In this paper, we presented a CoAP scheme with the enhanced proxying and cache mechanism. The proposed scheme was implemented and tested in 6LoWPAN networks. Results showed that the proposed improvement has the advantage of low network delay.

Acknowledgment

This work was supported by the National High Technology Research and Development Program of China (Grant No. 2015AA043801).

References

- [1] Zhu Chuan, Zheng Chunlin, Shu Lei, et al. A survey on coverage and connectivity issues in wireless sensor networks, J. Journal of Network and Computer Applications. 35 (2012) 619-632.
- [2] Mainwaring A, Culler D, Polastre J, et al. Wireless sensor networks for habitat monitoring. Proc. of the 1st ACM international workshop on Wireless sensor networks and applications. New York: ACM Press, 2002. 88-97.

- [3] Levin S, Schmidt S. IPv4 to IPv6: Challenges, solutions, and lessons, J. Telecommunications Policy. 38(2014) 1059-1068.
- [4] Joel J. P. C. Rodrigues, Paulo A. C. S. Neves. A survey on IP-based wireless sensor network solutions, J. International Journal of Communication Systems, 23 (2010) 963-981.
- [5] Shelby Z, Sensinode. RFC 6690: Constrained RESTful Environments (CoRE) Link Format, IETF, August 2012.
- [6] Shelby Z, ARM, Hartke K, et al. RFC 7252: The Constrained Application Protocol (CoAP), IETF, June 2014.
- [7] Dunkels A, Gronvall B, Voigt T. Contiki-a lightweight and flexible operating system for tiny networked sensors. Proc of the 29th Annual IEEE International Conference on Local Computer Networks. Washington D C: IEEE Computer Society Press, 2004. 455-462.
- [8] Fikret Sivrikaya, Bulent Yener. Time Synchronization in Sensor Networks: A Survey, J. IEEE Network. 18 (2004) 45-50.