Applied Soft Computing 68 (2018) 811–820

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An effective handling of secure data stream in IoT

Jaejin Jang [a], Im.Y Jung [a,*], Jong Hyuk Park [b,*]

[a] School of Electronics Engineering, Kyungpook National University, 80 Daehakro, Daegu 41566, South Korea
[b] Dept. of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea

ARTICLE INFO

Article history:
Received 27 January 2017
Received in revised form 20 April 2017
Accepted 8 May 2017
Available online 17 May 2017

Keywords:
Secure data stream
IoT
Usability

ABSTRACT

Internet-of-Things (IoT) applications are a primary domain for data streams, which travel through a heterogeneous network consisting of the Internet and low-speed IoT to be sent to a data collector. However, when a large data stream is transmitted, overhead results from the difference between the speed and maximum transfer unit of IoT and the existing Internet. The overhead increases as data size increases. This problem is a critical factor for IoT devices that are sensitive to power consumption and data streams that must be handled in real time. To solve this problem, we compressed the data stream using a low-density parity check (LDPC) code. Since compression using the LDPC code can be applied even when the data stream is encrypted, the compression can be used in applications requiring privacy or confidentiality. Therefore, this study proposes a method to improve the usability of encrypted data streams in the IoT environment. We implemented IoT devices that generated data streams using Raspberry Pi, a desktop computer, and collectors that collect data streams. The results of experiments using temperature sensor data show that the communication time for data stream transmission decreased by 56.1–75.5%. In addition, the power consumption of IoT devices for data transmission decreased by 54.8–75.3%. In order to perform compression handling by the IoT device, the maximum memory usage and CPU usage increased by 0.3% and 10.1%, respectively. As a result of this research, it is expected that the transmission time to collectors, as well as the power consumption of IoT devices, can be reduced while securing data streams generated by IoT devices.

1. Introduction

The Internet of Things (IoT) is a key domain in which data streams are generated, and the volume of this data is expected to increase [1–9]. A data stream has an unbounded size, the rate at which the data is generated varies widely, and the data must be processed online or in near real–time [5–7,9–20]. A data stream collector collects data streams, and the collected information is processed to determine its knowledge or value [21]. When a data stream generated from an IoT device is transmitted to a collector, the data stream traverses a heterogeneous network composed of the IoT and Internet. The IoT often uses a slow network such as Bluetooth Low Energy (BT-LE), ZigBee, or Z-wave to accommodate the limited resources of IoT devices, while the Internet uses fast networks such as Ethernet or Wi-Fi [22]. In addition, since the IoT network has a small maximum transfer unit (MTU), as the size of data to be transmitted increases, the number of fragmentations and reassemblies for transmission increase sharply compared to the Internet. Table 1 lists the speeds and MTUs of the Internet and IoT [23–30].

Owing to this difference in network performance, when data is transmitted from IoT devices to the Internet, most of the transmission time is consumed in the IoT network.

Fig. 1 shows the proportion of transmission time consumed when various sizes of data are transmitted from the low-speed IoT to the Internet. As seen in the figure, when transmitting to the collector, most of the time is consumed by the transmission to an IoT border router. Table 2 shows that this process consumes more than 80% of the transmission time in the IoT network.

Increases in transmission time and fragmentation/reassembly tasks depending on the data size are critical to IoT devices. This is because IoT devices have limited resources and are sensitive to power consumption [31]. Table 3 lists the specifications of the devices used to implement a popular IoT prototype [32]. As seen in the table, the central processing unit (CPU) used is slower than a desktop or laptop, the RAM is smaller, and the power operates at a low voltage level [33].

To solve this problem, we propose a handler that can improve the usability of the data stream. This method can be applied to an

* Corresponding authors.
E-mail addresses: iyjung@ee.knu.ac.kr (Im.Y Jung), jhpark1@seoultech.ac.kr (J.H. Park).

https://doi.org/10.1016/j.asoc.2017.05.020
1568-4946/

Table 1
Speeds and MTUs of Internet and IoT networks.

Network		Speed	MTU (bytes)
Internet	802.3 (Ethernet)	10 MB–107 GB/s	1500
	802.11ac (Wi-Fi)	454 MB–7.26 GB/s	2304
IoT	Z-Wave	40–100 KB/s	127
	ZigBee	250 KB/s	127
	Bluetooth Low Energy	1 MB/s	27

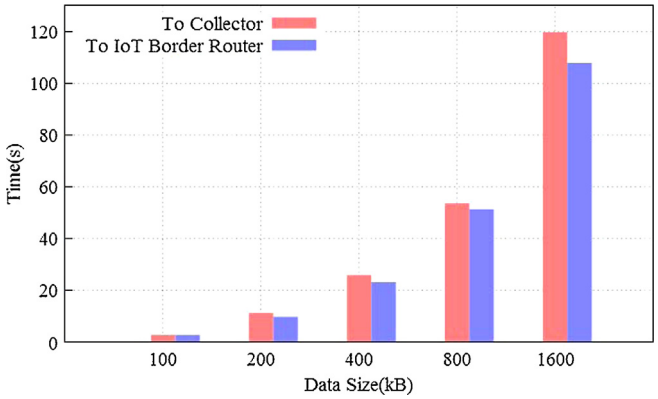

Fig. 1. Specific weight of transmission time in Internet and low speed IoT.

Table 2
Specific weight rates of transmission time.

Data size (KB)	Rate (%)
100	100
200	87.09
400	89.58
800	95.66
1600	90.09

Table 3
CPU, RAM, and power of IoT prototype platform.

Name	CPU	RAM	Power
Raspberry Pi	ARM BCM2835	256–512 MB	5 V/USB
Arduino	ATMEGA8, ATMEGA168 ATMEGA328, ATMEGA1280	16–32 KB	7–12 V/USB
BeagleBone Black	AM335x 1GHz ARM CoretexA8	512 MB	5 V
Phidgets	PhidgetSBC	64 MB	6–15 V
Udoo(Quad)	Freescale i.MX6Quad 4 ARM CoretexTM–A9 Atmel SAM3X8E ARM CortexM3 CPU	1 GB	6–15 V

encrypted data stream, as well as applications requiring security. By compressing the data stream using low-density parity check (LDPC) codes, the size of the data stream is reduced and transmission speeds up as the size decreases.

A reduction in transmission time can help to satisfy the characteristics of the data stream, which should be processed in real time and reduce the power consumption of the IoT device.

A variety of studies have been conducted to overcome the differences between Internet and IoT networks, but the research has focused on increasing the payload by compressing the header of each protocol. However, in this study, we solved the problem by compressing the payload itself rather than the header, even if the payload was encrypted.

The results of experiments with IoT devices and collectors using Raspberry Pi and a desktop computer show that the average transmission time to a collector was reduced by 61.63%, while the average power consumption of an IoT device was reduced by 61.08%. The CPU and memory usage for performing the proposed handling increased by 0.50% and 0.30%, respectively, for a data size of 1600 KB, which was the largest usage.

The contributions of this study are as follows:

- Reduced transmission time to data stream collector. The data stream has characteristics that should be processed in real time. In order for the data stream to be processed, it must first be collected into a collector. In an IoT environment, however, the data stream is delayed owing to network differences. We reduced the delay by an average of 61.63% by compressing data using an LDPC code. This can help satisfy the data stream characteristics that need to be processed quickly.
- Reduced power consumption of IoT devices. Since an IoT network uses a slow network such as BT-LE, ZigBee, or Z-wave, the transmission time increases significantly when the size of the transmitted data stream increases. In addition, because these slow networks have small MTUs, fragmentations and reassemblies for data transmission also increase. This increase is a critical factor in IoT devices that are resource constrained and sensitive to power consumption. We reduced the power consumption for data stream transmission by compressing the data stream to reduce its size. Thus, the power consumption for a data stream that should be transmitted frequently can be reduced. In order to perform the proposed handling, the Raspberry Pi has a run time of less than 1 s, and an increase in memory and CPU usage is acceptable.
- Payload compression studies of packets. Studies aimed at reducing the network performance gap between the IoT and Internet have focused on the header compression of packets. In contrast, this study investigated the payload compression of packets. This method is applicable even if the payload is encrypted. This new approach could provide a new approach for existing research that focuses only on header compression.

This paper is organized as follows; Section 2 explains background for our study-basic terminology and theories. In Section 3, we discuss related works. In Section 4, we propose a method for effective handling of secure data stream. Section 5 presents the experiments and analysis, including implementation, experimental method, and interpretation of results. In Section 6, we conclude our study and discuss future work.

2. Background

In order to understand related research, this section describes the IoT network structure and its basic terminology and theories.

Network architecture of the IoT: Fig. 2 shows an example of an IPv6 network and a 6LoWPAN network [34]. All hosts, including hosts in 6LoWPAN, have IPv6 addresses. Each router is responsible for routing between hosts in the sub network and external hosts. In this study, a device in the 6LoWPAN network in Fig. 2 generates the data stream, and the server is a collector that collects the data stream.

IPv6 over Low-Power WPAN (*6LoWPAN*): 6LoWPAN is a low-power network that supports IPv6 and is based on IEEE 802.15.4 [35]. In the case of IPv4, which is currently widely used, addresses cannot be given to IoT devices owing to a lack of address space. IPv6 is applied to solve this problem [36]. It is based on IEEE 802.15.4, a standard network for low-power devices to satisfy the low-power characteristics of IoT devices.

Constrained Application Protocol (*CoAP*): CoAP is a protocol for supporting the application layer in resource-limited devices such as IoT devices [37]. It is based on User Datagram Protocol (UDP)