IoT gateway – implementation proposal based on Arduino board

Artur Grygoruk
R&D Center Orange Poland
ul. Obrzeżna 7
02-691 Warsaw, Poland
Email: artur.grygoruk@orange.com

Jarosław Legierski
R&D Center Orange Poland
ul. Obrzeżna 7
02-691 Warsaw, Poland
Email: jaroslaw.legierski@orange.com

Warsaw University of Technology, Faculty of Electronics and Information Technology
ul. Nowowiejska 15/19 00-665 Warsaw, Poland

Warsaw University of Technology, Faculty of Mathematics and Information Science,
ul. Koszykowa 75, 00-662 Warsaw, Poland

Abstract — The paper presents proposal of practical implementation simple IoT gateway based on Arduino microcontroller, dedicated to use in home IoT environment. Authors are concentrated on research of performance and security aspects of created system. By performed load tests and denial of service attack were investigated performance and capacity limits of implemented gateway.

I. INTRODUCTION

IoT gateway concept is one of the most important aspect in Internet of Things idea. This network element is presented as a proxy between sensing network and application layers. Many small and autonomous devices and sensors urgently needs this component for communication with higher layers of the network.

In contemporary world of digital communication one of the major aspects is data transmission protection. The ways of implemented protection mechanisms depends on infrastructure details and characteristics of services dedicated to end users. IoT Gateway is a very interesting approach from this point of view and very often a single point of failure for IoT infrastructure. IoT gateway installed in single instance can be observed as Single Point of Failure [2] and is really vulnerable to all threats which are based on network traffic volumetric attack.

II. EXISTING SOLUTIONS

There can be defined two types of IoT gateway implementations. The first one: gateway installed in form of dedicated software located e.g. in typical wireless router or in smartphone as an application [2]. The second implementation is based on a gateway which is using a dedicated hardware. IoT gateway must meet requirements such as: hardware low cost, easy extensibility and application-layer support. The fact is that standardized to different network platforms IoT gateway doesn’t exist. Each type of IoT device and each vendor uses own IoT gateway implementation e.g. on smartphones which each supplier can offer client buying his own IoT gateway application which is installed in smartphone device. This approach is generally different than presented in Wifi segment where each computer, tablet or smartphone can use one common Wifi gateway.

Because of low hardware cost, availability and possibility of modifying the hardware and software in very easy way, the authors of this paper created a prototype of IoT gateway based on Arduino platform. Arduino is an open-source electronics single board microcomputer based on easy implementable hardware and software. In the literature we could find many examples of using Arduino (mostly used as sensor node) and different most advanced platform such as Raspberry Pi which were used to build the IoT Gateway: [4],[5],[6], [7]. In the literature is presented only one similar IoT gateway implementation based on Arduino. In [3] authors presents the concept of Arduino board based IoT gateway dedicated specially to the medical purposes. This gateway implemented on Arduino Yun is dedicated for monitoring vital parameters of human body using different sensors such as: heart rate sensor, blood pressure, pulse oximetry or body temperature.

III. IoT HOME GATEWAY CONCEPT

IoT Gateway is a single place where a lot of sensors and other components can communicate with applications using standard protocols included in wireless technology like mobile networks or WiFi. Sensing domain elements are connected to one root component which is a point of communication with rest part of devices [1]. IoT home gateway [2] can be defined as an element connecting simple sensors installed in dedicated network often connected wireless e.g. using BLE (Bluetooth Low Energy) technology with home network layer.

![IoT Home gateway concept](image)
As it was presented on Figure 1. IoT gateway fulfils a role of point between sensors and network domain, where tiny sensors can communicate with other subnetworks. For IoT networks it is recommended to use protocols which packets have low overhead and stateless communication method e.g. Constrained Application Protocol (CoAP). CoAP is an application part protocol to present readings from sensors in unreliable transmission. It uses only UDP datagrams to send information very fast and with low latency. It is great protocol for reading sensor data and controlling actuators to drive motors.

IV. SYSTEM ARCHITECTURE

On Figure 2. high level system architecture was presented. Arduino based on IoT gateway (2) collects an information from different sensors from sensing domain (1). In tested environment: Passive Infra Red sensor, sound sensor, pressure and smoke sensors were used. Web Application (3) hosted on board presents information from sensors and exposes them for third party systems. In implemented gateway two different Ethernet Modules (4) (Ethernet ENC28j60 and Wiznet550 with tcp offload capability) were used to provide connection to network domain. As IoT gateway system core element three boards: Arduino UnoR3, Mega2560 and Leonardo (2) were tested. As third party application http client (6) e.g. web browser was connected to network domain (5).

Moreover, the IoT Gateway designed architecture provides a connection to web service which presents in browser the information returned by each sensor.

A) Arduino environment

Two web applications which fulfil a crucial role of plugin for sensing domains, have been developed and installed on Arduino Leonardo, Mega2560 and UnoR3 supported by different Ethernet Modules (hardware Wiznet550 and software ENC28j60). Configuration of Ethernet Network Module which is presented on Fig. 3. was provided by ICSP pin with settings and configuration on board side. User can send and read information from sensors by using http client and dedicated web application

B) Web Application

Web application was hosted on Arduino board. It was a standard web server which exposed information from connected sensors. It displayed a single html page with presented a few changing values of sensor readings after refreshing the web page content from web browser.

V. MEASUREMENTS

A) Test environment

Fig. 4. presents test environment used for IoT gateway performance tests. Based on Apache JMeter tool (3) http traffic to Arduino Gateway was generated (1). IoT gateway returned web page (2) with information from sensors (with http 200 message). In other cases no response or http 404 message was generated. CPU/RAM meter (4) based on Linux script running on dedicated Laptop was used to measure Arduino device performance. Moreover, the second part of the research was performed with usage of Hping3 security tool. It gives a lot of possibilities to generate a huge traffic simulating Distributed Denial of Service. TCP SYN Flood and Http Slowloris attacks were used to block an access to a web service. Unfortunately, it could be observed that service is very sensitive to receiving big part of http traffic. It was unresponsive to next requests sent by clients. It had to be restarted to provide access after web logic failure.
B) Load test – error rate

Based on test scenario described in details in point A of this paper load tests for three Arduino boards and two Ethernet cards were performed. Figures 5-10 presents results of load tests (observed packet error rate in network traffic).

Moreover, based on the result of load test for Wiznet550 network adapter module maximal number of 15 simultaneous sessions was defined. Large number of sessions results in showing high error rate of receiving packets. The better result was observed for ENC28J60 Ethernet card and for this component the lower error rate about 5-8% for 300-800 TCP sessions was detected.

A) Load test – average response time

Besides the error rate during the load tests average response time was measured.

Table 1 and 2 presents observed CPU and RAM usage. It can be observed that percentage usage of CPU for Arduino board is higher about 10% when Enc28j60 Ethernet module was connected to device. It is a result of receiving and parsing packets on core of application web server. When Wiznet550 tcp off load connector is used a
big part of packet processing is handled by this element. It protects presented gateway against change of web server logic into block state.

<table>
<thead>
<tr>
<th>Number of Threads</th>
<th>Arduino Leonardo</th>
<th>Wiznet ENC28j60</th>
<th>Enc28j60</th>
<th>Arduino Leonardo</th>
<th>Wiznet ENC28j60</th>
<th>Enc28j60</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4,500</td>
<td>1,065</td>
<td>6,882</td>
<td>15,900</td>
<td>10,317</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>1,667</td>
<td>1,008</td>
<td>6,981</td>
<td>22,367</td>
<td>10,138</td>
<td>9,991</td>
</tr>
<tr>
<td>10</td>
<td>9,333</td>
<td>0,935</td>
<td>6,036</td>
<td>15,367</td>
<td>9,608</td>
<td>9,802</td>
</tr>
<tr>
<td>20</td>
<td>6,133</td>
<td>0,852</td>
<td>7,233</td>
<td>16,800</td>
<td>1,276</td>
<td>10,020</td>
</tr>
<tr>
<td>30</td>
<td>3,467</td>
<td>0,898</td>
<td>7,226</td>
<td>10,233</td>
<td>1,123</td>
<td>9,945</td>
</tr>
<tr>
<td>40</td>
<td>3,500</td>
<td>0,797</td>
<td>7,638</td>
<td>13,133</td>
<td>0,999</td>
<td>9,836</td>
</tr>
<tr>
<td>50</td>
<td>12,467</td>
<td>1,113</td>
<td>7,326</td>
<td>9,467</td>
<td>1,126</td>
<td>9,797</td>
</tr>
<tr>
<td>100</td>
<td>10,500</td>
<td>0,906</td>
<td>5,638</td>
<td>16,200</td>
<td>1,042</td>
<td>10,235</td>
</tr>
<tr>
<td>200</td>
<td>6,833</td>
<td>0,805</td>
<td>6,112</td>
<td>6,300</td>
<td>1,137</td>
<td>-</td>
</tr>
<tr>
<td>300</td>
<td>8,700</td>
<td>0,808</td>
<td>6,730</td>
<td>13,233</td>
<td>0,949</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2. Load test result – RAM usage [%]

<table>
<thead>
<tr>
<th>Number of Threads</th>
<th>Arduino Leonardo</th>
<th>Wiznet ENC28j60</th>
<th>Enc28j60</th>
<th>Arduino Leonardo</th>
<th>Wiznet ENC28j60</th>
<th>Enc28j60</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35,900</td>
<td>14,627</td>
<td>38,833</td>
<td>35,850</td>
<td>24,578</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>37,500</td>
<td>14,700</td>
<td>38,851</td>
<td>34,900</td>
<td>24,827</td>
<td>38,474</td>
</tr>
<tr>
<td>10</td>
<td>38,600</td>
<td>14,157</td>
<td>38,985</td>
<td>34,500</td>
<td>24,941</td>
<td>38,565</td>
</tr>
<tr>
<td>20</td>
<td>38,100</td>
<td>14,780</td>
<td>38,822</td>
<td>35,133</td>
<td>14,479</td>
<td>38,531</td>
</tr>
<tr>
<td>30</td>
<td>37,900</td>
<td>14,793</td>
<td>38,731</td>
<td>35,300</td>
<td>14,497</td>
<td>38,539</td>
</tr>
<tr>
<td>40</td>
<td>38,400</td>
<td>14,791</td>
<td>38,682</td>
<td>34,867</td>
<td>14,699</td>
<td>38,483</td>
</tr>
<tr>
<td>50</td>
<td>37,500</td>
<td>14,792</td>
<td>38,827</td>
<td>34,200</td>
<td>14,690</td>
<td>38,507</td>
</tr>
<tr>
<td>100</td>
<td>39,400</td>
<td>14,788</td>
<td>39,314</td>
<td>34,800</td>
<td>14,722</td>
<td>38,197</td>
</tr>
<tr>
<td>200</td>
<td>38,900</td>
<td>14,800</td>
<td>39,310</td>
<td>39,300</td>
<td>19,700</td>
<td>-</td>
</tr>
<tr>
<td>300</td>
<td>38,733</td>
<td>14,800</td>
<td>39,191</td>
<td>38,900</td>
<td>19,669</td>
<td>-</td>
</tr>
</tbody>
</table>

RAM percentage usage results presented for Arduino boards don't depend on kind of Ethernet module.

Sometimes the usage WIZnet550io module resulted in the error XML Parse Exception. It is a result of blocking a part of data on the same link which is used by legitimate user and by attackers. Service doesn't have to be suspended in case of receiving too many requests to web application. Packet error rate (PER) has always been the biggest value for Ethernet module WIZnet550io. In contrast to Wiznet550io Ethernet ENC28j60 requires a large number of users to return permanent failure. It leads to rejection of the packets due to insufficient capacity of Arduino board resources at very high network traffic.

VI. FUTURE WORK

In the future, the authors of this publication are planning to focus on improving security aspects. The most important is implementation of encryption for data transmission between the IoT gateway and http client [8]. Nowadays a lot of Man in The Middle attacks can be performed successfully because of sending data via plain text. Implementation of SSL protocol and certificate on web server side should prevent against this situation.

Due to the fact that only limited resources are available such as: CPU, memory and number of I/O ports the implementation of similar IoT environment based on more advanced hardware platform should be performed. For example the usage of Intel Galileo should allow to implement some additional features such as Web application or firewall features inside the board because of better hardware capabilities in comparison with Arduino board. The use of Intel Galileo allows to keep compatibility of sensors layer with the platform Arduino and allows to focus on the software development.

VII. SUMMARY

Presented in this paper Arduino boards are offered as a standalone device without network adapter. For load tests Arduino boards were equipped with Wiznet550io module and ENC28j60 network adapter. Better results of performance were observed for ENC28j60 from 10 TCP sessions for Arduino Leonardo to 160 TCP sessions for Arduino UnoR3.

As a device with very limited hardware Arduino can host services which can't be stable if too many users want to get access to the resources in the same time. Arduino board can be used as the IoT gateway only in very small IoT environment. ENC28j60 Ethernet module could better cope with http request avalanche when in the same time Wiznet550io module was really poor to protect IoT gateway. However, it should be emphasized that there is a need to build new solution for IoT gateway protection.

Prototype of IoT Gateway was made as part of the Open Middleware 2.0 Community by Orange Labs program [9].

REFERENCES