AFIN 2017

The Ninth International Conference on Advances in Future Internet

ISBN: 978-1-61208-583-8

September 10 - 14, 2017

Rome, Italy

AFIN 2017 Editors

Renwei (Richard) Li, Future Networks, Huawei, USA
Eugen Borcoci, University Politehnica of Bucharest, Romania
Kiran Makhijani, Huawei Technologies, USA

AFIN 2017

Forward

The Ninth International Conference on Advances in Future Internet (AFIN 2017), held between September 10-14, 2017 in Rome, Italy, continued a series of events dealing with advances on future Internet mechanisms and services.

We are in the early stage of a revolution on what we call the Internet now. Most of the design principles and deployments, as well as originally intended services, reached some technical limits and we can see a tremendous effort to correct this. Routing must be more intelligent, with quality of service consideration and 'on-demand' flavor, while the access control schemes should allow multiple technologies yet guarantying the privacy and integrity of the data. In a heavily distributed network resources, handling asset and resource for distributing computing (autonomic, cloud, on-demand) and addressing management in the next IPv6/IPv4 mixed networks require special effort for designers, equipment vendors, developers, and service providers.

The diversity of the Internet-based offered services requires a fair handling of transactions for financial applications, scalability for smart homes and ehealth/telemedicine, openness for web-based services, and protection of the private life. Different services have been developed and are going to grow based on future Internet mechanisms. Identifying the key issues and major challenges, as well as the potential solutions and the current results paves the way for future research.

The conference had the following tracks:

- Internet services and applications
- Internet challenges

We take here the opportunity to warmly thank all the members of the AFIN 2017 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to AFIN 2017. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also gratefully thank the members of the AFIN 2017 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that AFIN 2017 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of the future Internet. We also hope that Rome, Italy provided a pleasant environment during the conference and everyone found some time to enjoy the historic charm of the city.

AFIN 2017 Chairs

AFIN Steering Committee

Renwei (Richard) Li, Future Networks, Huawei, USA
Eugen Borcoci, University Politehnica of Bucharest, Romania
Alex Galis, University College London, UK
R.D. van der Mei (Rob), Centre for Mathematics and Computer Science (CWI), the Netherlands
Jun Peng, University of Texas - Rio Grande Valley, USA
Hiroyuki Sato, University of Tokyo, Japan
Sergio Ilarri, University of Zaragoza, Spain
Christos Bouras, University of Patras and Research Academic Computer Technology Institute, Greece
Adel Al-Jumaily, University of Technology, Sydney, Australia

AFIN Research/Industry Committee

Kiran Makhijani, Huawei Technologies, USA
Alexander Papaspyrou, adesso GmbH, Germany
Martin Zelm, INTEROP - Virtual Lab, Brussels, Belgium

AFIN 2017
Committee

AFIN Steering Committee

Renwei (Richard) Li, Future Networks, Huawei, USA
Eugen Borcoci, University Politehnica of Bucharest, Romania
Alex Galis, University College London, UK
R.D. van der Mei (Rob), Centre for Mathematics and Computer Science (CWI), the Netherlands
Jun Peng, University of Texas - Rio Grande Valley, USA
Hiroyuki Sato, University of Tokyo, Japan
Sergio Ilarri, University of Zaragoza, Spain
Christos Bouras, University of Patras and Research Academic Computer Technology Institute, Greece
Adel Al-Jumaily, University of Technology, Sydney, Australia

AFIN Research/Industry Committee

Kiran Makhijani, Huawei Technologies, USA
Alexander Papaspyrou, adesso GmbH, Germany
Martin Zelm, INTEROP - Virtual Lab, Brussels, Belgium

AFIN 2017 Technical Program Committee

Rocío Abascal-Mena, Universidad Autónoma Metropolitana - Cuajimalpa, Mexico
Cristina Alcaraz, University of Malaga, Spain
Muhammad Aleem, Capital University of Science and Technology (CUST), Pakistan
Adel Al-Jumaily, University of Technology, Sydney, Australia
Rachida Aoudjit, Université Mouloud Mammeri de Tizi Ouzou, Algeria
Zubair Baig, Edith Cowan University, Australia
Paolo Bellavista, University of Bologna, Italy
Ana M. Bernardos, Universidad Politécnica de Madrid, Spain
Peter Bloodsworth, University of Oxford, UK
Eugen Borcoci, University Politehnica of Bucharest, Romania
Kechar Bouabdellah, University of Oran 1 Ahmed Ben Bella, Algeria
Christos Bouras, University of Patras and Research Academic Computer Technology Institute, Greece
Manuel José Cabral dos Santos Reis, University of Trás-os-Montes e Alto Douro, Portugal
Lianjie Cao, Purdue University, West Lafayette, USA
Kevin Chalmers, Edinburgh Napier University, UK
Fisnik Dalipi, Linnaeus University, Sweden
Maurizio D'Arienzo, Università della Campania Luigi Vanvitelli, Italy
Jacques Demerjian, Lebanese University, Lebanon

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Proactive and Reactive Mechanisms for Protecting Ads on the Internet from Adware and Malware

Abinash Sarangi
Bing, Artificial Intelligence and Research
Microsoft
Redmond, USA
absarang@microsoft.com

Abstract— **Ads on websites and search engines help keep the Internet services free and accessible to all. These ads are vulnerable to malicious attacks by adware, malware on user's machine and user agent. A webpage has several limitations in its ability to protect its integrity on the user agent. The proposed solution provides the webpage with the ability to incorporate a two-layer protection in preventing malware and restoring the webpage's integrity.**

Keywords-adware prevention; javascript; page validation rules; mutation observer; malicious code injection; malware prevention; Web security.

I. INTRODUCTION

With the Internet's growth, adwares are adopting sophisticated mechanism, running as browser plugins or as background service in order to attack on selective websites. Adware and malware attack is a several millions-dollar industry [2][3] and is being democratized by advanced tools, such as black hole exploit kit [3][4]. A simple adware acting as a browser plugin can manipulate the DOM (Document Object Model) of any website (e.g. Bing or Google) and replace the original ads with its own fraudulent ads. This results in loss of revenue for the affected website. Our research for the work, discussed in this article, suggested at least 4.5% of Bing users had some form of an adware or malware, which inserted unwanted content into Bing's search engine results page, resulting in several million dollars of revenue impact [1][5]. This is true for Google, Facebook and any such Internet service which monetizes using ads [7]. The real challenge of this problem is, the malicious program is running on the user's machine and Internet services are accessed using Web browsers on that machine, limiting what such a service provider can do to prevent the adware from within the webpage itself. This is the very reason why Web services have not been successful in dealing with malware/adwares [5].

The rest of the paper is structured as follows: in Section II, we discuss the problem in detail. In Section III, we discuss the technological and functional aspect of the solution. In Section IV, we discuss the proposed solution, its effectiveness as observed during the research and experimentation. Finally, we conclude the article in Section V.

II. DETAILED PROBLEM DESCRIPTION

Viruses, malwares, adwares and ransomware are getting sophisticated and performing selective attack on websites to generate revenue for themselves by hijacking ads. Search engines and content portals primarily rely on ads on their results page for revenue. When the Web page is loaded on a browser on a machine which is infected with malwares and adwares, the Web page is systematically attacked, and its content is modified. The page's ads are removed, and malware's ads are injected. All of it happens right on the user's browser after the page loads. The search engine or the Web page loses revenue. There is very little the page can do to defend itself on the user's browser. Some important questions to ask when looking for a solution to this problem include: how does the page ensure the page is rendered on the browser as it was emitted by the server? How to ensure the integrity of its content on the client?

Figures 1 and 2 show malware attacks on Bing and Google, respectively.

III. FUNCTIONAL INSTRODUCTION TO SOLUTION

We researched and experimented several mechanisms and built a JavaScript based framework which can consume a rule set generated by server to validate the state of the website and ensure its integrity. The framework, being JavaScript based, can run from within the website seamlessly on all (modern) browsers and devices and protect the webpage from malicious programs running on the user's machine at a higher privilege. The framework uses the unique rule set pertaining to the current page and allows only valid mutation to the DOM from known sources to the website. If any mutation fails the validation, the DOM is restored to the state prior to the mutation. This mechanism ensures that, even if a user's machine has malicious programs or adwares attempting to inject fraudulent ads, the attempt is prevented because the webpage can protect itself from within and ultimately save significant ad revenue. This research has been implemented and thoroughly measured for success and effectiveness through A/B testing at scale [1][5].

Security of websites and services is a growing challenge, and the future Internet needs more research and awareness in the field to deal with any vulnerabilities or exploits that may exist.

 ISBN: 978-1-61208-583-8

IV. SOLUTION

Our research for the work discussed in this article indicates up to 4.3% of search engine users have plugin/malware that modifies the search engine's results page in a way that interferes with the original content [1]. The problem is wide spread in all markets and more on browsers that support plugins or extensions.

A. How does adware work

On an infected machine, the adware either executes as a background process or injects itself as a browser extension. Once active, the adware monitors each browser navigation and executes its checks and monitoring. For example, on Chrome browser, it executes the content scripts to validate if the website being loaded is a target. Once a targeted website e.g. google.com, bing.com. amazon.com etc. is detected, the adware downloads scripts and content (ad images, videos, text, flash objects) specific to the targeted site. The scripts are added to the original page using DOM injection as Script tag. HTML (Hypertext Markup Language) elements are in turn added by the script to the targeted page's DOM. Depending on how severe the interference is, the injected elements could take up part of the site using HTML elements like: DIV, LI, Object, IMG, IFRAME etc. or cover the entire viewport with elements such as modal dialogues. Once elements are injected, the user sees a mixture of content from the original website and the adware. Stolen styles result in deceiving the user to not being able to discern the original content from adware's fraudulent content. Injected ads are contextual and based on the user query as seen in Figures 1 and 2 below.

Usually, the adware code is Polymorphic [8] and encoded e.g:

```
var
l3=(0x12<(0xF5,17.)?'B':(47,22)<=(111.,53.6E1)?(13.83E2,"
a"):(0x1AA,116.7E1));
```

B. Impact of the problem

- Slower Web pages leading to bad user experience
- Ad revenue loss for websites and search engines
- Privacy and security threat for the users
- Unusable website due to clickjacking

C. Proposed solution

The solution that we propose to this problem is a two-step approach. We call the approaches the proactive defense and the reactive defense. Server-side solutions and code have very limited ability as the problem persists on the client side, on the user's machine. We developed a JavaScript based framework which can run on the browser and leverage on the fact that, when the page is generated on the server, we have the knowledge of what the page's content is. The framework utilizes the server generated knowledge to validate the page once it is rendered on the client and either prevents proactively or removes reactively, any anomalies detected.

1. Proactive defense

Adware and malware use the browser APIs using languages such as JavaScript to manipulate the DOM (Document Object Model) of a page. APIs like insertBefore(…), appendChild(…) are used to insert both user interface elements as well as scripts and style elements. In this proposed approach, we override these browser defined functions to user defined functions much before the onload event is fired on the document. In the user-defined avatar of the functions, we validate the element being inserted, with an allowed list or a baseline and either allow or disallow the insertion. For example, if it's a script tag with a source we do not recognize, disallow the insertion. If we know all images on the current page are base64 encoded images, we can disallow all IMG tags with source attribute set to some 3rd party domain. Figure 4 below demonstrates the proactive mechanism with a block diagram.

2. Reactive defense

Our second step of solution is reactive defense. When coupled with the proactive mechanism, the proposed framework provides a two-layer defense, but each of these mechanisms is independent of each other and can be used stand alone. The reactive mechanism in our experiment has proven to be more effective and robust. As the name suggests, this technique requires a rule set generated at the server for the current page. The rule-set is a table that defines the page layout for consumption by the framework, to validate the page and provide a baseline. The rule-set table is generated at server as a map of relative distance of all page components from a static / fixed point on the page, e.g. the search box on a search engine's results page is usually fixed and rest of the content on the page is dynamic, that may change during the page life cycle. So, the map would look something like:

Data-tag: ads_top: [{x:30,y:60,h:80,w:40}, {ads_bgr, img}]

Here, it defines the element with data tag keyword ads_top with its relative position to the fixed point on the page, its dimensions and some metadata attributes, such as any class names and if it has any children element such as IMG, Object etc.

When this map is populated for all the business-critical components of the page, the reactive frame work can add observers on these elements and validate any mutations, i.e. insertion, deletion, style change, visibility change.

Whenever the mutation observer [9] detects a change and the subsequent validation with respect to the rule-set fails, the reactive framework rejects that change and restores the element to the previous state. In our experiment, the 75th percentile performance number for this operation was about 20ms for a series of 20 mutations. Hence, it is not perceived by the user and the page's perceived performance is not impacted. Refer to Figure 3 below for execution steps and flowchart.

 ISBN: 978-1-61208-583-8

D. Experiment details and outcome

A/B testing experiment [10] was done online on a control and treatment group. Treatment and control both had 1 million users, each in markets across en-US, de-DE, en-CA, fr-Fr, en-GB. The experiment was running for a duration of 2 weeks, while revenue and user engagement data were collected. Revenue, as well as user engagement metrics, were statistically significant in positive move in treatment with p-Values in the order of 10^-15.

As much as 0.6% of revenue increase [1] was achieved. In addition, session success rate and click through rates were positively impacted. Overall, it was a successful experiment from a user as well as business value perspective.

E. State of the art review

Existing solutions to mitigate malwares are primarily at operating system and network security level. The user is required to install antimalware applications or turn on security features in the operating system. Antimalware applications rely on a malware and virus signature database which needs constant updates. Malwares can exploit vulnerabilities and affect the user's machine. Another problem with the existing application based solutions is that many users do not have these applications running on their machines. The proposed solution has no action on the user and protects the website from malicious injections from within the Web site. It protects the website (in our research Bing.com) that incorporates this technique and it remains protected even on an infected machine.

V. CONCLUSION AND FUTURE WORK

This paper presented two approaches to protect websites against malwares: Proactive defense, which prevents malicious script injection and Reactive defense, which detects unauthorized change to the website and removes the change to restore the website to its integral state. Though both approaches can work independently, they are most effective when used in combination, resulting in revenue loss prevention and better user experience.

Planned future improvements to this work include making this generic and building this into the browser as a security feature.

ACKNOWLEDGMENT

This work was developed with help from Windows defender research team: Sarvesh Nagpal, Rahul Lal, Marcelo De Barros. Manish Mittal from Bing search engine team helped with research, implementation and experimentation of the solution.

REFERENCES

[1] Windows defender research, "Malware Protection Center", Microsoft Threat Intelligence Journal, pp. 19-21 https://download.microsoft.com/download/D/C/A/DCACBABC-1711-456B-98E1-180E88BFDC68/MMPC%20Threat%20Intelligence%20October%202015.pdf [accessed July 2017]

[2] Sara Yin, "Flashback Malware Robs Google of $10,000/Day in Ad Revenue", PCMag, May 2012 http://securitywatch.pcmag.com/none/297323-flashback-malware-robs-google-of-10-000-day-in-ad-revenue [accessed July 2017]

[3] Technofaq, "An In-depth Look at the Malware Industry", Technofaq.org, May 2015 https://technofaq.org/posts/2015/05/an-in-depth-look-at-the-malware-industry [accessed July 2017]

[4] Jon Oliver, Sandra Cheng, Lala Manly, Joey Zhu, Roland Dela Paz, Sabrina Sioting, "Blackhole Exploit Kit:A Spam Campaign, Not a Series of Individual Spam Runs", Trend Micro Incorporated research paper, 2012, pp. 1-12 https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_blackhole-exploit-kit.pdf [accessed July 2017]

[5] Ronny Kohavi, Alex Deng, Roger Longbotham, Ya Xu, "Seven Rules of Thumb for Web Site Experimenters", KDD 2014, p. 4 http://www.academia.edu/18352920/Seven_Rules_of_Thumb_for_Web_Site_Experimenters [accessed July 2017]

[6] Robert Siciliano, "Business Identity Theft; Big Brands, Big Problems", HuffingtonPost, Oct 2014 http://www.huffingtonpost.com/robert-siciliano/business-identity-theft-b_b_5643934.html [accessed July 2017]

[7] Larry Dignan, "Google: Click Fraud Costs Us $1 Billion A Year", zdnet, March 2007 http://www.zdnet.com/article/google-click-fraud-costs-us-1-billion-a-year/ [accessed July 2017]

[8] Carey Nachenberg, "Understanding and Managing Polymorphic Viruses", Symantec enterprise papers, volume XXX, pp. 1-4, https://www.symantec.com/avcenter/reference/striker.pdf [accessed July 2017]

[9] Mozilla API Documents, "MutationObservers", July 2017, document explains the API use https://developer.mozilla.org/en-US/docs/Web/API/MutationObserver [accessed July 2017]

[10] Ron Kohavi and Roger Longbotham, "Online Controlled Experiments and A/B Tests", Encyclopedia Of Machine Learning and data Mining, April 2015 http://www.exp-platform.com/Documents/2015%20Online%20Controlled%20Experiments_EncyclopediaOfMLDM.pdf [accessed July 2017]

 ISBN: 978-1-61208-583-8

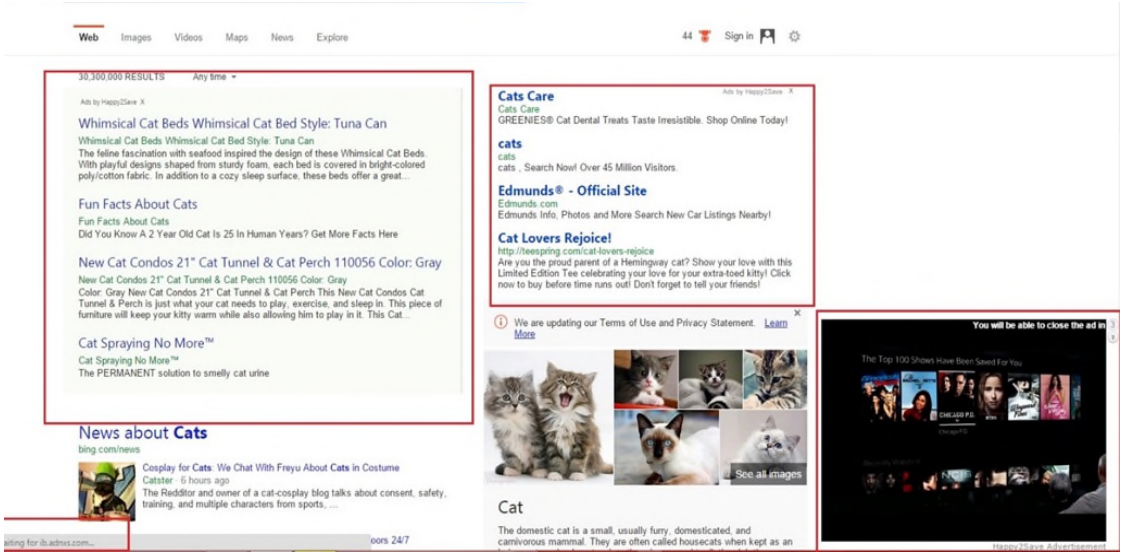

Figure 1. Adware injected ads on Bing's search page. The Ads style (css) mimics that of Bing's styles.

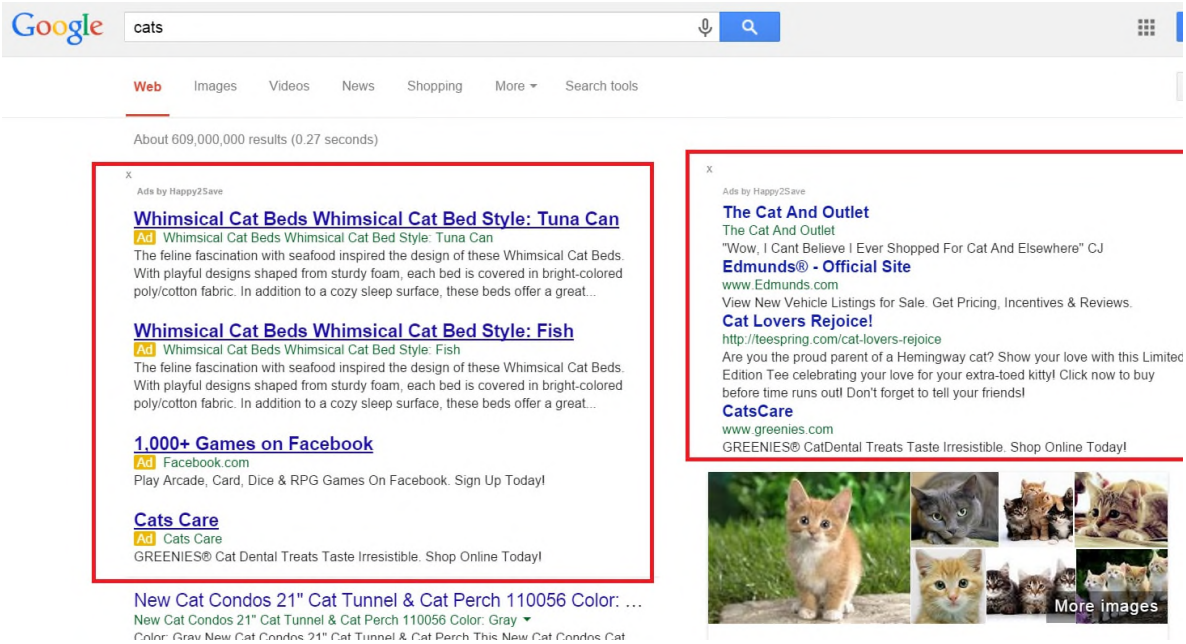

Figure 2. Adware injected ads on Google's search page. The Ads style (css) mimics that of Google's styles.

 ISBN: 978-1-61208-583-8

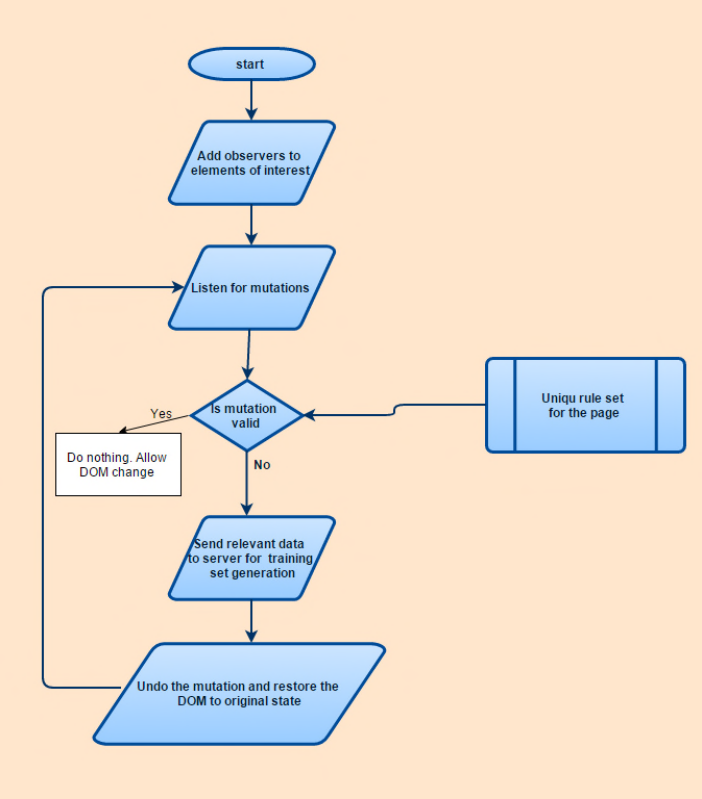

Figure 3: Execution flow of the reactive defense mechanism.

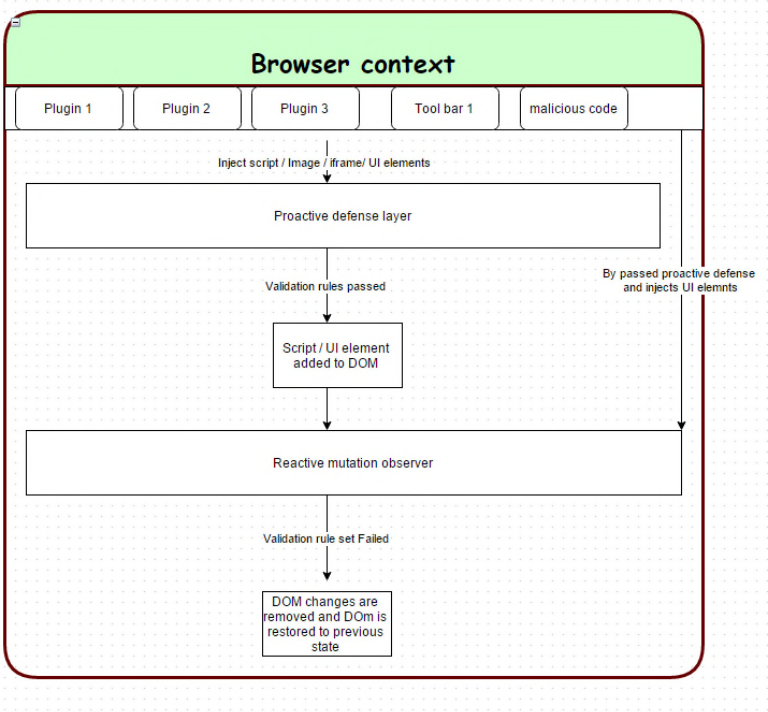

Figure 4: A block diagram explaining where the proactive and reactive defense mechanisms fit in.

 ISBN: 978-1-61208-583-8

Locality-Aware Chord Networks Based on Cloud-Assistance

Chun-Hsin Wang and Cheng-Han Kuo
Department of Computer Science and Information Engineering
Chung Hua University, Hsinchu, Taiwan, R.O.C.
E-mail: chwang@chu.edu.tw; m10202041@chu.edu.tw

***Abstract*—Peer-to-Peer (P2P) technologies have been widely applied in various network applications, such as BitTorrent, P2Plive, Skype, etc. Improving the performance of P2P networks has always been a topic of interest. The key to improving P2P networks is to solve the consistency problem, which states that the overlay network topology is not congruent to the physical network topology. The overlay network is a logical network topology built on existing physical networks. If overlay networks are constructed according to the location of peers, the consistency problem could be solved. In this paper, locality-aware Chord P2P networks based on cloud-assistance are proposed. A cloud database of mapping IP addresses onto geographical positions is built to completely manage the locality information of peers. Four new schemes of identification (ID) assignments associated with geographical positions of peers have been designed to construct the rings of Chord networks. The arrangement of peers on a ring is close to their physical positions. As a result, unnecessary searching routing can be reduced. Besides, a new download method which can select the nearest nodes to get resources has also been designed. The simulation experiments show the searching and downloading performance of our proposed locality-aware Chord networks can be improved.**

Keywords-P2P Networks; Cloud; Overlay Networks; Chord Networks.

I. INTRODUCTION

The ways of sharing resources are diverse, from traditional client-server model to peer-to-peer (P2P) and cloud-based models. Recently, P2P and cloud-based models have appeared in more and more network applications. Compared to P2P networks, cloud services can provide more scalable storage and computing power. However, for a successful cloud-based resource sharing, enough network bandwidth between user nodes and cloud systems is necessary. Besides, the use of P2P networks might be more convenient than cloud services because peers can join P2P networks without any authentication. Therefore, there still exist many popular P2P network applications, such as BitTorrent-like P2P systems [1], PPLive [2], and PPStream [3].

Two basic operations of P2P networks are searching and downloading processes. Peers need to search the desired resource by overlay network [4] to find which peer owns the resource and then download it directly. The overlay network is a logical network topology built on existing physical networks by logically connecting all of joining peers. In general, the edge of the overlay network is set up by a Transmission Control Protocol (TCP) connection. A long-latency delay may exist at the edges of the overlay network because the overlay network topology is not congruent to the physical network topology.

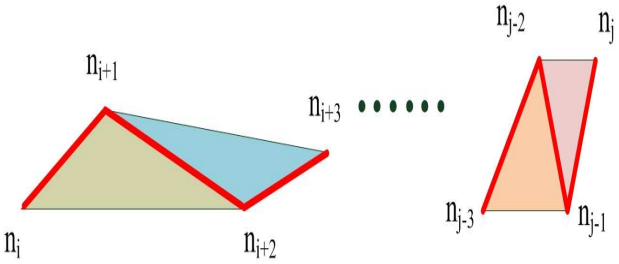

Figure 1. Triangle searching path.

Well-known peer-to-peer systems for file sharing are proposed in Napster [5] and Guntella [6]. The directory of the shared files is managed by a centralized server in Napster. The system scalability and single-point failure problem exist in the Napster system due to the centralized management. Guntella P2P system is also suffering from scalability problems because the message for locating a resource needs to be flooding over the overlay network. The more nodes in the system, the more flooding traffic could occur. To overcome the scalability problems, some structured P2P based on Distributed Hash Tables (DHT), such as Chord [7], CAN [8], and Tapestry [9], have been proposed. Each joining peer and the shared resource are given an unique identifier (ID) respectively generated by the same DHT hash function. The overlay networks are constructed as a tree or ring network topology by the IDs of peers. The information of the shared resources, such as the IP addresses of the owners, are distributed over the joining peers in load balance. In a DHT-based P2P system, a searching query message can be forwarded to its destination within $\bigcirc(logN)$ hops on overlay networks, where N is the number of nodes in the P2P system.

Although the DHT-based P2P systems have the advantage of scalability, the locality information of joining peers is not considered in construction of overlay networks. The consistency problem still exists and unnecessary triangle routing in physical networks could occur during the searching process. In a DHT-based P2P system, a searching query message is forwarded to the next peer until the owner(s) of resource is(are) found. There must exist a searching routing path for each resource query. An example is shown in Figure 1. On the routing path $(n_i, n_{i+1}, n_{i+2}, \cdots, n_j)$, the distance (n_i, n_{i+1}) should not be longer than the distance (n_i, n_{i+2}) in physical networks. It is obvious that the distance (n_{j-3}, n_{j-2}) is longer than the distance (n_{j-3}, n_{j-1}). We refer to this kind of unnecessary routing as triangle routing. The more triangle routing events occur in resource searching, the more delays and network bandwidth is wasted.

 ISBN: 978-1-61208-583-8

In the literature, many related works [10]-[14] attempted to solve the consistency problem by building topology-aware overlay networks with the assistance of locality information of peers, such as Autonomous System (AS), or Internet Service Provider (ISP) peers belong to. The improvement of these works is limited due to lack of precise locality information of nodes. In [14], a packet loss module installed on routers is given the ability to recognize P2P traffic and find out the AS and ISP to which packets belong. When the source and destination IP addresses of P2P packets are not at the same AS or ISP, the packet loss module will drop them with a predetermined probability. This way, peers have a high probability to get resources from their nearby peers and data traffic across different ISPs (or ASs) could be reduced.

Cloud-assisted P2P network applications in [15]-[17] have been proposed in the literature. They focus on some specific P2P applications but the fundamental problems of P2P networks are not further discussed. In [15], cloud services are applied to improve the transmission quality of P2P streaming. In [16], a reliable P2P-based service-oriented architecture is proposed by cloud-assistance. To provide successful resource reservation, the cloud system maintains the capabilities of peers, such as computing power, storage space, and network bandwidth. Resource reservations can be successful by several redundant resources from selected peers. In [17], the cloud system is used to handle spikes in the query of a popular resource. The queries of peers will automatically be transferred to the cloud system to seek popular resources when the load of peers is over the predefined threshold.

The P2P consistency problem can be efficiently improved if the locality information of all joining peers is known. A simple way of managing locality information is by having centralized dedicated servers, but the single point of failure, overloading, and scalability problems can not be avoided. One of the best solution is to migrate locality information of joining nodes from traditional servers to a cloud system with scalable storage and computing power. To completely manage locality information of joining nodes, a free database GeoLiteCity [18] is adopted for mapping IP addresses onto geographical positions. When peers join the system, geographical positions of peers are mapped by a query of GeoLiteCity and saved in another database.

In this paper, we focus on improving the Chord P2P networks based on cloud-assistance. In the original Chord system, a logical ring network topology is constructed in increasing order of IDs of peers in a clockwise direction. A peer can find its predecessor and successor to join the ring of Chord networks according to its ID number generated by a hash function. In fact, the logical position of a peer on the ring depends on its ID number without any concern of its locality information. As a result, the consistency problem can not be controlled. In our proposed locality-aware Chord networks, we have designed new strategies of ID assignments according to geographical positions to have an arrangement of peers on a ring approximately close in their physical positions.

The rest of the paper is organized as follows. The proposed system model based on cloud-assistance is described in the next Section. The locality-aware Chord networks with new schemes of ID assignments are given in Section III. The performance of resource searching and downloading in our proposed Chord networks is evaluated by simulations in Section IV. Finally, some concluding remarks and future work are given in Section V.

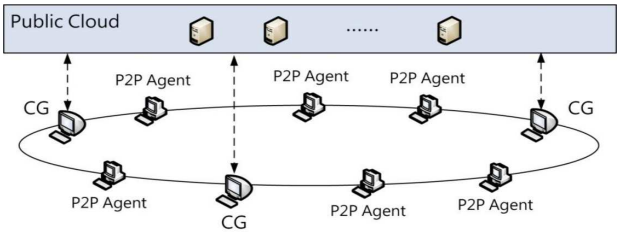

Figure 2. Chord networks with cloud-assistance.

II. System Model based on Cloud Computing

The proposed locality-aware Chord networks are designed on the basis of our previous work. In [19], we have implemented a locality-aware unstructured P2P network using a cloud-assisted platform. A free database GeoLiteCity is built on Hadoop [20] cloud system for mapping IP addresses onto geographical positions referred to as locality information of peers. On that platform, two kinds of peers are implemented, namely P2P agent and Cloud Gateway (CG) node. The CG node with more computing power and network bandwidth can be the interface between peers and the cloud system. A P2P agent, which is a general peer, can request the CG node to upload its IP address to the cloud system. The geographical position of a peer can be gained by a query of GeoLiteCity using its IP address and then maintained in the cloud system.

In this paper, we focus on improving Chord networks based on the cloud-assisted platform. The system architecture of the proposed locality-aware Chord networks is shown in Figure 2.

Due to the maintenance of geographical positions of all peers, peer can select the nearest resource owner instead of random selection from searching result to download. This is because the nearest resource owner can be found by computing and sorting the physical distance between the requesting peer and owners of resource in our cloud-assisted platform.

Besides, the bidirectional searching algorithm proposed in [21] is adopted in our proposed Chord networks. According to the ID of a resource, a peer can search the owners of a resource in clockwise or anti-clockwise direction on Chord networks from finger tables [21] maintained by peers. The logical hop count of searching path can be reduced on average.

III. Locality-Aware Chord Networks

Considering the locality information of peers, four schemes of ID assignments of peers in the proposed locality-aware Chord networks are designed as follows.

A. Longitude Positioning with ID Exchange

The simplest way of arranging peers on a ring is to map peers onto the equator, which is a natural ring on earth. According to the longitude degrees of peers, peers can be projected onto the equator to form a clockwise ring from East to West longitude. Figure 3 shows an example of a Chord network constructed by four peers and three default CG nodes on the equator.

 ISBN: 978-1-61208-583-8

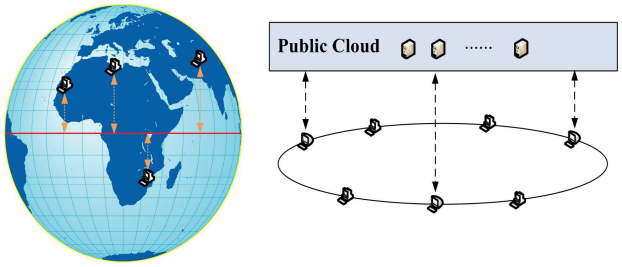

Figure 3. Longitude positioning Chord networks.

Figure 4. ID assignment of global positioning hash.

In our proposed cloud-assisted platform, the geographical positions of all peers are maintained in the cloud system. The ring of Chord networks can be constructed by the order of longitude degrees of peers. Since the original IDs of peers are generated by a hash function, the increasing order of IDs of peers might not be consistent with it of longitude degrees of peers. However, the consistency problem can be solved by exchanging IDs of peers.

Now, we describe the scheme of Longitude Positioning with ID Exchange (LPIDE) as follows. When a new joining peer is coming to the system, the predecessor and successor of the peer on a ring onto the equator can be easily computed by the cloud system. Simultaneously, the increasing order of IDs in the Chord network can be also verified. Once the increasing order of IDs is violated, the cloud system will inform the related peers to exchange IDs for solving the consistency problem. Two peers can exchange their IDs by exchanging their predecessors, successors, and finger tables in constant time.

B. Global Positioning Hash

The scheme of longitude positioning with ID exchange can not distinguish peers with the same longitude degrees. As a result, these peers could be neighbors on a ring mapping on the equator whereas some distance may exist among them. In addition, exchanging IDs of peers may often incur extra overhead.

To improve the lack described as above, a new scheme referred to as Global Positioning Hash (GPH) is proposed. It provides a new ID assignment of peers according to the geographical positions of peers such that the arrangement of peers on a ring approximately close in their physical positions.

The new ID of a peer is composed of two parts as shown in Figure 4. The second part (i.e., low significant bits) is generated by a hash function of IP address, which is the same as in the original Chord networks. The second part makes the ID of peers unique, even peers have the same geographical positions. The first part (i.e., most significant bits), denoted by ID(Global Position), is defined by the function of geographical position including longitude and latitude degrees, as shown in equation(1).

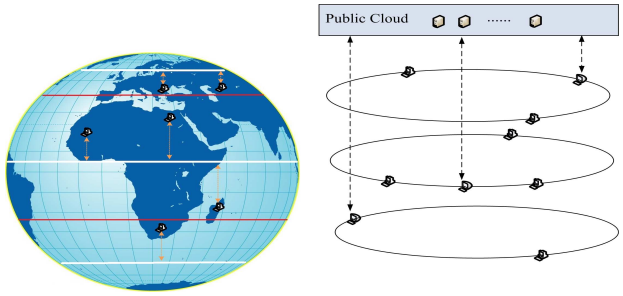

Figure 5. Triple rings Chord networks.

$$ID(Global\ Position) = \lceil (a + 180) * 200 + (b + 90) \rceil \quad (1)$$

where a is longitude degrees and b is latitude degrees.

The first part of the ID decides the order of peers on the ring of Chord networks. The idea of equation (1) is to map geographical positions of peers onto a first quadrant in two-dimension plane, and then execute a linear transformation by sum of the scaled longitude degrees and latitude degrees. This makes sure ID(Global Position) is a positive and an unique integer. The equation (1) reflects relative locations of peers. According to the IDs generated by the GPH scheme, the arrangement of peers on a ring could be approximately close in their physical positions. Unnecessary triangle routing of resource searching could be reduced.

C. Triple rings with Chord

Peers could be from anywhere in the world. The searching performance may suffer in Chord networks with a single ring due to the distribution of peers in physical location. The scheme of Triple Rings with Chord (TRC) is designed to distribute load from a single ring to the three rings of Chord networks.

Peers join one of the three rings in Chord networks according to their location. Considering most of Earth's land is located on Northern Hemisphere, the three rings are planed as shown in Figure 5. The first ring is formed by the peers at above (or equal to) $23.5°N$, the second ring is constructed by the peers at between $23.5°N$ and $23.5°S$, and the peers at under (or equal to) $23.5°S$ can join the third ring.

Each joining peer can upload its IP address to the cloud system by a random CG node to find which ring it has to be joining. In this scheme, each ring is constructed in the same way as it is in the original Chord networks. To improve the searching performance, the way of publishing a new resource is adapted. When a new resource is published on one of the three rings, it would be published on the other two rings by the assistance of a random CG node.

D. Triple rings with global positioning hash

The scheme of Triple Rings with Global Positioning Hash (TRGPH) is a hybrid scheme of TRC and GPH schemes. Peers can join one of the three rings according to their physical locations but each of ring is constructed by the scheme of global positioning hash.

 ISBN: 978-1-61208-583-8

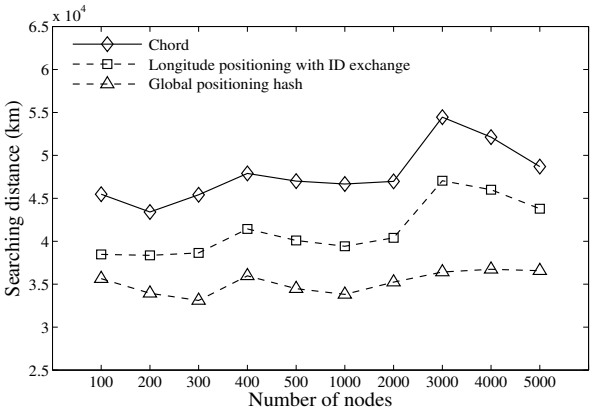

Figure 6. Searching distance for the two proposed schemes.

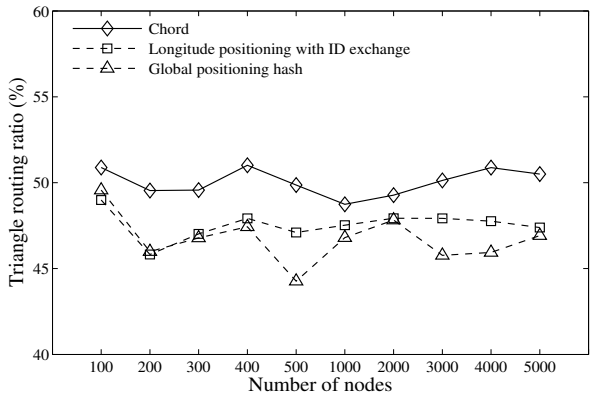

Figure 7. Ratio of triangle routing.

IV. PERFORMANCE EVALUATION

In this section, simulations are performed to study the performance of our proposed locality-aware Chord networks.

A. Simulation Environment

To simulate Internet topology, the topology generator IGen [22] is used to generate random graphs mapped onto real Earth and modeled by Waxman [23]. Nodes are only distributed on Earth's land and the distance between two connected peers is defined by their geographical distance. The simulation programs are written in Java language.

In the system initialization, there are eight kinds of file resources to be shared and each of them is owned by three different random peers in the system. Peers may join or leave the P2P system in our experiment. The times of peers joining the P2P system form a Poisson process. The mean time of a peer joining the system is 10 minutes. The lifetimes of peers in the P2P system form an exponential distribution. The mean lifetime of peers is two hours. For each data point in our experiment, 10 random graphs are generated and 5000 queries for requesting a random resource from eight kinds of files are performed in each graph. The times of queries from random peers in the P2P system form a Poisson process. The mean time of a query happened is twenty minutes.

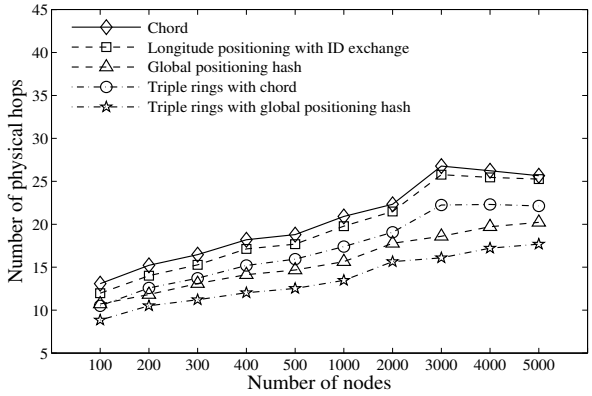

Figure 8. Physical hops of searching cost for all of proposed schemes.

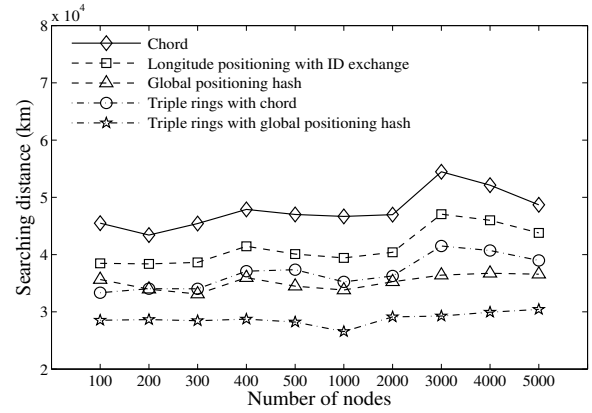

Figure 9. Searching distance for all of proposed schemes.

B. Simulation Results

The performance of Chord networks can be evaluated by the cost of resource searching and downloading.

During the searching process, a query message is forwarded on a logical ring network to find out where the owner(s) of a resource is(are), and then the response message including searching result will be returned. Due to the consistency problem, long distance and more than one routers may exist between two neighboring peers. In our simulation, the physical distance and number of routers that the query and response messages are traveling during the searching process are referred to as the searching distance and physical hops, respectively.

Figure 6 shows the average searching distance in our first two proposed schemes for different number of nodes in the networks. Compared to the original Chord system, the searching distance of our proposed two schemes are shorter than that of the original Chord system. This is because the locality information is considered, and then unnecessary searching routes could be reduced.

The ratio of triangle routing is calculated to evaluate how congruent the overlay network is with and physical network topology. For each query message, every three contiguous peers along its routing path, the triangle routing event is verified and statistically counted to compute the ratio of triangle routing. Figure 7 shows the GPH scheme has lower ratio of

 ISBN: 978-1-61208-583-8

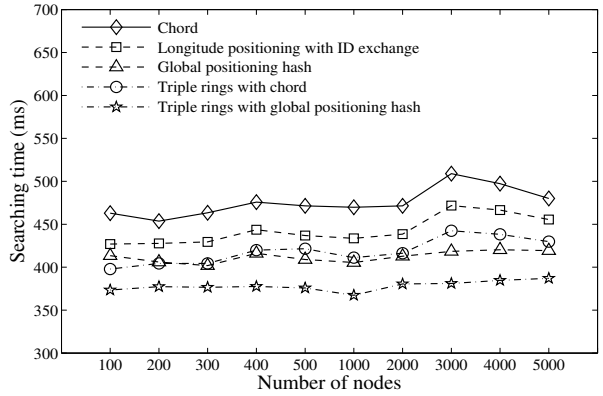

Figure 10. Searching time for all of proposed schemes.

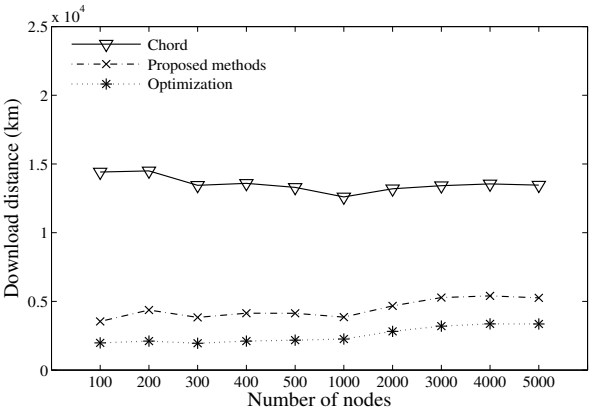

Figure 12. Download distance for the proposed methods.

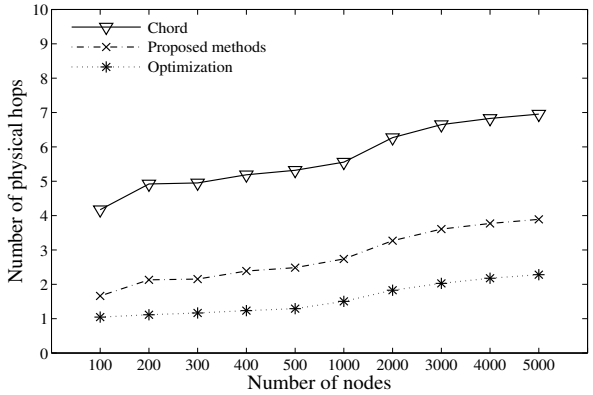

Figure 11. Physical hops of download cost.

triangle routing than the other two schemes in most situations. This phenomenon also verifies that the searching distance for GPH scheme is shorter than that of LPIDE scheme. It is worth noting that the ratio of triangle routing for global positioning hash is still larger than 45%.

Considering the distribution of peers, triple rings are designed in our proposed Chord networks. Figure 8 and Figure 9 show the average number of physical hops and distance respectively for all of the proposed schemes in different number of nodes. From these two figures, we can observe that the performance of GPH scheme is still better than that of TRC scheme. Since the way of constructing logical rings in TRC scheme is still the same as it in the original Chord system, the improvement of consistency problem is limited. These two figures also show the performance of TRGPH scheme is better than that of the others. This is because load can be distributed into different rings and simultaneously the order of peers on logical rings is decided by their geographical positions.

The time necessary for the searching process can be defined as the propagation delay plus the queuing delay when the transmission time of messages is ignored. The propagation delay can be computed by the searching distance divided by the signal speed, $2 * 10^8\ km/s$. In our simulation, the queuing delay of messages elapsed at the routers on the Internet is modeled by an exponential distribution and the mean time is $240\ ms$. Figure 10 shows the average searching time for all of the proposed methods in different number of nodes. The simulation results are similar with those measured by the metrics of physical hops and distance. The average searching time for TRGPH is less than $120\ ms$ compared to the original Chord system when the number of nodes is 3000 in the networks. In summary, the performance of TRGPH scheme is better than the others.

From Figures 6, 8, 9 and 10, we can observe the performance increases to its worse maximum at 3000 nodes and then it decreases. This phenomenon is the effect of number of on-line peers and the distribution of peers locations in network topology. Before the end of simulation, almost all of nodes are on-line when the number of nodes in the networks is less than 3000, while more peers are off-line when number of nodes in the networks is starting from 3000 to 5000. Peers need to take more cost to find their required resources when the number of off-line peers suddenly increases. From the Figure 8, we can see that the difference of physical hops between 2000 nodes and 3000 nodes is more than 4 but the difference of physical hops for 3000 to 5000 nodes is less than 1 for the original Chord system. The minor difference for the later is due to the distribution of peers locations in network topology.

From the searching result, a peer could directly get the resource from the owners without the help of overlay networks. The cost of downloading can be defined as the physical hop count, or distance between the requesting peer to the resource owner. In our proposed Chord networks, peer can find the resource owner which is the nearest physical distance from it by cloud-assisted platform. In our simulation, the cost of downloading is evaluated by selecting the nearest peer to get a resource instead of randomly selecting from the searching result.

Peers can find the nearest resource owner by cloud assistance whenever the ways of ID assignment are adopted in our proposed schemes. In our simulation, the performance of downloading is the same for all of the proposed methods except the original Chord system. It is worth noting that the nearest resource owner is defined by the minimum geographical distance between the requesting peer and the resource owner instead of the shortest path between them in the networks. The performance value measured by such a peer with shortest path from the requesting peer is viewed as the optimal value in our simulation.

Figure 11, Figure 12, and Figure 13 show the physical hops,

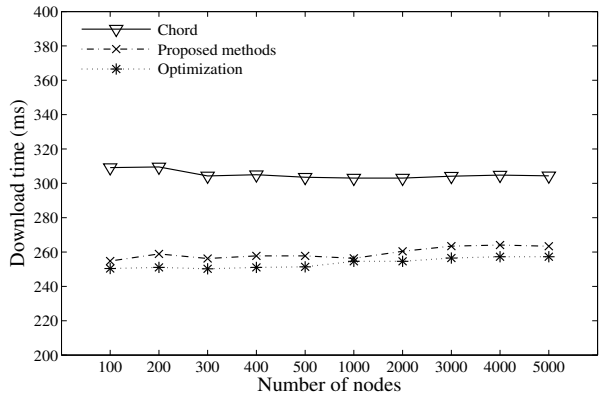

Figure 13. Download time for the proposed methods.

distances, and download time respectively for our proposed methods and the original Chord system compared to the optimal solution in different number of nodes in the networks. From these figures, we can see that the performance of our proposed methods is close to the optimal solution and better than the original Chord system.

V. Conclusions and Future Work

In this paper, locality-aware Chord P2P networks based on cloud-assistance are proposed. Four new schemes of ID assignments associated with geographical positions of peers have been designed. The arrangement of peers on the rings constructed by our proposed methods is approximately close in their physical positions. The ratio of unnecessary triangle routing in the searching process can be reduced. In addition, a new download method which can select the nearest nodes to get resources has also been designed. The simulation experiments show the TRGPH scheme has better performance than the others.

The ID assignments of our proposed schemes only consider the physical locations of peers but the free database GeoLiteCity could not provide precise enough physical locations of peers. In the future, the other IP-geolocation mapping schemes, such as in [25]-[26], will be considered to design the schemes of ID assignments to improve the performance. In addition, the real life networks from the data sets [27] will be adopted in our simulation experiments to compare the performance affected by network topologies and distribution of peers locations.

Acknowledgment

This study was supported by the Ministry of Science and Technology of Taiwan, under the Grant No. MOST 105-2221-E-216-010 E-216-017.

References

[1] D. Harrison, BitTorrent homepage, http://www.bittorrent.org/, retrieved: May, 2017.

[2] PPLive Inc., PPLive homepage, http://www.pplive.com/, retrieved: May, 2017.

[3] PPStream Inc., PPStream homepage, http://www.ppstream.com/, retrieved: May, 2017.

[4] H. Wang, Y. Zhu, and Y. Hu, "To Unify Structure and Unstructured P2P Systems," In Proceeding of the 19th International Parallel and Distributed Processing Symposium (IPDPS05), Denver, Colorado, April 2005, pp. 1-10.

[5] Napster, http://www.napster.com, retrieved: May, 2017.

[6] Gnutella, http://www.gnutella.com, retrieved: May, 2017.

[7] I. Stoica et al., "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," IEEE/ACM Transactions on Networking, Vol. 11, No. 1, February 2003, pp. 17-32.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content Addressable Network," In Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM), San Diego, CA, 2001, pp. 161-172.

[9] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," Technical Report UCB/CSD-01-1141, UC Berkeley, Apr. 2001, pp. 1-27.

[10] H. J. Wang and Y. T. Lin "Cone: A Topology-Aware Structured P2P System with Proximity Neighbor Selection" Future generation communication and networking (FGCN 2007), Dec. 2007, pp. 43-49.

[11] W. Y. Wu et al., "LDHT: Locality-aware Distributed Hash Tables, "International Conference on Information Networking(ICOIN 2008), Jan. 2008, pp. 1-5.

[12] T. Y. Chung, Y. H. Chang, C. Y. Liu, and Y. M. Chen, "SSTIF: Stable Self-Organized Infrastructure-based Peer-to-Peer Network" The 9th International Conference on Advanced Communication Technology (ICACT 2007), Feb. 2007, pp. 1441-1446.

[13] F. Hartmann and B. Heep, "Coordinate-based Routing: Refining NodeIds in Structured Peer-to-Peer Systems," IEEE Proceedings of the International Conference on Ultra Modern Telecommunications Workshops(ICUMT 2009), St. Petersburg, Russia, October 2009, pp. 1-6.

[14] H. Hoang-Van, K. Mizutani, T. Miyoshi, and O. Fourmaux, "P2P traffic localization by forcing packet loss, 2013 IEEE/ACIS 12th Int. Conf. Comput. Inf. Sci. ICIS 2013 - Proc., 2013, pp. 323-328.

[15] A. H. Payberah, H. Kavalionak, V. Kumaresan, A. Montresor, and S. Haridi, "CLive: Cloud-Assisted P2P Live Streaming," IEEE 12th International Conference on Peer-to-Peer Computing, 2012, pp. 79-90.

[16] K. Graffi et al., "Towards a P2P Cloud: Reliable Resource Reservations in Unreliable P2P Systems," in 16th International Conference on Parallel and Distributed Systems, 2010, pp. 27-34.

[17] J. Dharanipragada and H. Haridas, "Stabilizing peer-to-peer systems using public cloud: A case study of peer-to-peer search," in 2012 11th International Symposium on Parallel and Distributed Computing, pp. 135-142.

[18] Maxmind, https://www.maxmind.com/en/geoip2-databases, retrieved: May, 2017.

[19] C. H. Wang, C. H. Kuo, and Y. C. Chern, "Locality-Aware P2P Networks Based on Cloud-Assistance, The 8th International Conference on Ubi-Media Computing (UMEDIA 2015), PEWiN workshop (PEWIN 2015), pp. 38-43.

[20] Hadoop, http://hadoop.apache.org/, retrieved: May, 2017.

[21] J. Wang, S. Yang, and L. Guo, "A Bidirectional Query Chord System Based on Latency-Sensitivity, 2006 Fifth Int. Conf. Grid Coop. Comput., 2006, pp. 164-167.

[22] IGen, http://informatique.umons.ac.be/networks/igen/, retrieved: May, 2017.

[23] B. M. Waxman, "Routing of Multipoint Connections, IEEE J. Sel. Areas Commun., vol. 6, no. 9, 1988, pp. 1617-1622.

[24] Thrift, https://thrift.apache.org/ ,retrieved: May, 2017.

[25] B. Wong and I. Stoyanoy, "Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts, In Proceedings of NSDI07, 2007, pp. 313-326.

[26] D. Li et al., "IP-Geolocation Mapping for Moderately Connected Internet Regions," in IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 2, Feb. 2013, pp. 381-391.

[27] https://github.com/uofa-rzhu3/NetLatency-Data/ retrieved: May, 2017.

 ISBN: 978-1-61208-583-8

Internet of Vehicles Functional Architectures - Comparative Critical Study

Eugen Borcoci, Serban Obreja, Marius Vochin
University POLITEHNICA of Bucharest - UPB
Bucharest, Romania
Emails: eugen.borcoci@elcom.pub.ro, serban@radio.pub.ro, mvochin@elcom.pub.ro

***Abstract* — The continuous growth of the vehicles number, together with associated problems encountered in transportation systems have driven significant developments in the framework of Intelligent Transport System (ITS). Recently, an advanced solution - Internet of Vehicles (IoV) is proposed, seen as a part of Future Internet and specifically of Internet of Things (IoT), aiming to offer novel advanced commercial and technical capabilities. IoV will integrate the previous Vehicular Ad Hoc Networks (VANET) and also functionalities already developed in ITS. However, the architectural aspects of the IoV are still open research issues. This paper attempts a comparative critical study of several functional architectures proposed for IoV, including recent ones based on Cloud/Fog computing and Software defined networking (SDN) - control.**

Keywords* — *Internet of Vehicles, VANET; Cloud computing; Fog computing; Software Defined Networking; Network Function Virtualization.

I. INTRODUCTION

The vehicular communications have been intensively studied, designed, standardized and implemented in the last two decades. The umbrella and framework for such developments is the *Intelligent Transport System* (ITS) [1][2]. Associated technologies are *Dedicated Short-Range Communications* (DSRC) and *Wireless Access in Vehicular Environments* (WAVE) [2]. IEEE 802.11p and IEEE 1609 represent a set of standards for DSRC/WAVE networks.

Vehicular Ad Hoc Networks (VANET) [1] have been defined to support basic vehicular communications: vehicle to vehicle (V2V), vehicle to road (V2R), or vehicle to Infrastructure (V2I). The initial VANET components are the *On-Board-Unit* (OBU) placed in the vehicles and *Road-Side–Unit* (RSU) placed on the roads. The RSUs can inter-communicate and also could be linked to external networks like Internet. The main applications of VANET have been oriented to safety and traffic management applications.

The VANETs have several limitations related to their pure ad hoc network architecture (in V2V case), unreliable Internet service, incompatibility with personal devices, non-cooperation with cloud computing, low accuracy of the services, operational network dependency and restricted areas of applications and services. Therefore, extending the VANET architecture is considered today as a strong need.

Recently, *Internet of Vehicles* (IoV) has been proposed as a significant enhancement in vehicular communication area. It could be seen as a global span of a vehicle network [3][4]. The IoV is considered as a special case of *Internet of Things* [5][6], where the "things" are either vehicles or their subsystems. The IoV will connect the vehicles and RSUs through different *Wireless/Radio Access Technologies* (WAT/RAT), while traditional Internet and other heterogeneous networks will be used for wide area. The IoV objectives can include the traditional VANET services but also novel ones, e.g., vehicle traffic management in urban or country areas, automobile production, repair and vehicle insurance, road infrastructure construction and repair, logistics and transportation, etc. The IoV can be supported by recent technologies like centralized *Cloud Computing* (CC) combined with *Fog* or *Edge Computing* [7]; the latter can offer a better time response, more flexibility and degree of functional distribution, which are more appropriate for vehicular world.

In terms of management and control, *Software-defined networking* (SDN) [8] can offer to IoV its centralized up-to-date logical view upon the network, programmability, facilitating a flexible network management and on-the-fly modification of the network elements behavior. *Network Function Virtualization* (NFV) [9] can add flexibility by virtualizing many network functions and deploying them into software packages. Dedicated *Virtualized Network Functions* (VNF) can be defined, then dynamically created/destroyed, assembled and chained to implement legacy or novel services. NFV can cooperate with SDN to realize new flexible and powerful IoV architectures.

The large communities of users/terminal devices in IoV need powerful and scalable *Radio Access Technologies* (RAT). The 4G and the emergent 5G, based on cloud computing architectures (*Cloud Radio Access Network*-CRAN) are significant candidates for constructing the IoV access infrastructure [10].

Despite IoV promises high capabilities, there still exist many challenges, both in conceptual and architectural aspects and also from implementation and deployment point of view. Many IoV advanced features and integration with the above technologies (CC, Fog, SDN, NFV) are still open research issues.

This paper attempts a comparative critical study of several functional architectures proposed for IoV, including recent ones based on Cloud/Fog computing and Software

 ISBN: 978-1-61208-583-8

defined networking (SDN) - control. An enriched functional architecture with Fog computing and SDN control is proposed.

The paper is organized as follows. Section II is an overview of related work with a critical presentation of some IoV generic architectures. Section III revisits the SDN-based architectures of IoV. Section IV proposes an enriched integrated architecture, Fog-SDN oriented. Section V presents conclusions and future work.

II. IoV GENERIC LAYERED ARCHITECTURES EXAMPLES

IoV is usually seen as a part of the more general Internet of Things (IoT), so it is of interest to evaluate how the proposed IoV architectures are generally consistent with IoT architecture.

Al-Fuqaha et al. [5] present an overview of IoT, identifying the Iot elements, i.e., *identification, sensing, communication, computation, services and semantics.* Several variants of IoT layered architectures are presented, where the most comprehensive has 5-layers: *Objects (perception) (OL), Object Abstraction (OAL), Service Management (SML), Application (AL)* and *Business (BL)* layer. These layers are different from those of the classical TCP/IP architectures, but the layering principles are still preserved.

The *Object* layer represents IoT physical sensors and actuators, performing functionalities such as querying location, temperature, weight, motion, vibration, acceleration, humidity, etc. The digitized data are transferred to the OAL through secure channels.

The *Object Abstraction* layer transfers data to the SML through secure channels. Layer 2 networking transfer functions are included here, based on technologies like RFID, GSM, 3G, 4G, UMTS, WiFi, Bluetooth Low Energy, infrared, ZigBee, etc. Additionally, cloud computing functions and data management processes are handled at this layer. The *Service Management* layer plays a middleware role, by pairing a service with its requester based on addresses and names. The SML supports IoT application programmers to work with abstracted heterogeneous objects. It also processes received data, takes decisions, and delivers the required services over the network wire protocols. The *Application* layer provides to the customers the requested services (with appropriate quality). The AL covers different vertical markets (e.g., smart home, smart building, transportation, industrial automation and health care, etc.). The *Business Layer* manages all IoT system activities and services. Using data provided by AL, it creates a business model, graphs, flowcharts, etc.; it is related to design, analysis, implementation, evaluation, monitoring and management (of the lower layers), and developing IoT system related elements. Decisions can be taken following Big Data analysis. Security features are included. Note that the architecture described above is a high level view only; further structuring can be made and mapping on various existing protocols [5].

For IoV, several architectures are recently proposed and discussed. A short critical overview and comparison is exposed below.

Bonomi et al. [6] proposed a four - layered architecture for connected vehicles and transportation. The layers are also called *"IoT key verticals",* suggesting that a given layer includes not only classical layer functions (i.e., L1, L2,…) but rather groups of functions, which could be mapped on several classical layers. Also, the four layers are rather corresponding to different geo-locations of the subsystems (vehicles, networking infrastructure, cloud data centers, etc.) The bottom layer (*end points*) represents the vehicles, plus their communication protocols (basically for V2V communication, using the IEEE 802.11p). The layer two (*infrastructure*), represents communication technologies to interconnect the IoV actors (via WiFi, 802.11p, 3G/4G, etc.). The third layer (*operation*) performs management actions; it verifies and ensures compliance with all applicable policies, to regulate the information management and flow. The fourth layer is called *cloud* (public, private or enterprise) based on a defined profile coupled with the possibility of receiving services (voice, enterprise video and data) on demand. Note that this architectural view is a mixed one, and does not clearly separate the sets of functions of different levels.

Kayvartya et al. [4] have proposed an IoV five layer architecture, to support an enriched set of vehicular communications, in addition to traditional V2V, V2R/V2I, i.e., *Vehicle-to-Personal* devices (V2P) and *Vehicle-to-Sensors* (V2S). Each particular IoV communication type can be enabled using a different WAT, e.g., IEEE WAVE for V2V and V2R, Wi-Fi and 4G/LTE for V2I, CarPlay/NCF (*Near Field Communications*) for V2P and WiFi for V2S. The system includes vehicles and *Road Side Units* (RSU), but also other communication devices. Embedding such a large range of devices makes IoV more complex, (compared to VANET), but more powerful and market oriented.

Three architectural planes are defined: *management*, *operation* and *security*. The network model is composed of three functional entities: *client, connection* and *cloud.* The layers are (see Figure 1): *perception, coordination, artificial intelligence, application* and *business.*

The *perception* layer (PL) functions generally correspond to those of the traditional physical layer. The PL is instantiated by *sensors* and *actuators* attached to vehicles, RSUs, smart-phones and other personal devices. Its main task is to gather information on vehicle, traffic environment and devices (including movement –related parameters).

The *coordination* layer (CL) represents a virtual universal network coordination entity for heterogeneous network technologies (WAVE, Wi-Fi, 4G/LTE, satellites). While the basic job is transportation, some other processing tasks are added, of information received from heterogeneous networks with aim to create a unified structure with identification capabilities for each type of network.

 ISBN: 978-1-61208-583-8

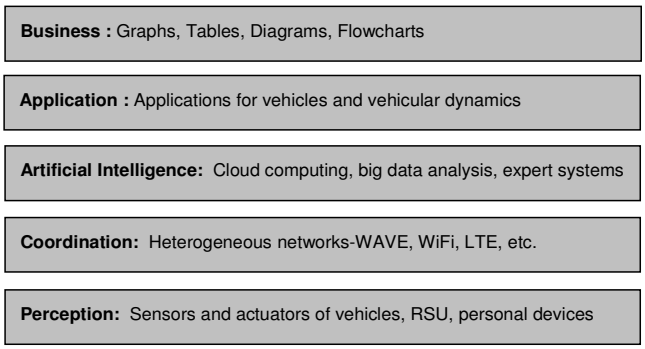

Figure 1. Five layer IoV architecture [4].

The *artificial intelligence* layer (AIL) is represented by a generic virtual cloud infrastructure, working as an information management centre. It stores, processes and analyzes the information received from the lower layer and then takes decisions. Its major components are: *Vehicular Cloud Computing (VCC), Big Data Analysis (BDA)* and *Expert System.* The AIL should meet the requirement of applications and services working on top of it.

The *application* layer (AL) contains smart applications (e.g., for traffic safety and efficiency, multimedia-based infotainment and web based utility). The AL include safety and efficiency applications (VANET legacy) and provides smart services to End Users (EU) based on intelligent analysis done by AIL. The AL efficiently discovers the services provided by AIL and manage their combinations. It also provides EU application usage data to the business layer. Currently, it is recognized that these smart applications constitute a major driving force to further develop IoV.

The *business* layer (BL) includes IoV operational management functions, basically related to business aspects: to foresight strategies for the development of business models based on the application usage data and statistical analysis of the data; analysis tools including graphs, flowcharts, comparison tables, use case diagrams, etc.; decision making - related to economic investment and usage of resources; pricing, overall budget preparation for operation and management; aggregate data management.

The architecture is split in three parallel planes: *operation, management and security.* The work [4] also proposed a possible mapping between the five layers and different protocols already developed in vehicular communications by ITS, VANET, IEEE, etc. The *operation* plane contains actually traditional *data* plane functions but still has some control and management role.

At *perception* layer, current technologies can be used for access in ITS and VANET (see Figure 2). However, the CL includes not only TCP/IP transport and network protocols but also different solutions (with no IP usage). Examples are: IEEE 1609.4 along with a *Global Handoff Manager* (GHM-open research) and other protocols proposed at network layer in projects like CALM, WAVE. For instance, in the stack there exist WSMP - Short Message Protocol and FAST -Fast Application and Communication Enabler.

In AIL, cloud capabilities are seen as major contributors, working on top of lower sub-layer: CALM Service Layer (CALM-SL) and WAVE-1609.6 service related protocols. The upper sub-layer consists in Vehicular Cloud Computing (VCC) and Big Data Analysis (BDA) related protocols. They can offer cloud services of type "*X as a Service*": Storage (STaaS), Infrastructure (INaaS), Network (NaaS); Cooperation (CaaS), Entertainment (ENaaS), Gateway (GaaS); Picture (PICaaS) and Computing (COMaaS).

Still further research work is necessary, given the current unavailability of enough suitable protocols for VCC and BDA. Another open issue is that VANETs projects, generally, do not have clear definitions of the upper sub-layer, while some IoT projects are recently working towards these.

The *Application Layer* (AL) includes two sets of applications: *Smart Safety and Efficiency* (SSE) and *Smart Business Oriented* (SBO). The current WAVE resource handler protocol 1609.1 can be used on the top of these applications, to manage the resources among smart applications. The *Business Layer* (BL) in [4] proposes various business models like Insurance (INS), Sale (SAL), Service (SER) and Advertisement (ADV). The set of these functionalities could be further enriched in the future.

The architecture has the merit that integrates in the management and security planes some existing functional blocks and protocols (see Figure 2), already developed in WAVE (P1609.x), CALM and C2C projects.

However the mentioned 5-layer architecture does not touch some important and recent aspects in developing IoV architecture, e.g., how to distribute computation intelligence between a central cloud and fog units (which are placed at the network edge) while cloud-fog combination seem to be an efficient and attractive solution for a distributed system like IoV. Also, SDN-like control possibilities are not discussed in this architecture.

Contreras-Castillo et al. [11] propose a seven layer architecture, supporting the functionalities, interactions, representations and information exchanges among all the devices inside a IoV ecosystem. The authors claim that this architecture (having more than five layers) has as objective to reduce the complexity of each layer and better standardize the interfaces and protocols used in each layer. The interaction model considers the following entities which can communicate to each other: vehicle (V), person (P), personal device (P), network infrastructure (I), sensors (S), any device (D) and roadside device (R). Consequently, the communications might be of type V2V, V2R, V2I, V2D, V2P, V2S, D2D.

 ISBN: 978-1-61208-583-8

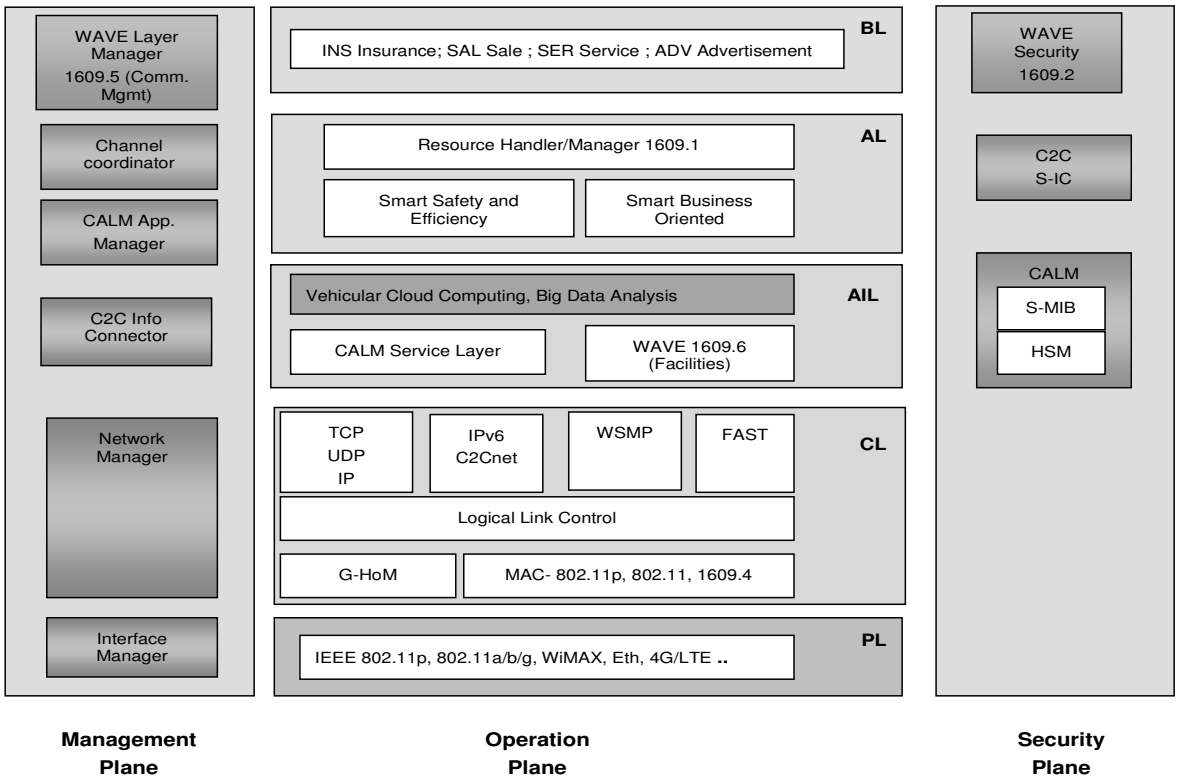

Figure 2. Five layer IoV architecture mapped on particular protocols [4]

PL Perception layer; CL Coordination layer; AIL Artificial Intelligence layer; AL Application layer; BL Business layer; C2C Car to Car; CALM Communication Architecture for Land Mobile; DSRC Dedicated Short Range Communication; WAVE Wireless Access in Vehicular Environment; FAST Fast Application and Communication Enabler; LLC Logical Link Control; G-HoM Global Handoff Manager; WSMP WAVE Short Messages Protocol; BDA Big Data Analysis; VCC Vehicular Cloud Computing; SSE Smart Safety and Efficiency; SBO Smart Business Oriented; INS Insurance; SAL Sale ; SER Service ; ADV Advertisement; HSM Hardware Security Manager; S-IC Security Information Connector; S-MIB Security Management Information Base

The network model should support collaboration between multi-users, multi-vehicles, multi-devices (sensors, actuators, mobile devices, access points), multi-communication models (point to point, multi-point, broadcast, geo-cast) and multi-networks (wireless or wire networks with various technologies like WiFi, Bluetooth, WiMAX, 3G, 4G/LTE, etc.).

The layers defined in [11] are (bottom-up list): *User interaction, Data acquisition, Data filtering and pre-processing, Communication, Control and management, Business and Security.*

Note that this "layered" architecture actually does not follow the principles of a layered stack architecture (where each layer traditionally offers some services to the above one). For instance, the *Control and management layer* and *Security* layer seem to be rather architectural "planes "and not traditional layers; they have to interact with all other five layers. No notion of an architectural "plane" is explicitly defined in [11].

The *User interaction* layer contains in-vehicle computing systems including: a. information-based systems to provide information (e.g., on routes, traffic conditions, car parking availability and warning/advice regarding risks) to components of the driving environment, the vehicle or the driver; b. control-based systems to monitor changes in driving habits and experiences and operational elements of the driving task (e.g., adaptive cruise control, speed control, lane keeping and collision avoidance). It is stated in [11] that designing user interfaces for in-vehicle systems is still posing many new research challenges. Note that this "layer" actually contains functions of several layers defined in other architectures (e.g., some structured in a similar way as classic TCP/IP stack).

The *Data acquisition* layer has tasks covering all three traditional planes (data, control and management). Apparently it overlaps functions of "networking" Layer 2. It gathers data (for safety, traffic information, infotainment), from a given area of interest, from all the sources (vehicle's internal sensors, GPS, inter-vehicle communication, Wireless Sensor Networks (WSN), or devices such as cellular phones, sensors and actuators, traffic lights and road signals located on streets and highways. Intra- and inter vehicular interactions are within the scope of this layer. Various access technologies and associated protocols are

supposed to perform the tasks. For intra-vehicle communication, the proposals are: Bluetooth (2.4 GHz) , ZigBee(868 MHz, 915 MHz and 2.4 GHz) Wi-Fi HaLow (900 MHz) Ultra-wideband (3.1–10.6 GHz), with data rates up to 480 Mbps and coverage distances up 1000m. For inter-vehicles communication technologies can be: IEEE WAVE/DSRC with IEEE 802.11p for PHY and MAC layers and the IEEE 1609 family for upper layers; 4G/LTE (1700 and 2100 MHz).

The *Data filtering* and *pre-processing* layer is necessary, given that IoV, may generate huge amounts of data, not all being relevant for all entities. This layer analyses and filters the collected information, to avoid the dissemination of irrelevant information and reduce the network traffic. Examples of protocols to be used in this layer are: *Xtensible Messaging and Presentation Protocol* (XMPP), *Constrain Application Protocol* (CoAP), *HTTP Representational State Transfer* (HTTP REST), *Message Queuing Telematic Transport* (MQTT), *Lightweight Local Automation Protocol* (LLAP). Several data filtering approaches are referenced in [11], but novel intelligent and efficient data mining techniques are considered to be necessary.

The *Communication* layer actually performs both data and control function at networking level, given the set of protocols suggested as: 6LoWPAN, IPv4, IPv6, Routing Protocol for Low Power and Lossy Networks (RPL), etc. This layer should select the best network to send the information, based on several selection parameters.

The *Control and management* layer is the global coordinator for managing different network service providers within the IoV environment. Its functions are: to manage the data exchange among the various services; to manage the information generated by devices: in-vehicle or around sensors, roadside infrastructure and user devices in the environment; apply different suitable policies (e.g., traffic management and engineering, packet inspection, etc.)

The *Business* layer processes information using various types of cloud computing infrastructures locally and remotely. Typical functions are: storing, processing and analysing info received from the other layers; making decisions based on data statistical analysis and identifying strategies that help in applying business models based on the usage of data in applications and the statistical analysis. (tools such as graphs, flowchart, critical analysis, etc.). The protocols proposed for this layer are: *CALM Service Layer*, WAVW 1609.6, TR-069, *Open Mobile Alliance Device Management* (OMA-DM).

The *Security* layer (despite of its naming of "layer") is actually an architectural plane which communicates directly with the rest of the layers. It implements security functions (data authentication, integrity, non-repudiation and confidentiality, access control, availability, etc.) to exchange data among sensors, actuators, user's devices through secure networks and service providers. The protocols envisaged are similar to those presented in Figure 2.

The seven layer architecture of [11] does not touch the integration of SDN/NFV approach. The cloud services are located at business level (as vehicular cloud computing) while we believe that a more natural placement could be as in Figure 2, i.e., under application layer. Some mixture of "layers" and "plane" notions is apparent; there is a lack of enough orthogonality of different "layers".

III. SDN CONTROLLED IOV ARCHITECTURES

This section shortly presents related work dedicated to VANET/IoV with SDN control.

Y.Lu et al. [12] applies SDN control to VANET, to get more flexibility, programmability and support for new services. The architectural components are: SDN controller, SDN wireless nodes and SDN- enabled RSUs. The SDN controller is a single entity performing the overall control of the system. The SDN wireless nodes are vehicles, seen as data plane elements (SDN - forwarders). The SDN RSUs are also treated as data plane elements, but hey are stationary. The benefits of the approach are proved by simulation, while considering some specific use cases (e.g., routing). However, a complete layered functional IoV architecture is not discussed.

K.Zeng et al. [13] propose an IoV architecture called *software-defined heterogeneous vehicular network* (SERVICE), based on Cloud-RAN technology, able to support the dynamic nature of heterogeneous VANET functions and various applications. A multi-layer Cloud-RAN multi-domain is introduced, where resources can be exploited as needed for vehicle users. The system is hierarchically organized (there are defined: remote, local and micro clouds) and virtualization (for flexibility) is considered for implementation. The high-level design of the soft-defined HetVNET is presented. The SDN control is organized on two levels (one primary controller and several secondary controllers; each one of the latter controls a given service area). A complete layered functional IoV architecture is not in the paper scope.

A Fog-SDN architecture called FSDN is proposed for advanced VANET by Truong et al. [7], for V2V, V2I and Vehicle-to-Base Station communications. The Fog computing brings more capabilities for delay-sensitive and location-aware services. The SDN components are: *SDN Controller* (it controls the overall network behavior via *OpenFlow* –interfaces; it also plays as Orchestration and Resource Management for the Fog); *SDN Wireless Nodes* (vehicles acting as end-users and forwarding elements, equipped with OBU); *SDN RSU* (it is also a Fog device); *SDN RSU Controller* (RSUC) (controlled by the central SDN controller; each RSUC controls a cluster of RSUs connected to it through broadband connections. The RSUC can forward data, and store local road system information or perform emergency services. From Fog perspective RSUCs are fog devices); *Cellular Base Station* (BS) performing traditional functions (they are SDN-controlled via OpenFlow and can also offer Fog services). This study does not discuss a full functional layered IoV architecture.

Kai et al. [14] present an overview of Fog–SDN solution for VANET and discuss several scenarios and issues. It is shown that a mixed architecture Fog-SDN (similar to that

 ISBN: 978-1-61208-583-8

proposed in [7]) can be powerful and flexible enough, to serve future needs of IoV. Again, we note that this study does not discuss a full functional layered IoV architecture.

Chen et al. [15] discusses an IoV architecture and solutions based on SDN control. However, a full functional layered architecture is not discussed.

IV. SDN-FOG ENABLED IOV FUNCTIONAL ARCHITECTURE

This section proposes a layered functional IoV architecture of a heterogeneous network including SDN control and Fog capabilities. We propose a possible infrastructure (Figure 3), which could be a horizontal extension of that proposed in [7]. The Data plane includes: mobile units (vehicles) equipped with OBUs; advanced RSUs, which could have enough resources (computing, storage) as to play also Fog role (F-RSU), or could be regular RSU like in traditional VANETs; base stations (BS) of type WiMAX/3G/4G-LTE. A fixed network (partial mesh) can interconnect the RSUs. The SDN Data plane contains the forwarding nodes and can be geographically organized in several service areas. The SDN Control plane is organized on two levels: primary SDN controller (P-SDNC) controlling the overall behavior of the network and secondary controllers (S-SDNC), one for each service area. The S-SDNC can also contain the resource management functions of the Fog infrastructure. The P-SDNC is logically connected to each S-SDNC via the Control plane overlay or physical links. The SDN south interfaces between the controllers and the lower level can be supported by OpenFlow protocol or similar. This infrastructure is enough general as to be considered as a candidate or IoV.

Figure 4 shows a proposal to enrich the layered functional architecture introduced in [4], by adding SDN and Fog functionalities, supposing that the infrastructure is that of Figure 3. The second layer is renamed in Network and Transport Layer (NTL), showing in a more explicit way the role of this layer. The Operation Plane is renamed in Control and Data Plane.

The functionalities of the P-SDNC can be embedded in the management plane, given that its role is to govern the overall network behavior (e.g some overall policies can be coordinated by this module). The regional SDN control is placed naturally at NTL level as to control the SDN forwarders and also the functions of Fog nodes located in the access area. Additionally S-SDNC functions can be included in the AIL, to serve this layer needs in terms of Fog AI resource control. The cloud services should be split between centralized Cloud computing and Fog nodes but this is out of scope of this study (it is for further work).

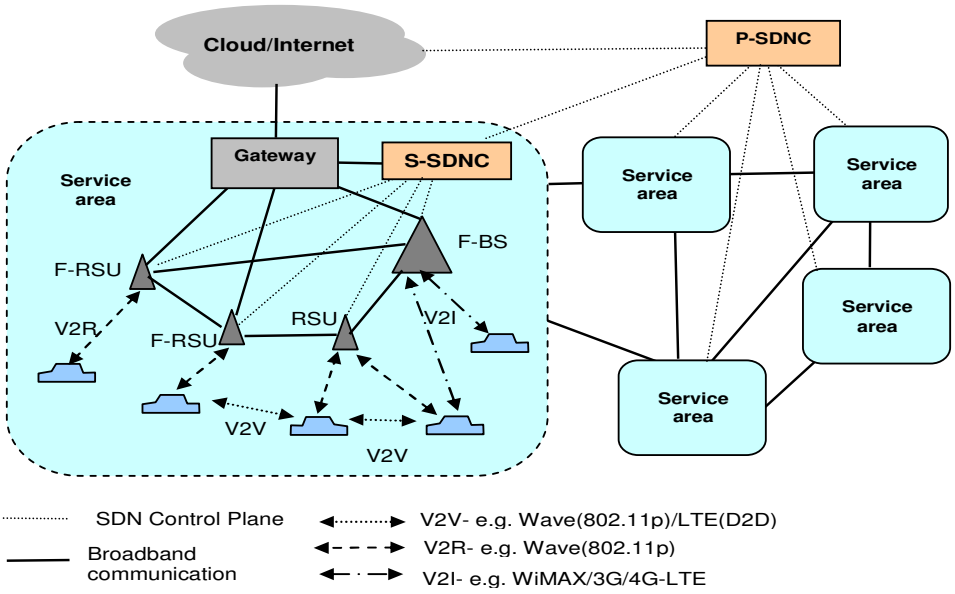

Figure 3. Generic IoV system architecture
F-BS - Fog-capable Base Station; F-RSU Fog-capable Remote Side Unit; P-SDNC- Primary SDN Controller; S-SDNC Secondary-SDN Controller; D2D- device to device communication

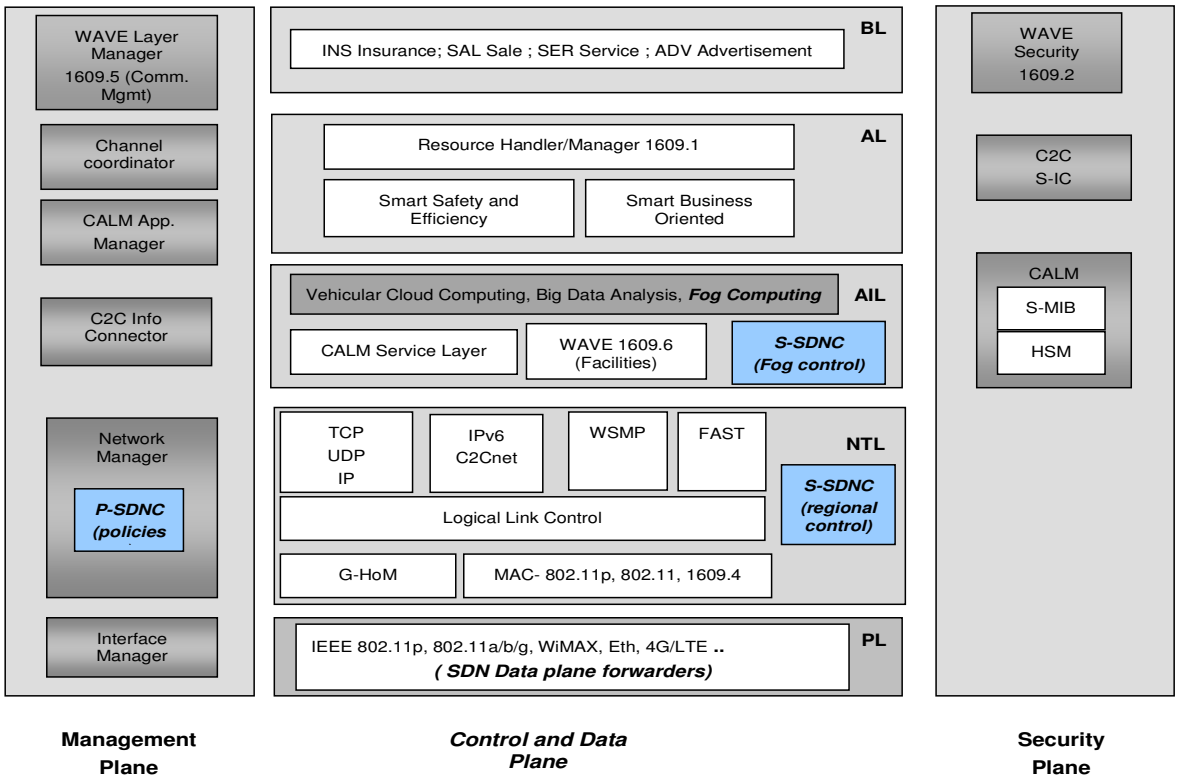

Figure 4. Functional IoV 5-layer architecture enriched with SDN control and Fog computing

V. CONCLUSIONS AND FUTURE WORK

This paper presented a comparative critical view of several IoV architectures proposed in the literature, focused on functional layering aspects. Among several proposals, we selected a five layer multiple-plane architecture, considering this model as a good and orthogonal approach which consistently include the major IoV functionalities and is giving the possibility to clearly define interfaces between layers and planes. The architecture is consistent with IoT architectural vision. In Section IV, a modified Fog-SDN based IoV infrastructure is proposed, where the associated layered architecture is enriched by considering the additional Fog-based approach and SDN distributed control. This work could be a contribution toward an IoV reference architecture.

Future work should be done to allocate and map different functions of the general functional layered architecture to specific entities of a complete IoV system. This should be done based on their different roles and placement: terminals (vehicles), RSUs, Fog Nodes, BS, core network, cloud data centers, etc. The virtualization aspects and their impact on the architecture are not discussed in this study. This is also a subject for further work.

VI. ACKNOWLEDGMENTS

This work has been partially funded by University Politehnica of Bucharest, through the "Excellence Research Grants" Program, UPB – GEX. Identifier: UPB–EXCELENȚĂ–2016 Research project Intelligent Navigation Assistance System, Contract number 101/26.09.2016 (acronym: SIAN).

REFERENCES

[1] S. Sultan, M. Moath Al-Doori, A.H. Al-Bayatti, and H.Zedan "A comprehensive survey on vehicular Ad Hoc Network", J.of Network and Computer Applications, Jan. 2014, https://www.researchgate.net/publication/259520963, [Retrieved: December, 2016].

[2] Y. Li, "An Overview of the DSRC/WAVE Technology", https://www.researchgate.net/publication/288643779, 2012, [Retrieved: May, 2017].

[3] Y. Fangchun, W.Shangguang, L. Jinglin, L. Zhihan, and S.Qibo, "An overview of Internet of Vehicles", China Commun., vol. 11, no. 10, pp. 115, October 2014.

[4] O. Kaiwartya, A.H. Abdullah, Y. Cao, A. Altameem, and M. Prasad, "Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects" IEEE Access, Special Section on Future Networks, Architectures, Protocols and Applications, Vol. 4, pp.5536-5372, September 2016.

[5] A. Al-Fuqaha, M.Guizani, M. Mohammadi, M. Aledhari, and M.Ayyash, "Internet of Things: A Survey on Enabling

 ISBN: 978-1-61208-583-8

Technologies, Protocols, and Applications", IEEE Communications Surveys &Tutorials Vol. 17, No. 4, pp.2347-2376, 2015.

[6] F.Bonomi, R.Milito, J.Zhu, and Sateesh Addepalli, "Fog Computing and Its Role in the Internet of Things", August 2012, http://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf, [Retrieved: January, 2017].

[7] N.N.Truong, G.M.Lee, and Y.Ghamri-Doudane, "Software defined networking-based vehicular ad hoc network with fog Computing", Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM'15), May 2015, Ottawa, Canada. Piscataway, NJ, USA: IEEE, , pp. 1202–1207, 2015.

[8] B. N. Astuto, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks", Communications Surveys and Tutorials, IEEE Communications Society, (IEEE), 16 (3), pp. 1617 – 1634, 2014.

[9] B.Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network Function Virtualisation: Challenges and Opportunities for Innovations". IEEE Communications Magazine, pp. 90-97, February 2015.

[10] M. Peng, Y. Li, J. Jiang, J. Li, and C.Wang, "Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies", IEEE Wireless Communications, pp.126-135, December 2014.

[11] J.C. Contreras-Castillo, et al., "A seven-layered model architecture for Internet of Vehicles",Journal of Information and Telecommunications , Vol. 1, No. 1, pp. 4–22, 2017.

[12] Y.Lu, M.Gerla, R. Gomes, and E. Cerqueira, "Towards software-defined VANET: Architecture and services", MedHocNet.2014.6849111, https://www.researchgate.net/publication/271472780, [Retrieved:April, 2017].

[13] K. Zeng, L. Hou, H. Meng, Q. Zheng, and N. Lu, "Soft-Defined Heterogeneous Vehicular Network: Architecture and Challenges", IEEE Network, vol. 30, pp.72-79, July/August 2016.

[14] K.Kai, W.Cong, and L.Tao, "Fog computing for vehicular Ad-hoc networks:paradigms, scenarios, and issues", The Journal of China Universities of Posts and Telecommunications, www.sciencedirect.com/science/journal/10058885, http://jcupt.bupt.edu.cn, 23(2), pp.56–65, April 2016, [Retrieved: January, 2017].

[15] J.Chen, H. Zhou, N. Zhang, P. Yang, L.Gui, and X. Shen, "Software defined Internet of vehicles: architecture, challenges and solutions", Journal of Communications and Information Networks Vol. 1, Issue (1): pp.14-26, 2016, DOI: 10.11959/j.issn.2096-1081.2016.002.

 ISBN: 978-1-61208-583-8

A Framework for the Effective Deployment of Wireless Dynamic Sensor Networks

Maurizio D'Arienzo
Dipartimento di Studi Politici Jean Monnet
Università della Campania
e-mail: maurizio.darienzo@unicampania.it

Simon Pietro Romano
DIETI - Dipartimento di Ingegneria Elettrica
e delle Tecnologie dell'Informazione
Università degli Studi di Napoli "Federico II"
e-mail: spromano@unina.it

***Abstract*—Deployment of wireless sensor networks is a key activity for the enhancement of the state-of-the-art wireless networking systems. In this paper, we present an architecture scheme capable to integrate several heterogeneous modules in a complete and inexpensive wireless dynamic sensor network system. The system can manage a high rate of data gathered from mobile units and can provide precise information in both real time and through a back end service. A real-world implementation of this system is proposed together with some in field performance measurements.**

***Keywords*-Wireless Sensor Network; ZigBee; Performance measurements.**

I. INTRODUCTION

Wireless Sensor Networks (WSN) do represent a growing technology that finds application in different fields like environmental control, interaction among people or computers, remote activation of systems. Such networks can be composed of several units like sensor nodes, actuator nodes, gateways possibly connected to a wide area network like the Internet, and one or more clients. A particular application of WSN is composed of mobile elements, thus introducing a more dynamic scenario that can be defined as Wireless Dynamic Sensor Network (WDSN). The mobility of sensing devices requires a rapid network reorganization and the search of alternative route paths to keep a constant association of single elements to the network. Specific protocols for WDSN help the process of creation, self-organization, route discovery and management of this type of networks [1][2].

The implementation of new services dedicated to WDSN require the integration of several components and technologies to collect, transmit and elaborate data. The data generated or used by units like sensors or actuators are often exchanged through a variety of basic transmission systems mainly based on serial protocols. This data bundle is usually not complex, thus the data preparation preceding the transmission can rely on the resources provided by compact units like micro controllers. As far as the creation of a distributed network, the ZigBee protocol [3] is an international multi hop standard protocol designed with the aim of allowing a straightforward setup and management of a WDSN. The ZigBee protocol allows the creation of different network configurations, from single sink topologies up to a full mesh disposition, and several dedicated devices are nowadays available on the market to develop tailored systems. Although the interconnection among modules is provided by standard protocols, a specific design for the actual implementation of a WDSN system is always required.

There are several emerging applications of WDSN exploiting the integration of modules useful in smart cities [4], smart health [5][6][7], and smart home [8] environments. Among the other proposals, we cite the implementation of an intelligent traffic control system to avoid traffic jams, an infrastructure to manage a bike sharing service, and a framework for the supervision of agricultural cultivation or animal farms. Some of these examples are further analyzed in the Section VI. In the next section II, we focus the attention on our proposal for a framework for the deployment of a complete WDSN capable to integrate in a single transmission system heterogeneous sensors having different characteristics. Section III introduces the proposed architecture scheme, while Section IV provides some details about the current implementation. Section V reports the results of first experiments, before the final considerations presented in Section VII.

II. WDSN TECHNOLOGIES AND PROTOCOLS

The composition of electronic devices in an integrated system requires a shared method to exchange data among the involved units. Simple yet effective communication between two electronic devices can happen by means of Universal Asynchronous Receiver Transmitter (UART) protocol. Transmissions are asynchronous, that means an external clock signal is not required and the number of wires or I/O pins is minimized. The lack of a clock entails a shared configuration between the pairing devices. As a first rule to be shared, the *baud rate* specifies the data communication speed, that is anyway limited to 115200 bits per second. Data are then transmitted in frames and a basic parity bit check mechanism increases the communication reliability. Although simple, the protocol is so common that many electronic components always offer a pair of UART Tx and Rx pins.

Although the UART is a simple and useful protocol, it is not conceived to address multiple communications on a single channel. In the years between 1980 and 1990, Philips developed the I^2C (*Inter Integrated Circuit*) protocol to manage several units by using a single pair of input and output lines, named SCL for *Signal clock* and SDA for *signal data* [9]. The data exchange is still serial and happens on this single bus at predefined baud rate ranging from 100 kbit/s

 ISBN: 978-1-61208-583-8

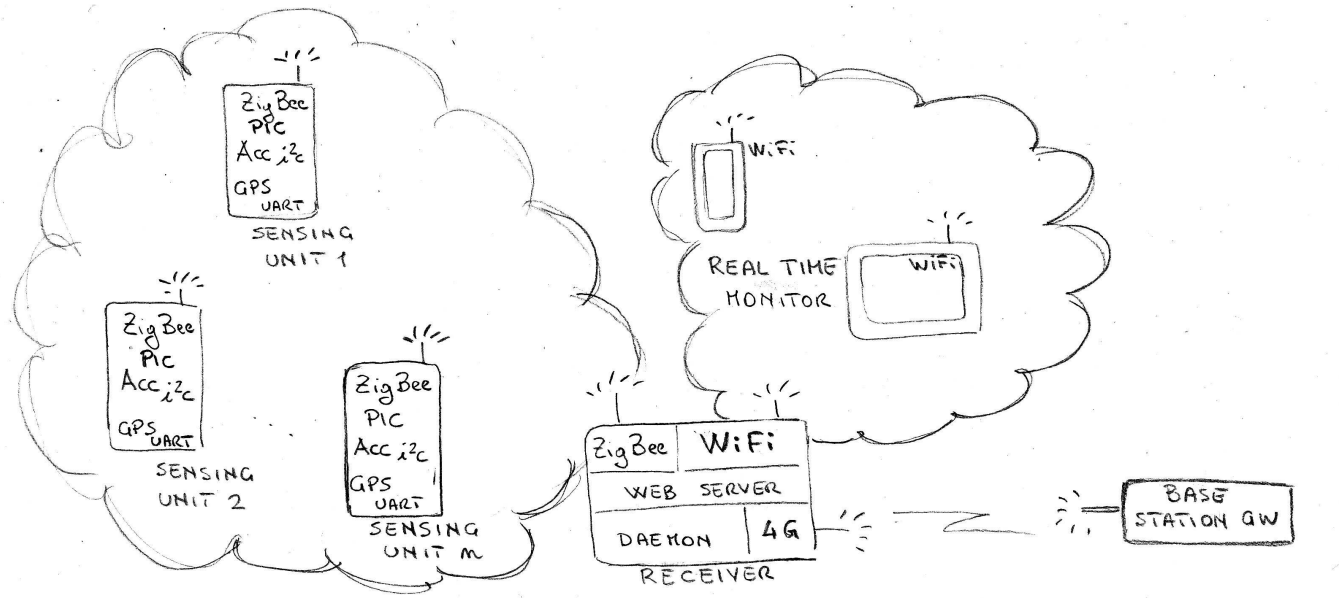

Fig. 1: The architecture scheme

up to 3.4 Mbit/s in the more recent components. All the I^2C units are identified by a unique address, thus connection of multiple units to a single collector is allowed. Despite the possibility to manage even more complex configurations, the usual setup is composed of a single master and multiple slaves. As a basic assumption, both SCL and SDA lines are kept to the high logic level thanks to pull up resistors. The master is responsible for managing the SCL clock line and for starting the communication by lowering the SDA line. Both read and write operations are allowed on the bus. Following a master write of the slave address, the involved slave can eventually take control of the SDA line if a read operation is required. SDA transitions are allowed only when the SCL is low, while SDA data are considered valid when the SCL is high. The ninth bit of each sequence is always considered as an acknowledgement. At the end of each procedure, the master raises up the SDA lines to stop the communication.

Among the available technologies to exchange data on an average distance range, ZigBee is gaining momentum as the leading solution providing multiple configurations, with tens of cheap devices available on the market. ZigBee is a registered trademark managed by an alliance of companies that produce both components and systems. Physical and MAC layers are compliant with the IEEE 802.15.4 standard protocol. Transmission among the devices happens worldwide in the 2.4GHz frequency band, as well as the 868MHz band available, mainly in Europe, for scientific and medical experimentations. Depending upon the unit power, transmission can range from about 10 meters up to several kilometers in the open space. In a ZigBee network a specific device can be configured as either *coordinator*, or *router*, or *end-device*. For each network there is only one coordinator responsible for the initialization and the setup of operational parameters. One or more routers act as intermediate nodes and rely on the Ad Hoc Distance Vector (AODV) protocol to relay data from other nodes towards the coordinator. Finally the end-devices have limited functionality and are usually battery-powered. They limit themselves to sending collected data to the routers or the coordinator, but they cannot relay any data. To better manage the limited amount of energy available, a particular *beacon* can be either enabled or disabled. In the most simple situations the routers are not power constrained, so the beacon is not enabled and they can be always active to wait for data from end-devices. When the routers have to limit their energy consumption, they can alternate between a sleep and an active state. In this case, a beacon is generated to alert the adjacent devices. According to the 802.15.4 standard, 3 topologies can be set up with zigBee, as it will be explained in the following. In the *star* topology, several end devices are connected to a single coordinator, with no need for ZigBee routers. Although this represents the simplest solution, the coordinator can become a bottleneck for the network as it represents a single traffic sink, so this kind of configuration can be adopted when the number of end devices is restrained, or when the transmissions are infrequent. In the *tree* topology the network can be extended with the introduction of the ZigBee routers. The end devices are called children and can connect either to ZigBee routers or to the coordinator, that act as parent nodes in the hierarchy. The coordinator can be connected to both ZigBee routers and end devices. This configuration allows for a wider network extension, but it does not offer redundancy in case of node failure. Notice also that albeit considered in the standard 802.15.4 protocol, zigBee does not support the cluster tree configuration. The last solution is the *mesh* topology. In this configuration, one coordinator, several routers and end devices build up a multi hop network. The range of the network can be extended, and alternative paths may be available in case of node failure. On the downside, the configuration and the management of this topology are more complex and introduce overhead with respect to the previous

 ISBN: 978-1-61208-583-8

solutions.

The most recent update to version 3.0 is an effort to unify the various ZigBee proposals in a single standard.

III. Architecture scheme

The architecture we propose is sketched in Figure 1. It is composed of multiple *sensing units*, at least one *receiver* capable to collect data, *monitor units* and one *base station* potentially connected to the Internet.

Sensing units are identified by their unique code number so that data can be differentiated. The update frequency can be configured according to the particular application. All data are collected at the receiver station and the wireless technology is responsible to manage the correct synchronization among the units for an effective radio transmission. Sensor modules are composed of a dedicated control unit, one or more active sensors, and a transmitter. The control unit relies on a micro controller and is in charge of synchronizing the data collection and to send the bundle of data to the receiver. Sensor modules can be heterogeneous and usually provide data through different protocols.

The receiver collects wireless data and makes them ready for presentation and further computations. It is equipped with a powerful aerial to guarantee the reception of data coming from even remote sensor units. The receiver station can also be provided with additional wireless modules for short or long range data forwarding. A short connection can be useful to provide a real time visualization of data, whereas a long range connection offers the chance to reach a more powerful base station capable to store the data, make off line computations or share data on a wide area network.

Monitor units are mainly intended to be real time devices that can provide a quick view of the current situation. They connect to the receiver station and can visualize all sensed data at a glance. To ensure the widest level of interoperability, the receiver station should make available legacy technologies for a straightforward network connection. Specific applications can be developed to enhance the presentation of data.

Finally, the receiver unit can send data to a distant base station for more complex applications. As the receiver unit and the base station can often rely on a greater availability of energy resources, they can be equipped with more powerful radio devices capable to cover longer distances. The base station should be close to an Internet access point to widen the number of applications that can make use of collected data.

We now present some implementation details related to the introduced architecture modules.

IV. Implementation

We made a complete implementation prototype of a system based on the architecture described above. In order to verify the architecture feasibility, most of the modules are designed starting from inexpensive components and configured to be interfaced with electronic devices available on the market.

A. Sensing unit

The sensing unit prototype is pictured in Figure 2. It is based on the PIC16F87-I/P micro controller acting as a control unit. This Peripheral Interface Controllers (PIC) provides an internal clock up to 8MHz, and supports both UART digital communication and I^2C peripherals. Two different sensor modules are connected to the control unit: a) a Global Position System (GPS) UART sensor that includes the PIC receiver serial line; b) a 3D accelerometer and magnetometer LSM303DLHC sensor based on the I^2C protocol, linked to the specific PIC input line. The magnetometer provides raw data that can be used to calculate the sensing unit position as well as the pitch, roll and heading. The PIC serial line output is instead connected to a wireless module supporting the ZigBee protocol. To provide a high rate of samples, all sensor modules are set to a refresh rate of 5Hz. The control unit executes a

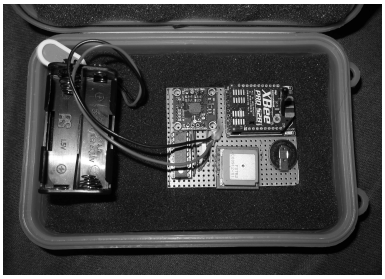

Fig. 2: The sensing unit prototype

routine coded in low level programming language. The routine enters a loop starting from the data read from the sensing units. Data read from each sensor module are composed of 5 samples that get stored in the local PIC memory. A unique code number is appended to all the samples to let the sensing unit be identified; the composed data bundle is eventually sent across the serial output to be processed by the transmitting unit. To ensure the correct management of the data processing phase, all units are set to a baud rate of 38400. Frame composition

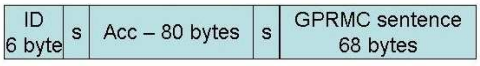

Fig. 3: Sensing unit frame composition

is depicted in Figure 3 as a plain text sentence of 156 bytes. Every set of data ends with a one byte separator. As the current configuration sends 5 sentences per second, the load is of 6240 bits/sec for each sensor.

The radio communication is ensured by a ZigBee module. In order to provide a wide range coverage, a powerful Xbee Pro S2B international module from Digi is installed on the sensing unit. Although this international version limits the power consumption to 10mW, a good aerial allows for a broad radio coverage, as reported by the comparative experiments presented in the following section.

B. Receiver

The receiver is implemented on a Raspberry Pi mini computer running a dedicated image that initializes the USB port

 ISBN: 978-1-61208-583-8

for the interconnection with the radio receiving module and to enable an access point service through a Wi-Fi dongle. The data collection from sensors is ensured by a high gain ZigBee receiver module connected to the main unit through one of the available USB ports. As a result, data are received on the serial line and they are processed by a dedicated daemon that makes the necessary calculations to obtain the instant position, pitch, roll and heading of each sensing unit. A special script arranges such data in a XML file, with a single element for each sensing unit. These data can be immediately accessed in real time from nearby devices like tablets or smartphones via a dynamic web page always accessible from the receiver. Data on the web page are refreshed at the same rate of the data provisioning, that is 5 samples per second in the current configuration. Furthermore, a $4G$ internet key can send the data bundle to a distant base station for data storage, elaboration and presentation on a wide area network.

V. Performance measurement

We have developed a complete system to test the application. Two different configurations for sensing modules are available: the former with an external 5dB gain, 2.4GHz omni-directional aerial, capable of long distance coverage, the latter with a more compact and simpler wire aerial. Power on sensing modules is ensured by 6V 2500mAh batteries. The receiver benefits from a high gain omni-directional external aerial and exploits the power of a battery connected to a recharging system. We tested the coverage of both these configurations, as well as the batteries duration on the sensing units. The actual measures are presented in table I. The first row reports the maximum distance before packet loss occurs. The radio coverage of sensing units equipped with external aerial is far greater than that ensured by internal wire aerial. As far as the battery consumption, the difference between the two configurations is neglectable. We then estimate the energy consumption to send a single bit through the ratio $\frac{MaxPower}{(lifetime)\times frameL}$, with the lifetime expressed in seconds. Considering a bit rate of 6240 bits/s in our current configuration, we calculate a $3.51e-5$ mA consumption per bit sent.

TABLE I: PERFORMANCE MEASUREMENTS

	EXT. AERIAL	INT. AERIAL
DISTANCE [m]	3076	412
LIFETIME [hh:ss]	3:10	
ENERGY/BIT [mA]	$3.51e-5$	

A mobile monitor can exploit the web based system active on the receiver sensor to show the real time measurements, with a refresh rate of $200ms$. A screenshot is shown in Figure 4.

We finally evaluated the precision of the compass. The accelerometer and magnetometer provide the data that can be used to calculate the pitch, roll, and to compensate the tilt of the compass calculation. We then compared the measurements of the tilt compensated compass with the compass reading received from GPS. We derive such data from a reading batch of 300 samples. A routine active in the receiver unit executes

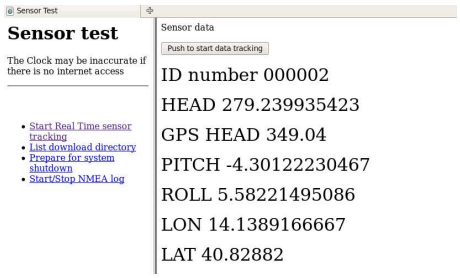

Fig. 4: A screenshot of real time web-based measurements

the required trigonometric computation to get pitch, roll, and tilt compensated compass. We then make an offline difference between the tilt compensated compass and the analog reading from the GPS. We did expect a difference in the precision of these units. The average and the standard deviation of this difference is reported in Table II.

TABLE II: TILT COMPENSATED COMPASS MEASUREMENTS

	AVERAGE	STD DEVIATION
CMP-GPS CMP	65.52	2450.48

VI. Related work

Many related works have studied and applied the concept of WDSN. In [10] the authors evaluate the benefits and drawbacks of the ZigBee technology when applied to the situation of moving nodes. Among the other features, they highlight the presence of a single ZC (ZigBee coordinator), the limited concurrent associations among nodes (120 in total), the maximum time that a child node can remain without communication with its parent node according to a configurable sleep time set on the end node to limit power consumption. Furthermore, it is reminded that routing tables in ZigBee nodes are refreshed every 10 seconds. A real test composed of several fixed end devices and one moving coordinator is presented as a proof of concept to check the re-association feature.

A similar composition of technologies like Radio Frequency IDentification (RFID), Global System for Mobile communication (GSM), ZigBee and PIC micro controller is presented in [11] with the purpose of enabling a system capable to control the car traffic in congested cities. Special sensors are positioned at street junctions to catch the number of passing vehicles and estimate the congestion so that the red lights duration can be tuned accordingly. Also, the system allows for the interception and monitoring of stolen cars, while the approaching of emergency vehicles can activate the green light at the next junction by exploiting ZigBee radio signaling.

ZigBee is still used to ensure the communication in a proposal for green street lights system [12]. Both sensors and actuators are mounted on street lights: sensors estimate the required level of light; data are then sent to a premise energy and management station that can finally send the most appropriate configuration towards the LEDs dimmer on the lamppost.

A complete system that collects data from moving sensing units is presented in [13]. Several sensors mounted on bicycles and worn by cyclists constitute an example of a wireless

 ISBN: 978-1-61208-583-8

BAN (body area network). Rich information concerning the travelled distance, the air pollution, the traffic intensity, fitness parameters like hearth rate, are conveyed to a back end server either through opportunistically encountered wireless access points, or by exploiting the cellular channel of mobile phones. Sensing units are based on the capabilities of Tmote Invent units. Some metrics are introduced to estimate things like pollution level or safety of paths. A localization system based on distance and magnetometric direction travelled is also compared to the information provided by GPS.

VII. Conclusions

The major contribution of this paper is the implementation of a framework for the deployment of a mobile sensor network that can provide high precision data at high rate. In the proposed architecture, a single controller manages the flow of measurements gathered from two different sensors making use, respectively, of UART-based communication and of the I^2C protocol. Data are then shared in a distributed system through the capability provided by a ZigBee network. A complete framework has been implemented to test both the functionality and the basic performance of such a system. Current implemented sensors provide high rate and precise information about the location of units, so they can find application in many fields. We are planning to increment the number of implemented nodes to create a multi hop network and to verify the impact that dense node coexistence may have on the overall system performance.

References

[1] Priyanka Rawat, Kamal Deep Singh, Jean-Marie Bonnin, and Hakima Chaouchi, "Wireless sensor networks: a survey on recent developments and potential synergies", Journal of Supercomputing, Springer Verlag, 2013. doi: 10.1007/s11227-013-1021-9

[2] X. Mao, S. Tang, X. Xu, X. Y. Li, and H. Ma, "Energy-Efficient Opportunistic Routing in Wireless Sensor Networks", in IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 11, pp. 1934-1942, Nov. 2011. doi: 10.1109/TPDS.2011.70

[3] Ata Elahi, and Adam Gschwender, "ZigBee Wireless Sensor and Control Network (1st ed.)", Prentice Hall Press, Upper Saddle River, NJ, USA, 2009.

[4] Muhammad Saqib Jamil, Muhammad Atif Jamil, Anam Mazhar, Ahsan Ikram, Abdullah Ahmed, and Usman Munawar, "Smart Environment Monitoring System by Employing Wireless Sensor Networks on Vehicles for Pollution Free Smart Cities", Procedia Engineering, Volume 107, 2015, Pages 480-484, ISSN 1877-7058, http://dx.doi.org/10.1016/j.proeng.2015.06.106

[5] Mahesh Kumar, "Healthcare monitorng system using wireless sensor network", Int. J. Advanced Networking and Applications 1497 Volume:04 Issue:01 Pages:1497-1500 (2012) ISSN : 0975-0290

[6] B. Vijayalakshmi, and C. Ram kumar, "Patient monitoring system using Wireless Sensor based Mesh Network" Computing Communication and Networking Technologies (ICCCNT), 2012 Third International Conference on, Coimbatore, 2012, pp. 1-6. doi: 10.1109/ICCCNT.2012.6396102

[7] R. Sharma, S. K. Gupta, K. K. Suhas, and G. S. Kashyap, "Performance Analysis of ZigBee Based Wireless Sensor Network for Remote Patient Monitoring", 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, 2014, pp. 58-62. doi: 10.1109/CSNT.2014.21

[8] Chunlong Zhang, Min Zhang, Yongsheng Su, and Weilian Wang, "Smart home design based on ZigBee wireless sensor network", 7th International Conference on Communications and Networking in China, Kun Ming, 2012, pp. 463-466. doi: 10.1109/ChinaCom.2012.6417527

[9] Philips Semiconductor, "I2C Bus Specification" version 2.1, January 2000.

[10] T. de Almeida Oliveira, and E. P. Godoy, "ZigBee Wireless Dynamic Sensor Networks: Feasibility Analysis and Implementation Guide", in IEEE Sensors Journal, vol. 16, no. 11, pp. 4614-4621, June1, 2016. doi: 10.1109/JSEN.2016.2542063

[11] R. Sundar, S. Hebbar, and V. Golla, "Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection", in IEEE Sensors Journal, vol. 15, no. 2, pp. 1109-1113, Feb. 2015. doi: 10.1109/JSEN.2014.2360288

[12] G. Shahzad, H. Yang, A. W. Ahmad and C. Lee "Energy-Efficient Intelligent Street Lighting System Using Traffic-Adaptive Control" in IEEE Sensors Journal, vol. 16, no. 13, pp. 5397-5405, July1, 2016. doi: 10.1109/JSEN.2016.2557345

[13] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G-S. Ahn, and A. T. Campbell, "The BikeNet mobile sensing system for cyclist experience mapping", In Proceedings of the 5th international conference on Embedded networked sensor systems (SenSys '07). ACM, New York, NY, USA, 87-101. DOI=http://dx.doi.org/10.1145/1322263.1322273

 ISBN: 978-1-61208-583-8

Generating Fill-in-the-Blank Tests to Detect Understanding Failures of Programming

So Asai, Yoshiharu Yamauchi
Graduate School of Information Science and Engineering
Ritsumeikan University, Shiga, Japan
email:{asai, yamauchi}@de.is.ritsumei.ac.jp

Yusuke Kajiwara, Hiromitsu Shimakawa
College of Information Science and Engineering
Ritsumeikan University, Shiga, Japan
email:{kajiwara, simakawa}@de.is.ritsumei.ac.jp

***Abstract*—This paper proposes a method to generate a fill-in-the-blank test to detect the understanding failures of students in introductory programming course as early as possible. In programming class in educational institutions, what is important for novices is to make them acquire abilities to exert knowledge and skills in programming in an appropriate way according to each situation. The method pays special attention to the fact that students sharing specific understanding failures are likely to write similar inappropriate code. To generate a fill-in-the-blank test, the proposed method determines code fragments to be blanked out in model code, differentiating inappropriate code written by past students from the model code. An experiment has revealed the method can detect students having understanding failure with high precision rates. The fill-in-the-blank tests generated with our method prevent students from leaving their understanding failure unsolved, because teachers can intensively supervise students who fail to acquire abilities to fully exert the knowledge and the skills in the early stages of the programming course.**

***Keywords*—Programming; e-learning; text mining; clustering.**

I. Introduction

In information technology industry, there is a serious lack of workers due to increase of demand for digital products and services [1]. Educational institutions are urged to educate students who have programming skills to resolve the problem.

In educational institutions, students learn how to write program code in a recommended way to exert the knowledge and the skills according to each situation. The goal of programing learning is to make students understand the knowledge and acquire the skills needed to write programs in recommended ways. However, since the instructor teaches all of the students in a uniform manner, some of them have difficulties to immediately understand the recommended ways to write programs. Such students would write slack programs to avoid missing submission deadlines of assignments, neglecting the lecture goals [2]. If the students understanding failure remains unsolved, they will not learn and, if the process continues, they will fail to learn programming.

Teachers are required to detect understanding failures of students as early as possible. To achieve it, they should examine students understanding every class with a test. The way to test the students is at the instructor's discretion. In addition, it imposes a big burden on them, if they have to manually assess answers of many students. Students often submit the same erroneous code as answer to identical assignments in educational institutions. This means, past and current students have identical understanding failures. In this work, we propose a method to detect understanding failures of programming. We adopt a fill-in-the-blank test for the detection of understanding failures, to mitigate the burden on teachers. Fill-in-the-blank tests placing blanks in the part of program where many students are likely to write inappropriate code are expected to be effective to find understanding failure of students. We focus on programs past students wrote to detect code fragments to blank out in fill-in-the-blank tests. Programs past students submitted are represented by vectors figured out in terms of the word similarity to example code and explanations from a textbook. To identify the part involving inappropriate code fragments, the vectors are classified into clusters, to distinguish ones that are the most different from model code. The clusters consisting of pieces of code that are the most different from the model code are referred to as inappropriate clusters. The difference of program code in an inappropriate cluster from the model code is identified to determine code fragments to be blanked out. The method proposed in the work provides an environment to score the fill-in-the-blank tests automatically. It is expected that the fill-in-the-blank tests along with the scoring environment would detect understanding failures of present students without a great effort from the part of the teachers.

In this paper, Section II introduces understanding failures and the efficacy of fill-in-the-blank test on learning programming. Section III explains our method to generate a fill-in-the-blank test. This section illustrates the automatic scoring environment. Section IV indicates the experiment to validate the method with its result. Section V evaluates the result to discuss the validity of the method. Section VI summarizes our work.

II. Understanding Failure of Programming

A. Problems in Programming Class

In the C programming course in educational institutions, students learn programming in each teaching unit, which corresponds to one combination of a lecture class and an exercise class. Students learn knowledge and skills in each teaching unit. To acquire programming ability comprehensively, students are required to solve assignments in a specific teaching unit, using the knowledge and skills they have already learned in the preceding ones. In exercise classes, it is not enough for students only to write a program that behaves in

 ISBN: 978-1-61208-583-8

a required in the assignment. They should train themselves to write the program code in an appropriate manner as they are written in model code. Through the training, they would get abilities to handle the knowledge and skills they have learned. In order to proceed with the training along with the arrangement of units, students must understand all the contents taught in every teaching unit.

Nowadays, various example programs are provided on the Internet and in books [3]. Students try to find example code [2]. If they do not understand the example code well, they would construct program in a cut-and-paste manner, using code fragments extracted from the example code. Such programs involve inappropriate code fragments to implement the required program behavior. They submit executable programs which are composed in the above way, without understanding how to write a program in a recommended way. They should understand each of the code fragments to organize the program as a sequence of code appropriate for the behavior. Even though this behavior can help with the assignment at hand, students should avoid this reckless copying of code fragments without understanding. It might lead the students to another understanding failure in coming classes because they fail to attain sufficient understanding and abilities. Successive understanding failures would drive them to give up programming. Once the students give up programming, it is difficult to direct them again to its learning. Teachers should find students who fail to attain enough understanding in programming classes as early as possible.

B. Appropriate Code and Inappropriate Code

It is required to write programs according to their specifications and a specified programming style [4]. It means there is a recommended way to write a program to be concise and easy to read to accomplish the specific behavior in programming. A program is represented by a sequence of statements meeting a specific pattern.

Students should write the program in a manner model code illustrates to satisfy the requirements in the programming exercises. They can write code similar to the model code, if they sufficiently understand the sequence of statements meeting a specific pattern taught in every teaching unit. This paper refers to a program which implements specified behavior in the recommended way as appropriate code. Appropriate code is represented by a sequence of statements meeting a specific pattern. It becomes similar to the model code in terms of the appearance and the frequency of the statements. On the other hand, code of students who cannot write an appropriate code contains parts different from the model code.

Let us consider two pieces of code for the assignment to understand iteration as shown in Figure 1. Both of them take the same behavior. Code A is described with a for-statement, and code B is described with a while-statement whose condition is true. Code B uses a break statement inside the if-statement to exit from the loop. Although both of them behave as specified, code B is different from the recommended way. In this paper, code which implements specified behavior with a sequence of statements against the recommended pattern is referred to as inappropriate code. It is assumed that inappropriate code is caused by understanding failure of students for correct programming.

```
#include <stdio.h>

int main() {
  int i = 0;
  for ( i = 0; i < 10; i++ ){
    printf("%d\n", i);
  }

  return 0;
}
```

Code A

```
#include <stdio.h>

int main() {
  int i = 0;
  while ( 1 ) {
    printf("%d\n", i);
    i++;
    if ( i >= 10 )
      break;
  }

  return 0;
}
```

Code B

Figure 1. Appropriate and inappropriate code examples

C. Fill-in-the-blank Assignments to Identify Understanding

It is important to evaluate whether students have acquired the knowledge and the skills of programming. Since this evaluation reveals the student achievement, it benefits not only to students but also the teachers, because teachers can plan how to supervise students.

A general way to measure understanding of programming is a scratch test, as which we refer to a test requesting students to write whole program code from scratch. Several patterns of code sequences can construct the program which generates the specified output for the same input. Even if students do not understand the code sequences to be learned, they can meet the requirement by other patterns. In addition, there is a possibility that they have combined code fragments in a reckless manner to generate a program. To assess such a program, the teacher has to read it in order to identify whether they have understood. He needs enormous time and effort because he takes care of many students. Since students proceed with learning of programming based on what they have learned, it is difficult for them to make progress, if they leave understanding failure. The teacher must evaluate the understanding every programming class despite enormous time and effort needed to do so. The teacher should also provide various assignments for students to confirm their understanding. It is not feasible to measure the understanding of programming for many students by scratch tests.

The alternative way to measure programming skills is a fill-in-the-blank test. The teacher blanks out a specific part of the sequence of statements which implements specified behavior, to measure the understanding of a student focusing on a specific point. A few kinds of code are suitable to fill the blank. The blanks are so small that the standard to assess students does not vary with teachers.

In addition, fill-in-the-blank tests are useful to measure program understanding [5]. Appropriate code to implement a

 ISBN: 978-1-61208-583-8

specific behavior consists of a sequence of statements meeting a specific pattern. Suppose a test has a blank hiding a part of a program, so that the blank cannot be filled without knowledge on a sequence of statements matching a recommended pattern. Unless students have programming skills to write the recommended code, they cannot fill the blank with correct code. Such fill-in-the-blank test can measure whether they have learned the programming skills. When they fill some answers to the blank, they read the entire program as well as the code fragments around the blank, to guess the behavior. Since they try to consider a procedure implemented by code around the blank, a proper fill-in-the-blank test makes the students understand how to write the recommended code.

However, when either the place or the size of blanks are irrelevant, it will not detect understanding failures on how to write the recommended code. Thoughtless setting of blanks impedes revealing their understanding failure. A support is necessary for teachers to determine the parts to be blanked out in order to reveal the students who have understanding failure.

D. Related Work

Kashihara et al. [6] generated fill-in-the-blank tests with a program dependence graph to find a blank suitable to measure the understanding of program code. This method makes only one blank in a program. It is considered there are several kinds of inappropriate pieces of code in a program. In particular, the diversity of inappropriate code is prominent in advanced assignments. Many fill-in-the-blank tests must be prepared in advance to detect various understanding failures with the method.

Ariyasu et al. [7] support the automatic generation of fill-in-the-blank tests meeting intention of teachers with syntactic analysis of program code. However, the teachers themselves must look over the understanding status of all the students to determine what should be examined in the test. It depends on the abilities of the teacher to generate fill-in-the-blank tests which can reveal programming understanding. These works take into account neither of latent understanding failure nor supports to determine what should be examined.

Funabuki et al. [8] proposed a Java learning system to generate a fill-in-the-blank assignment function which assists learning of reserved words. This system blanks out in the model code by selecting reserved words randomly. It is difficult to measure understanding failure for all of programming skills to be acquired since teaching units other than reserved words are not covered. An alternative method is necessary to generate fill-in-the-blank tests to solve these problems.

III. Revealing Understanding Failure from Past Student Code

A. Goal and Significance

To complete a fill-in-the-blank test, students have to determine a code fragment to fill in a blank after they understand a sequence of code around the blank. It forces them to understand the meaning of the code fragments before and after the blank. An automatic scoring system, which allows the students to retry the test many times, allows them to become aware of their own understanding failures.

Since any fill-in-the-blank assignment is issued based on a model code, programming novices can study how to organize a program with code fragments appropriately to achieve the required behavior of the program. For each assignment issued in previous programming courses, the teacher has accumulated examples of inappropriate code many of the past students in the educational institution wrote. Understanding failures seem to vary with educational institutions, because students with similar understanding belong to a specific educational institution. Teachers should not issue universal assignments presented in commercial textbooks. Using the information from past students, teachers can generate fill-in-the-blank tests suitable to detect understanding failures specific to the educational institution. Furthermore, the automatic scoring system releases teachers from boring tasks to examine a lot of program code of the students. The teachers can concentrate on the intellectual work to generate fill-in-the-blank tests to identify students who have understanding failures.

B. Inappropriate code of Past Students

In this work, we propose a method to generate fill-in-the-blank tests to identify understanding failures which cause students to write inappropriate code. Figure 2 shows the method overview. In a specific educational institution, students who have identical understanding failures have a tendency to write the same kind of inappropriate code over the years. Using this tendency, the method generates fill-in-the-blank assignments from program code past students wrote. It is assumed that the same assignments are given in the past and the present years. Present students are likely to write almost the same inappropriate code as past students did, if both of them have identical understanding failures. It is possible to find a code fragment to be blanked, if we reveal inappropriate code fragments past students wrote.

In order to find inappropriate code fragments, we classify programs written by past students. The similarity of a program to example code in each teaching unit is calculated based on the appearance frequency of statements such as if-statements and for-statements. Every program code is represented by a vector in a coordinate space. Each element of the vector corresponds to a teaching unit. Programs are classified into clusters in terms of the similarity to example code in the teaching units. Student code submitted for an assignment is regarded as inappropriate, if it is very different from a model code of the assignment.

It is conceivable that there are several inappropriate fragments in a program. To determine which parts of code are inappropriate, the method computed the difference between student code and the model code. This method reveals parts

 ISBN: 978-1-61208-583-8

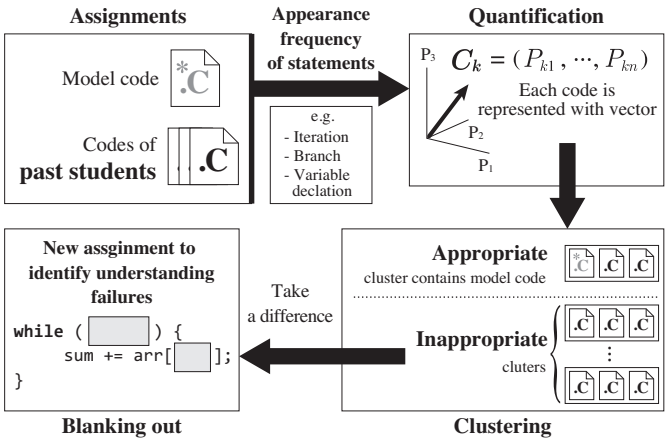

Figure 2. Method overview

involving inappropriate code. The parts express the code fragments to be blanked out.

Fill-in-the-blank tests generated by the method reveal whether each of current students understands the blanked-out parts. Students who cannot fill the blanked-out parts correctly are likely to have a specific kind of understanding failure. Teachers should supervise those students, to prevent them from leaving the understanding failure for any teaching units unsolved.

C. Extraction of Characteristics with Text Classifier

For each assignment, a teacher would prepare a model program to show students how to write an appropriate code in a recommended way as it is written in a textbook like [9]. The proposed method represents quantitatively how student code is appropriate like the model code.

Novice programmers of conventional programming languages such as C learn programming skills according to teaching units such as iteration and arrays. The characteristics of example code and sentences to explain them vary with teaching units in a textbook for C novice programmers. A program similar to example and their explanation in a teaching unit is regarded to be in accordance with what are learned there. For each assignment, the proposed method treats its model code and student code as source code to calculate their similarity to the teaching units in the textbook. A program is represented by a set of the probability that the program is similar to every teaching unit in the textbook.

Program code and comments in a program as well as example code and sentences in the textbook are divided into words with MeCab [10], which is a tool for morphological analysis. The text classifier implemented with the Naive Bayes filtering [11] records of the appearance degree of words in every teaching unit of the textbook in advance. For a program, the text classifier calculates the appearance degree of words used in every teaching unit of the textbook. The appearance degree probabilistically represents to which teaching unit of the textbook the source code is similar in terms of the

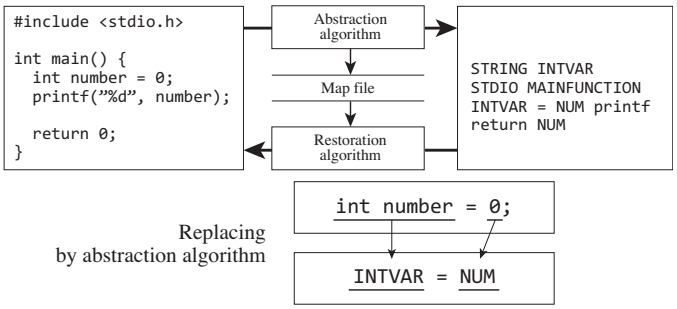

Figure 3. Translation to abstracted code

appearance frequency of the words. First, the proposed method calculates the appearance degree representing the similarity to every teaching unit. It selects a specific number of teaching units of high similarity. To address characteristics of source code, the proposed method uses a vector whose elements correspond to those teaching units. Second, the method represents characteristics of source code written by past students with vectors having those elements. The vectors of student programs obedient with the teaching units would be similar to that of the model code.

Let us calculate the characteristics of programs, including model code and student code. To represent the characteristics of programs, it is preferable to eliminate the variance of identifier names and literal values in every program. To achieve it, the method translates source code into abstracted code. Figure 3 shows an example of the abstraction. In the figure, integer variables, numerical values, and for-loop statements are replaced with INTVAR, NUM, and FOR, respectively. The abstraction makes it explicit which statements are used frequently in source code. Although this abstraction causes the order of words to permute, it does not matter because the method calculates only the appearance frequency of words. Using correspondence of replacing words to original code, the abstracted code can be restored to the original code.

The vectors of programs past students submit for an assignment is compared with that of the model program for the assignment. The vectors of programs, which use appropriate code like the model program are placed near the vector of the model program. Inappropriate code involves eccentric code sequences or redundant processing, which are not found in the model code. It is considered the vector of a program composed of them is considerably far from the vector of the model program. The method classifies the programs written by past students into 2 categories; one is appropriate like a model program, while the other is inappropriate.

D. Classifying into Appropriateness and Inappropriateness

In this work, it is assumed that both of past and present students who have a same understanding failure write a similar inappropriate code.

The proposed method classifies student program and the model program for an assignment to find inappropriate code the students are likely to write. The method applies a cluster

 ISBN: 978-1-61208-583-8

analysis for vectors which represent student programs and the model program. The number of clusters cannot be determined in advance, due to the variance of inappropriate code for every assignment. The method adopts the Ward's method for the clustering. At first, each assignment is assumed to have less than three kinds of inappropriate code. Using a dendrogram generated as a result of the analysis, the method partitions clusters so that code which seems to be inappropriate should not be placed in the same cluster as the model code. The method refers to the cluster containing the model code as an appropriate cluster, while the other cluster as an inappropriate cluster.

Progmras classified into appropriate cluster have high similarity with the model program. Students whose code in the appropriate cluster are considered to write their code, after they understand knowledge and skills to be acquired. Meanwhile, programs classified into inappropriate clusters contain some inappropriate code fragments. Students who wrote the programs in the inappropriate clusters seem to be accompanied with understanding failures corresponding to the inappropriate code. Those students should be supervised as early as possible not to leave their understanding failures uncorrected.

E. Blanking Out Code from Differences

If students write programs falling into an inappropriate cluster, teachers should notify the students that they may have a specific understanding failure. The students with the understanding failure should understand why the model code has code fragments different from theirs. Since it leads them to the right understanding, they could modify their program closer to the model program.

A filling-in-the-blank test is generated to examine whether the students who wrote programs in an inappropriate cluster commit the specific understanding failure. A program in the inappropriate cluster has a code fragment where the usage of statements in programming is different from that in the model program. A filling-in-the-blank test is generated with a part of the code fragment blanked out.

The proposed method has to identify what part is to be blanked out in the model code. All programs in an inappropriate cluster have same tendency in its usage of programming statements. One of programs in the inappropriate cluster is picked up, to take its difference from the model code. As it is pointed out in Section III-C, variance of identifiers is inconvenient to figure out the difference. Abstracted code made from the inappropriate code is used to take differences, in the same way as in the classification. When we use the abstracted code, the difference does not come from variance of identifier names, but from discordance of statement usages in C programming. In the model code, the proposed method detects the code fragments corresponding to the difference. In case of Figure 4, a part of the iteration statement in the model code is replaced with a blank, because there is difference in the while statements between the two programs. For example, the condition of the while-statement should be blanked out

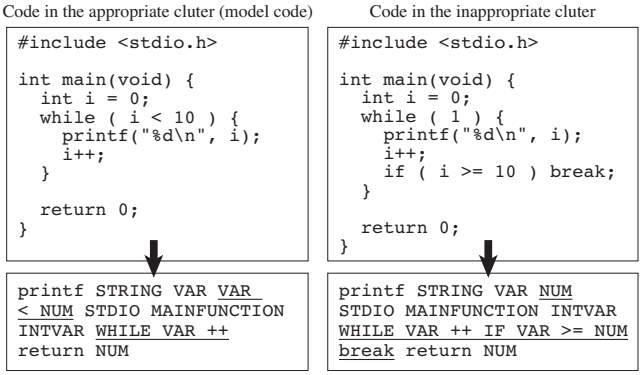

Figure 4. Difference between appropriate and inappropriate code

in the model code. Students having understanding failure for the iteration with while-statement cannot fill the blank correctly, because they always try to use an infinite loop. The above procedure is applied to all the inappropriate clusters to determine blanks suitable to detect understanding failure corresponding to each of the inappropriate clusters.

A blank in a filling-in-the-blank test implies the part where the students might write inappropriate code. Suppose they cannot fill one blank, while they can fill appropriate code for other blanks. It means they have a specific kind of understanding failure regarding to the code fragment around the blank. Providing students with various filling-in-the-blank tests, we can identify students committing every kind of understanding failure. Teachers should supervise the students to understand not only the code to fill the blank, but also why they should write the code, because they wrote the inappropriate code based on a wrong way of understanding.

F. Automatic Scoring for Fill-in-the-Blank Tests

It is important that fill-in-the-blank tests have the students consider what a correct answer is. Various fill-in-the-blank tests given to students clarify understanding failures of students. The proposed system provides an automatic scoring system of fill-in-the-blank tests, so that students may check the correctness of their answers at any time. This system is executed on Web server. Students engage in fill-in-the-blank tests on Web pages. When a student submits a fill-in-the-blank test filling its blanks with his code, the system scores it to notify him the result immediately. The immediate notification makes students strongly conscious of the tests they fail. It prevents the students from leaving themselves unaware of the understanding failures.

Conventional ways to automatic scoring of filling-in-the-blank tests mainly use string comparison of student's answers with ones teachers expect. It judges that the student's answers are correct when they are fully matched with the expected ones. Students send diverse answers with trivial differences. For example, when several variables are initialized with literals, their order does not matter. Since correct answers exist infinitely, the string comparison is unsuitable for the automatic scoring.

 ISBN: 978-1-61208-583-8

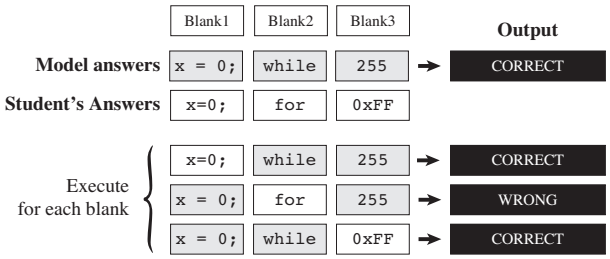

Figure 5. Working example of the automatic scoring system

The proposed method in this work adopts an alternative way which compares execution results of programs to address the diverse answers. Programming assignments for novices would require students to write a program with some outputs on CUI. The way of automatic scoring in the proposed method generates an executable code using both of strings a student answers and ones a teacher prepares. Only one of the blanks is filled with the student answer corresponding to it, while the others with the ones the teacher prepares. The filled code is executed to verify whether the execution output is the same as the output of the code filled with all of the strings the teacher prepares. It enables each of student's answers to be separately scored, even if a filling-in-the-blank test has multiple blanks.

Let us explain the scoring way with a fill-in-the-blank test with 3 blanks shown in Figure 5. Let the prepared answers for Blank1, Blank2, and Blank3 be code fragments "x = 0", "while", and "255", respectively. When they are specified for the blanks, the output is expected to be string "CORRECT", which is printed on CUI. Suppose a student specify code fragments "x = 0", "for", and "0xFF" as his answers.

First, Blank1 is filled with student's answer "x = 0", while the other blanks with the prepared answers. This program code prints string "CORRECT" when it is executed. Since the output is the same as the expected one, the student's answer is judged to be correct. Second, Blank2 is replaced with student's answer "for", while Blank1 and Blank3 are replaced with their prepared answers. The execution result of this program code is different from the expected one. It turns out student's answer "for" is wrong. Finally, Blank3 is filled with student's answer " 0xFF" . Suppose "CORRECT" is printed as a result of the execution. Although the student specifies a code fragment different from the prepared one for Blank3, he is judged to answer correctly, because the output is identical with the expected one. The student is notified that he presents correct answers for both of Blank1 and Blank3, while fails for Blank2.

IV. Experiment

A. Method and Purpose

We conducted an experiment to verify that the fill-in-the-blank tests generated by the proposed method can detect understanding failures. 122 students at College of Information Science and Engineering of Ritsumeikan University participated in the experiment. All of them attended an introductory C programming course, which consists of lectures and exercises, over a year ago.

In the experiment, each student answered the assignments with 3 phases. In the first phase, the student wrote a program from scratch. Let us refer to it as a scratch test. In the second and the third phases, the student solved the two types of fill-in-the-blank tests. One type of the fill-in-the-blank tests is generated with the method, and the other is produced manually by a teacher. The students repeated the answering for these fill-in-the-blank assignments. For the former type of the fill-in-the-blank assignment, the fill-in-the-blank tests were produced, blanking out several code fragments of its model code. To prevent the students from reusing code fragments, variable names in the model code differ in the two fill-in-the-blank tests. For the latter type, when a teacher produced fill-in-the-blank tests, he was provided with the assignment and its model code. He specified code fragments to be blanked out in the model code, as well as his intention regarding what he wanted to confirm with the blanking.

They signed in a website for the experiment with personal user ID and password to challenge the tests. To make students engage in programming assignments as usual, the students were permitted to carry out the tests in any environment such as home and in the university. We settled 2 weeks for the experiments. There was no time constraint other than the submission deadline. Since we intend that they solved the tests for themselves, code copied from websites and digital books should be excluded from the experiment data as cheatings. In order to find the cheating, we examined rapid character filling for blanks. In the scratch test, students can compile their programs at any time to check the output. In the fill-in-the-blanks tests, they could check whether the answers are correct, submitting their answers to the automatic scoring system. The system notified them the scoring result immediately. When they finished or gave up trying a test, they pressed the button to move to the next test.

B. Generating Fill-in-the-Blank Tests

We prepared fill-in-the-blank tests from past programming exercises based on the method. Among assignments given in C Programming Exercise Courses, which College of Information Science and Engineering of Ritsumeikan University offered in the first semester of 2016 academic year, the following five ones are chosen as the tests for the experiment. They are related to knowledge and skills for which students are likely to have understanding failure, such as iterations, functions, arrays, and pointers [12]. We refer to each of the followings as Test A to Test E, respectively. The parentheses indicate teaching units to which the problem relates.

Test A (iteration) Calculate to print interior angles of each regular N-sided polygon for the integer N from 3 to 12 with a while-statement. When the interior angle is not an integer, skip the iteration with a continue-statement.

Test B (function and array) Print the number of characters from A to Z included in any character strings specified from the standard input.

 ISBN: 978-1-61208-583-8

Test C (iteration) Let us build up a pyramid, placing cube stones without gaps. Given the number of cube stones, figure out the number of pyramid steps and remaining stones.

Test D (function and array) Implement a function to make product of two-by-three matrix and a 3-dimensional vector. Print the product for element values given from the standard input.

Test E (function, pointer and iteration) Inserting a hyphen before non-vowel characters in any character strings given from the standard input, print the string which has the same length as the original one. Use the given function to judge whether a character is a vowel.

For each of the assignments, a teacher wrote a model code, while students submitted their programs. To generate a fill-in-the-blank test for the assignment, the proposed method classifies vectors representing the model program and past student programs.

In order to generate a fill-in-the-blank test, the method picked up a program from the inappropriate clusters, respectively, to take its difference from the model code. On the other hand, the fill-in-the-blank tests were produced by a teacher who engaged in the class of C Programming Exercise. After he determined the intention of each test, he blanked out several parts in the model code, referring to assignments and their model programs.

C. Scoring Results

We obtained answers for the tests from the 122 subjects. Some of the answers were not finished solving tests completely or suspected of the cheatings. We do not use such invalid answers for evaluation of the proposed method. To ensure the fairness in the experiment, we used answers that each subject submitted for the first time. We score all the scratch tests by hand to find inappropriate code fragments. We judged answers for the scratch tests incorrect, if we find inappropriate code in them. The followings are inappropriate code fragments found in the scratch tests. For example, A1 and A2 are found for Test A.

A1 The place to do increment for the loop.
A2 The condition to call continue statement.
B1 The place to call the function toupper.
B2 Counting the number of each alphabetic character.
C1 The initial value to count the steps.
D1 Not initializing an array to store the matrix product.
D2 Not using for-statement for the matrix products.
E1 The direction to search character strings.
E2 Escaping from the loop with break statement.

Table I shows the rate of correct answers which the subjects have given finally in the scratch tests and the both kinds of fill-in-the-blank tests from the proposed method and the teacher production. The rate is calculated within the valid answers. In the scratch tests, correct code means an executable program working normally without any inappropriate code. The values in parentheses indicate the rate difference of each fill-in-the-blank test against the scratch test.

TABLE I. RATE OF CORRECT ANSWERS

TEST	SCRATCH	METHOD	TEACHER
A	0.87	0.92 (0.05)	0.94 (0.07)
B	0.29	0.83 (0.54)	0.83 (0.54)
C	0.60	0.60 (0.00)	0.92 (0.32)
D	0.61	0.74 (0.13)	0.87 (0.26)
E	0.20	0.70 (0.50)	0.31 (0.11)

TABLE II. CONDITIONAL PROBABILITIES

CLUSTER	$n(B)$	$n(S)$	$n(S \cap B)$	$P(B \mid S)$	$P(S \mid B)$
A1	11	14	9	0.64	0.81
A2	8	8	4	0.50	0.50
B1	37	70	28	0.45	0.75
B2	23	27	17	0.63	0.73
C1	40	38	25	0.66	0.63
D1	30	36	18	0.50	0.60
D2	24	35	19	0.54	0.79
E1	14	17	14	0.82	1.00
E2	11	10	8	0.80	0.73

D. Conditional Probabilities

In the experiment, we focus on the subjects whose answers were incorrect for either the scratch tests or the fill-in-the-blank tests. A conditional probability is defined in order to prove relevance of fails in the fill-in-the-blank tests generated by the proposed method with incorrect answers in the scratch tests. $P(Y \mid X)$, the probability of Y under the condition X, is given by the formula:

$$P(Y \mid X) = \frac{n(X \cap Y)}{n(X)} \quad (1)$$

Where $n(X)$ denotes the number of event X. We let S and B represent, respectively, an event where the scratch test is incorrect, and a fail in the fill-in-the-blank test. Table II shows the conditional probabilities.

V. EVALUATION AND DISCUSSION

A. Causing Inappropriate Code Fragments

Table II shows the conditional probabilities students write inappropriate code for the scratch tests and the fill-in-the-blank tests generated by the method.

$P(B \mid S)$ indicates the probability a subject who writes a specific inappropriate code in the scratch test mistakes in the filling-in-the-blank test presenting a blank he might fill with the same kind of inappropriate code. It is the recall, which implies to what rate fill-in-the-blank test can detect subjects who has understanding failure. On the other hand, $P(S \mid B)$ indicates that a subject who have mistaken in the blank of the fill-in-the-blank test writes the same kind of inappropriate code in the scratch test. It is the precision, which implies how much we can trust results of fill-in-the-blank tests generated by the method.

From the table, $P(S \mid B)$ shows the high values over 0.70 in the six inappropriate clusters, A1, B1, B2, D2, E1, and E2 of the fifteen. Meanwhile, $P(B \mid S)$ was more than 0.70 only in the two items. It suggests students giving the wrong answer in fill-in-the-blank tests are likely to write inappropriate code fragments in the corresponding scratch tests. That suggests the teacher may conclude that they have the understanding

 ISBN: 978-1-61208-583-8

failure for the code fragments to be filled in the test. The teacher should supervise them to correct their understanding failure. The recall rates, which are not high enough, mean fill-in-the-blank tests generated by the method fail to detect many students who are likely to write the inappropriate code. We need to deal with this problem.

Since the proposed method automatically scores fill-in-the-blank test as explained in Section III-F, it does not require heavy effort of the teacher to score. They can give students more fill-in-the-blank tests than scratch tests. In the experiment, we gave only one test for each kind of inappropriate code. To achieve specific behavior of a program, students need to acquire programming knowledge or skills corresponding to it. When students commit understanding failure for the knowledge or the skills, they are likely to write inappropriate code to achieve the program behavior. Even in other assignments, they would use the same kind of inappropriate code to write code founding on the identical programming skill. The teacher can detect more students with the understanding failure who write the inappropriate code, giving several fill-in-the-blank tests to examine the identical programming knowledge or skills.

B. Comparison with Correct Answer Rates

In the experiment, students engaged in two kinds of fill-in-the-blank tests; one was generated by the proposed method while the other produced by the teacher. Let us compare the rate of correct answers for the two kinds of fill-in-the-blank tests.

Fill-in-the-blank tests are required to detect students who write inappropriate code caused by understanding failure. Suppose a fill-in-the-blank test which places blanks in any part where the students might write inappropriate code for the assignment. It can detect any kind of understanding failure which might occur for the assignment. Such fill-in-the-blank test has a feature to detect the occurrence of every inappropriate code in the scratch test. In other words, it is preferable that the rate of correct answers of the fill-in-the-blank test approaches to that of the scratch test.

Let us compare the differences of the rate of correct answers for the fill-in-the-blank test from the rate of the scratch test, in both cases: the ones generated by the proposed method and the ones produced by the teacher. See Table I. For Test A, C and D, the method has smaller difference than the teacher. For Test B, the difference is identical each other, while the teacher has smaller difference for Test E. For four tests of the five, the proposed method has similar correct answer rate to the scratch test. For Test E, it is considered that difficulty of the test is advanced and its program is not suitable for blanking out. It is necessary to select basic assignments which contain the teaching units to be acquired.

The proposed method assumed students in an educational institution would commit same understanding failures for same assignments over years. Under this assumption, the method analyzes programs of the past students in the same educational institution quantitatively, to identify where inappropriate code frequently occur in the model code for each assignment. On the other hand, the teachers blanked out parts of the model code without quantitative analysis. Due to the quantitative analysis, fill-in-the-blank tests by the method were more successful to find the students who has understanding failure than the ones by the teacher.

VI. Conclusion

In this paper, we have proposed the method to generate a fill-in-the-blank test which detects understanding failure of present students. The method classifies programs past students wrote, to find inappropriate code fragments caused by the understanding failures. Scoring results of our fill-in-the-blank test contribute to finding students who would be at loss due to unconscious understanding failure early.

The high precision values from the experiment indicate that our fill-in-the-blank tests can reveal students having understanding failure for knowledge and skills of programming from inappropriate writing ways of code. The method enables teachers to supervise students poor in understanding intensively.

In the future, we give more fill-in-the-blank tests to detect students who have understanding failures completely. In addition, it is necessary to discuss concrete teaching ways suitable for each kind of understanding failure, analyzing inappropriate code caused by it.

References

[1] Ministry of Economy, Trade and Industry, "Summary of research results on latest trends and future estimates of it talent," http://www.meti.go.jp/press/2016/06/20160610002/20160610002.pdf [retrieved: December, 2016].

[2] M. Kim, L. Bergman, T. Lau, and D. Notkin, "An ethnographic study of copy and paste programming practices in oopl," in *ISESE '04 Proceedings*, 2004, pp. 83 – 92.

[3] Stack Exchange, Inc., "Stack overflow," http://stackoverflow.com [retrieved: January, 2017].

[4] B. W. Kernighan and P. J. Plauger, *The Elements of Programming Style 2nd Edition*. McGraw Hill, 1978.

[5] A. Kashihara, M. Soga, and J. Toyoda, "A support for program understanding with fill-in-blank problems," *JSISE*, vol. 15, no. 3, pp. 129 – 138, 1998, in Japanese.

[6] A. Kashihara, A. Terai, and J. Toyoda, "Making fill-in-blank program problems for learning algorithm," in *ICCC'99*, 2001, pp. 776 – 783.

[7] K. Ariyasu, E. Ikeda, T. Okamoto, T. Kunishima, and K. Yokota, "Automatic generation of fill-in-the-blank exercises in adaptive c language learning system," in *DEIM Forum*, 2009, pp. 776 – 783, in Japanese.

[8] N. Funabiki, Y. Korenaga, T. Nakanishi, and K. Watanabe, "An extension of fill-in-the-blank problem function in java programming learning assistant system," in *2013 IEEE Region 10 Humanitarian Technology Conference*, Aug 2013, pp. 85 – 90.

[9] M. Moriguchi, "Worried C," http://9cguide.appspot.com/en/index.html [retrieved: December, 2016].

[10] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis," in *Proceedings of EMNLP*, 2004, pp. 230 – 237.

[11] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825 – 2830, 2011.

[12] L. Kaczmarczyk, E. Petrick, J. P. East, and G. L. Herman, "Identifying student misconceptions of programming," in *Proceedings of SIGCSE'10*, 2010, pp. 107 – 111.

 ISBN: 978-1-61208-583-8

Smart Power Switch for Smart Homes

Khalid Tarmissi
College of Computers and Information System
Umm Al Qura University
Makkah, Saudi Arabia
kstarmissi@uqu.edu.sa

Mudasser F. Wyne
School of Engineering and Computing
National University
San Diego, USA
mwyne@nu.edu

***Abstract*— The main focus of this paper is to analyze and design a Proof-Of-Concept (POC) Smart Power Switch (SPS) that will give users the ability to conveniently turn their home into a Smart Home. This Smart Power Switch allows the user to control any of the home appliances from anywhere. A standard switch/power outlet in a home can be replaced with a SPS that can be controlled remotely with any smartphone or Personal Computer (PC). Smart Homes can also reduce energy usage by up to 30% for typical residential users. Our research can be considered as an application of Internet of Things (IoT) for Smart Homes. A common definition of Smart Home is an "electronic networking technology to integrate devices, appliances and security so that the entire home can be monitored and controlled centrally as a single machine.**

Keywords- Proof-Of-Concept (POC), Smart Home, Internet of Things (IoT), Information Technology (IT).

I. INTRODUCTION

Since its beginnings, the Internet has evolved significantly, and, in recent years, it has influenced every aspect of our lives. In line with this Internet explosion, it appears we are about to connect everything in our physical world to the Internet. One of the areas that will be affected by this change is our homes, the place where we spend most of our time. Although some electronic devices are already connected to the Internet, in the future, all the objects in our home may get connected to the Internet, thus transforming our homes into Smart Homes. The idea of a Smart Home has existed for a long time, but actual Smart Homes were not designed and built until the early 2000s [12]. However, since this idea is relatively new and the related technology is still under development, it has been referred to by many names such as Smart Homes, Home Automation, Digital Home, etc. A common definition of Smart Home is an "electronic networking technology to integrate devices, appliances and security so that the entire home can be monitored and controlled centrally as a single machine" [2]. Any device in our home that uses electricity can be put on our home network and on a single command. Whether the command is given by voice, remote control, tablet or smartphone, the home reacts. Most applications of Smart Home ideas are related to lighting, home security, home theater and entertainment, and thermostat regulation. Today's smart homes are also focusing on living green, and thus, trying to conserve energy, in addition to controlling homes remotely using smartphones, automating home lights and thermostats, and scheduling appliances [2]. What we expect to see in the near future is the real evolution of Smart Homes; we may be living in such homes.

The technology used in a Smart Home may employ all of the following elements: intelligent control, home automation and internal networks [1]. Following are some of the main aspects of Smart Home:

- The intelligent control is a control system that is composed of two parts: sensors, which will monitor and report the status, and a controller (human or software based) which will process the information given by the sensors, and complete the requested operation.
- The home automation function is achieved by electrical or electronic devices that will make changes to the existing environment by completing the required tasks.
- The required home automation commands are often sent over a cloud system or smartphone application.
- The home network will be the medium for all other parts of the system to communicate and send information.

The system may get multiple repetitive tasks over time, such as turning the light on after the sunset, and may require the whole cycle to repeat itself every day.

In section II we will present the state of the art on Smart Homes; section III covers market research about Smart Switch; section IV talks about securing Smart Home; section V outlines the development of the Smart Power Switch; section VI details the testing results; and finally we conclude in section VII.

II. STATE OF THE ART

Authors in [5] report that lot of research is being done by Information Technology (IT) developers on Smart Homes, in the context of in-home services. Research articles on Smart Homes and their users are on the rise; authors in [6] present a methodical analysis of research themes as well as the linkages and disconnects among them. However, clarity on the use of Smart Home technologies is still missing. Information Technology designers and inventers are always striving for better energy management and improved

 ISBN: 978-1-61208-583-8

functionality to meet the needs and requirements of homes called Smart Homes. A state-of-the-art Decision Support and Energy Management System (DSEMS) for residential energy users is proposed in [7]. The DSEMS is represented as a finite state machine and consists of a series of situations that are chosen based on user preferences. The paper also presents various designing and testing approaches and their results to show savings in energy usage and continuity of electricity supply. A new decision support tool for residential consumers is proposed in [8]. The tool can be used to optimize electrical energy services thus exploiting benefits to scheduling of energy services. A load promise framework that automatically changes electrical loads in order to minimize household use of energy resources and payment is proposed in [9].

III. SMART POWER SWITCH

Trends in the development of communication and electronic industry indicate that hundreds of smart communicating devices in future may stretch the functionalities of current devices. This may lead to creating the potential to change the way we work, learn, entertain, live, and innovate. Some researchers stated that by the year 2032 we could individually be in contact with 3000 to 5000 smart devices in our everyday life, and that would be truly transformative. In the following sections we will provide market research about smart switches and will attempt to give a comparison between a few similar products currently in the market.

A. WeMo® Light Switch

WeMo Light Switch [10] replaces a standard light switch in your home and can be controlled remotely with an Android Smartphone or Tablet, iPhone, iPad, or iPod touch. It can work with any existing Wi-Fi® network and anywhere your smartphone or tablet has an Internet connection (3G or 4G LTE). Table 1 lists Pros and Cons of WeMo Light Switch.

TABLE 1: PROS AND CONS OF WEMO LIGHT SWITCH

Pros	Cons
Remote Control	Doesn't Support 2/4 way Switches
Wall Wired Light Control	Only Compatible with 120vAC
Android/IOS Applications	Bad Performance Applications
Sunrise/Sunset Rules	No Power Monitoring
Compatible with IFTTT	No Web Application
	No Lights Dimming

B. ZULI

The Zuli Smartplug [11] communicates with a smartphone using Bluetooth Low Energy, giving users unmatched control, monitoring and automation at an affordable price. Table 2 summerizes Pros and Cons of Zuli Smartplug.

Table 2: PROS AND CONS OF ZULI SMARTPLUG

Pros	Cons
Location Based Automation using BLE	Cannot Control Wall Wired Lights/Appliances
Energy Monitoring	Cannot Control it Form Outside the House
Lights Dimming	IOS only app
Scheduling	Require Phone that support Bluetooth 4.0
Reusability	Compatible with USA plugs
	Only Compatible with 120vAC

IV. SECURING OUR SMART HOME

One of the main components for creating and managing a Smart Home is the control system that can be an embedded device. Embedded devices and systems have wide-ranging applications in residential consumer, commercial, industrial, automotive, health-care, and many other industries. Generally, embedded device are designed for a specific application; thus it has operating system or firmware for only designed applications. These devices are very small with low power consumption and little computing power. For example, a heart rate monitor embedded within a wristwatch can be connected to a smart phone for displaying the heart's status in real time; Point of Sale (POS) and Automatic Teller Machines (ATM) are also examples of embedded devices or systems. Many factors impact the process of choosing the best control system including cost and complexity of installation in addition to the best technologies available [2].

Cyber Security is one of the serious concerns in reference to Smart Homes, since information and control is wireless and over the internet. Our proposed system will be using wireless communication protocols to give devices and users the ability to send and receive information between each other as well as control home appliances. This may make the whole system very vulnerable to hackers' attacks. Thus the devices should be installed in the home in a secure way, ensuring that these are not visible to any intruder. Therefore, we implemented the Near Field Communication (NFC) protocol that allows these devices to establish peer-to-peer communication, thus enabling them to transfer data among each other by either touching them or placing them in very close proximity to each other. The device will not reveal

 ISBN: 978-1-61208-583-8

its id and pair with any smartphone until the user gets their phone and passes it on the switch; then the switch will reveal its id address and pair with the user's phone. Access to this device through any phone is not possible if the device is not registered by the owner of the device, thus preventing intruders from communicating with the device. However, as soon as the user establishes communication with the system, the device id of the user is exposed to the hackers or intruders thus raising security concerns. This problem is tackled by encrypting data at the sender device and then decrypting it at the destination device and vice versa. The encrypting technique we use depends on the machine-to-machine communication protocol (M2M).

V. SMART POWER SWITCH SYSTEM

This section outlines the development of the Smart Power Switch system, one of the important components of the smart system. We attempt to identify the user profile and requirements, for the Smart Power Switch system, by considering the findings outlined in a survey reported in [4]. In general, a typical family household may have children, are technologically savvy, as well as have a smartphone or computer and an Internet connection. These stated user profiles are not a must for every household; these are the more likely users that may be interested in living in a Smart Home. While this system is generally meant to be used in family households, it can also be used in various other environments. For example, companies, schools, and any other facility may want to turn their buildings into smart buildings. The stakeholders will be the same, but may be under different names, as there will be administrators, and users who will likely be using some parts of the system. The aim of this paper is to present a Proof of Concept (POC) Smart Power Switch system that gives users the ability to turn their home into a smart, internet-connected home by replacing the standard home switches and outlets with the smart ones. Moreover, by using the system we can control home lights/appliances remotely, schedule tasks for devices, and monitor the power consumption.

The Smart Power Switch system is divided into three main parts, The Application, The Cloud, and The Smart Switch, as shown in Fig. 1 [12]. There are two ways to communicate between the application and the switch. The application layer will be able to communicate with the smart switch through internet. In the case of controlling the switch remotely, a request submitted by the application over the internet to the cloud; the cloud in turn will send the command to the switch. The second scenario is the exact opposite: the energy consumption sensor located inside the smart switch will calculate the consumed energy and submit data to the cloud; that data will be processed in the cloud; and the analysis of the data collected will be sent back to the application when the user requests it [12]. In the second case, the user must be in the same local area network to which the smart switch is connected, and then the user will be able to control the switch locally without the need of the internet connection. The controlling events will be saved locally in the smart device, and later when there is an internet connection it will be sent to the cloud for processing.

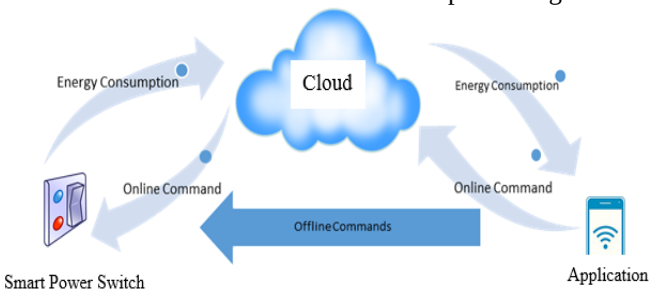

Figure 1: Components of Smart Power Switch System

VI. TESTING

In this section, we are going to report the testing carried out with the Smart Power Switch system. It provides details of the test plan, running the system and a summary of the results and the outcomes of the test. Testing plans for the system include: operating it around the clock, controlling it by a mobile app, and then connecting one TV to a standard power outlet, while the TV under test is connected to a smart sensor. The system will switch off the device once it is in standby mode, and the system reports all data to the cloud. After programming the cloud side of the system, an account is created on IBM Bluemix Platform, and services such as Node-RED Service, Cloudant Service, Internet of Things Foundation Service, or Node.js Application Service are set up. For testing with only one device, the processing power required is not great: The requirement for a system under consideration is 512MB per Instance per day, up to 20GB data storage and up to 500,000 API calls per month.

Programming the android application was accomplished with Android Studio, which allows a memory monitoring thread to run that keeps monitoring the memory and, in turn, reporting the results in a separate window. There are three possible authentication protocols to verify the communications between the app and the server: authenticating requests using Cloudant account credential, authenticating requests using auto-generated API keys, and authenticating requests from a users database. Authentication using users' database is the best security, because there are different access authorizations for different users. However, this has security which is a high priority requirement for our system.

Through the testing process, there were a few problems. One problem was that the fuse inside the smart switch was capable of handling currents under 5 Amps whereas our system was able to handle current up to 10 Amps. In one of

 ISBN: 978-1-61208-583-8

the testing processes, the fuse blew because the current was greater than 5 Amps. Therefore the fuse was replaced with a 10 Amp fuse. In Tables 3 and 4, we have provided the summary of the testing process, each with its rating out of 3, PASS/FAIL, and some notes.

TABLE 3: DEVICE SUMMARY

P/F	Rating	Notes
P	1	The devices was power by the current coming from the AC/DC converter successfully
P	1	The devices was controllable form the mobile app
P	2	The device calculated the current with 80% accuracy
P	1	The device published data to event topic successfully
P	3	The scheduled commands runs but with some delays

TABLE 4: CLOUD SUMMARY

Name	P/F	Rating	Notes
Running the System	P	1	The cloud system runs perfectly during the testing process period
Provisioning the Devices	F	-	Provisioning devices require more complex communication between the app and the device. So we hardcoded the device data in the cloud system.
Handling Requests	P	1	The system received all request via the MQTT broker.
Fast Data Querying	P	1	The Mapreduce function works efficiently and responded with the correct data.
Security	P	3	The cloud rejects requests coming from unauthorized users and securely transmits data in and out.

The testing process proved that the system is reliable and works efficiently. The results show, in Fig. 2, how implementing the smart power saving feature reduced the power consumed up to 20% per Wh.

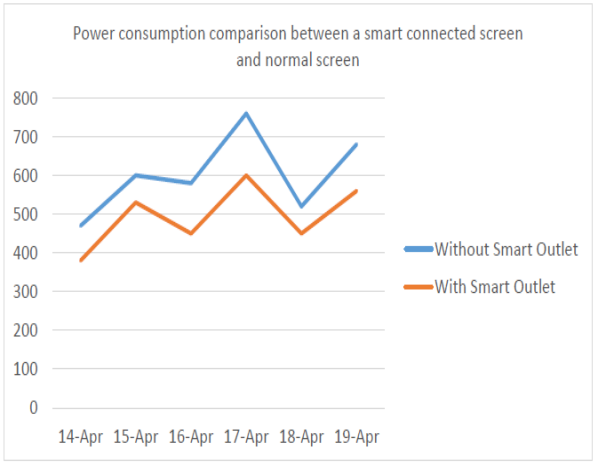

Figure 2: Comparison between a smart connected screen and a normal screen

VII. CONCLUSION AND FUTURE WORK

In this research, we defined the main proposes and objectives of our work and provided research about Smart Homes, we realize that the number of devices used in a household is increasing, and in the next few years we may have hundreds of devices accessible per person. We discussed how hard it is today to transform our life into a smart, internet-connected life. After implementing and testing the system for a few days in real life situations and comparing the devices to other Smart Homes systems, the most important finding is that it is possible to transform a normal home into a Smart Home by only changing the house power plugs and switches, as compared to complicated approaches presented by other researchers. Using our system to create Smart Homes provides a much lower cost solution than implementing traditional Smart Homes systems that may require in-wall wiring, and changing appliances to smart appliances. Transforming a home into Smart Home can benefit us in many different ways. Smart Homes can reduce energy usage up to 30% percent for normal users and can be a nice and comfortable place for elderly and disabled people. The system can also be scaled for huge buildings without compromising the same fast performance. This system worked as required for the proof-of-concept stage, however, for mass production the system requires more research in many areas including AI. In our project, we implemented a small set of AI code into the system, which resulted in a large benefit and reduced energy consumption. Additional work needs to be done to further improve the smartness of the system, and to enable it to make smart decisions securely and correctly. The hardware part of the prototype device also needs additional work leading to Design for Manufacturing (DFM).

 ISBN: 978-1-61208-583-8

VIII. ACKNOWLEDGEMENT

The work reported in this paper is based on the graduate project report by Mr. Firas AlMannna [12]. The authors would like to thank him for his hard work. We would also like to thank reviewers for their comments that helped us improve the manuscript.

REFERENCES

[1] Jiang, L., Liu, D. and Yang, B., "Smart Home research, Machine Learning and Cybernetics," Proceedings of 2004 International Conference, vol.2, pp. 659-663, 26-29 Aug. 2004

[2] Gann, D., Barlow,J. and Venables, T. "Making Homes Smarter" Chartered Institute of Housing, 1999.

[3] Pragnell, M., Spence L., Moore R. (2000) "The market potential for Smart Homes", Joseph Rowntree Foundation, York, UK, http://www.jrf.org.uk/sites/files/jrf/1859353789.pdf, July 2017.

[4] Poll, H., "Survey Reported on behalf of Lowe's Companies, Inc." Jun, 2014, http://www.prnewswire.com/news-releases/cost-confidence-and-convenience-lowes-survey-reveals-americans-attitudes-on-the-smart-home-272853581.html, July 2017.

[5] Vimarlund, V. and Wass, S. "Big Data, Smart Homes and Ambient Assisted Living", IMIA Yearbook of Medical Informatics, pp. 143-149, 2014.

[6] Wilson, C., Hargreaves, T. and Hauxwell-Baldwin, R., "Smart Homes and their users: a systematic analysis and key challenges", Personal and Ubiquitous Computing, Vol 19(2), pp. 463-476, February 2015.

[7] Siano, P., Graditi, G., Atrigna, M., Piccolo, A., "Designing and testing decision support and energy management systems for Smart Homes", Vol. 4, pp. 651-661, 2013

[8] Pedrasa, M., Spooner, T. and MacGill, I., "Coordinated scheduling of residential distributed energy resources to optimize Smart Home energy services", IEEE Transactions on Smart Grid, Vol. 1(2), pp. 134–143, September 2010.

[9] Rastegar, M., Fotuhi-Firuzabad, M. and Aminifar, F., "Load commitment in a Smart Home", Elsevier Applied Energy, Vol., pp. 45–54, Augist2012.

[10] WeMo Light Switch, Thunderbolt Technology, http://www.belkin.com/us/p/P-F7C030/ , July 2017.

[11] ZULI Smartplug, https://zuli.io/smartplug/, Zuli Inc., July 2017.

[12] AlManna, F., "Smart Power Switch", Graduate Project Report, Umm Al-Qura University, Saudi Arabia, November, 2014.

 ISBN: 978-1-61208-583-8