INNOV 2015

The Fourth International Conference on Communications, Computation, Networks and Technologies

ISBN: 978-1-61208-444-2

November 15 - 20, 2015

Barcelona, Spain

INNOV 2015 Editors

David Musliner, SIFT, LLC, USA

Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance (HLRN), Germany

INNOV 2015

Forward

The Fourth International Conference on Communications, Computation, Networks and Technologies (VALID 2015), held on November 15 - 20, 2015 in Barcelona, Spain, aimed at addressing recent research results and forecasting challenges on selected topics related to communications, computation, networks and technologies.

Considering the importance of innovative topics in today's technology-driven society, there is a paradigm shift in classical-by-now approaches, such as networking, communications, resource sharing, collaboration and telecommunications. Recent achievements demand rethinking available technologies and considering the emerging ones.

The conference had the following tracks:

☐ Networking

☐ Telecommunications

We take here the opportunity to warmly thank all the members of the INNOV 2015 technical program committee, as well as the numerous 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 INNOV 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the INNOV 2015 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that INNOV 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the areas of communication, computation, networks and technologies. We also hope Barcelona provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

INNOV 2015 Advisory Chairs

Pascal Lorenz, University of Haute-Alsace, France
Eugen Borcoci, University Politehnica of Bucharest, Romania

INNOV 2015

Committee

INNOV 2015 Technical Program Committee

Omar Alhazmi, Taibah University, Saudi Arabia
Alargam Elrayah Elsayed Ali, University of Khartoum, Sudan
Wan D. Bae, University of Wisconsin-Stout, USA
Henri Basson, University of Lille North of France, France
Michael Bauer, The University of Western Ontario, Canada
Khalid Benali, LORIA - Université de Lorraine, France
Eugen Borcoci, University Politehnica of Bucharest, Romania
Albert M. K. Cheng, University of Houston, USA
Grzegorz Chmaj, University of Nevada - Las Vegas, USA
Li-Der Chou, National Central University, Taiwan
Morshed U. Chowdhury, Deakin University-Melbourne Campus, Australia
Matteo Dell'Amico, EURECOM, France
Jacques Demongeot, IMAG/University of Grenoble, France
Uma Maheswari Devi, IBM Research, India
Tarek El-Ghazawi, George Washington University, USA
Mohamed Y. Eltabakh, Computer Science Department - Worcester Polytechnic Institute, USA
Agata Filipowska, Poznan University of Economics, Poland
David A. Gustafson, Kansas State University, USA
Fred Harris, University of Nevada - Reno, USA
Houcine Hassan, Universitat Politecnica de Valencia, Spain
Pao-Ann Hsiung, National Chung Cheng University, Taiwan
Shih-Chang Huang, National Formosa University, Taiwan
Yo-Ping Huang, National Taipei University of Technology, Taiwan
Sajid Hussain, Fisk University, Nashville, USA
Tazar Hussain, King Saud University (KSU) - Riyadh, Kingdom of Saudi Arabia
Wen-Jyi Hwang, National Taiwan Normal University, Taiwan
Sergio Ilarri, University of Zaragoza, Spain
Abdessamad Imine, LORIA-INRIA, France
Wassim Jaziri, Taibah University, Saudi Arabia
Miao Jin, University of Louisiana - Lafayette, USA
Eugene John, University of Texas at San Antonio San Antonio, USA
Khaled Khankan, Taibah University, Saudi Arabia
Igor Kotenko, St. Petersburg Institute for Informatics and Automation, Russia
Raquel Trillo Lado, University of Zaragoza, Spain
Marcela Castro León, Universitat Autònoma de Barcelona, Spain
Emilio Luque, University Autonoma of Barcelona (UAB), Spain
Xun Luo, Qualcomm Research Center, USA
Leslie Miller, Iowa State University, USA
Maria Mirto, University of Salento - Lecce, Italy

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

Performance Analysis of Encrypted Code Analyzer for Malicious Code Detection

Daewon Kim, Yongsung Jeon, and Jeongnyeo Kim
Cyber Security Research Department
Electronics and Telecommunications Research Institute
Daejeon, Korea
emails: {dwkim77, ysjeon, jnkim}@etri.re.kr

***Abstract*—Signature-based malicious code detection systems cannot in real-time detect unknowns, such as polymorphic and metamorphic codes, which can be used as zero-day attacks. More serious situation is that many automated engines easily generate new malicious codes without the attacker's special knowledge. We have already proposed a method to detect polymorphic parts of suspicious packets in anomalous network traffic. In this paper, we introduce the experiments and analysis to show the real field effectiveness and performance of our method.**

Keywords-zero-day attack; malicious code; polymorphic code; unknown attack; intrusion prevention system.

I. INTRODUCTION

Static analysis methods [2] to detect polymorphic exploit codes can be avoided by exploits using static analysis resistant techniques, which includes disassembly thwarting and self-modifying code techniques. To catch the techniques, dynamic analysis methods that directly emulate the instructions of packets include full dynamic analysis methods [3], which use a CPU emulator, and a hybrid analysis method [4], which uses both static and dynamic analyses.

Full emulation methods have an advantage in that they can detect most encrypted malicious codes. However, the overhead of emulating instructions makes it difficult to apply to real high-speed networks. A hybrid method offers better performance than a full method because the starting point of emulation can be selected through the support of a static analysis. However, hybrid methods are still insufficient for real networks owing to the complicated operational process of a static analyzer and an emulator.

Our previous work [1] showed that it can detect the decryption routine using the disassembly thwarting and self-modifying techniques. In this paper, we will present more practical examples and experiment results to show real field effectiveness.

The rest of the paper is organized as follows. In Section 2, we overview our method and describe the operation steps. In Section 3, we show our evaluation results. Finally, we conclude the paper in Section 4.

II. ENCRYPTED CODE ANALYZER

In this section, we will present the overview and example of our previous work [1] to help the understanding of our experiment results.

A. Overview

Our encrypted code analyzer detects the loop code instructions in an encrypted exploit code to decrypt the encrypted code itself. Normally, for ease of development, the decryption routine of an encrypted code stores the current program counter value on the stack and uses the value as the address for accessing the memory of an encrypted original code. Our previous work includes four kinds of components.

Firstly, seed detector detects the instructions loading the current program counter, which is a base value, into the stack. The Register Loading Base Value (RLBV) detector traces a register loading the program counter value on the stack. The Memory access Using Base Value (MUBV) detector traces the movements of registers including the base value between instructions. Lastly, loop detector determines the existence of a loop code if the final register traced by the MUBV detector is used for the instructions to access memory.

B. Operation Process

Non-malicious codes normally don't hide their original codes. One of new techniques for avoiding signature-based detection systems is code encryption. Encrypted malicious codes, which are polymorphic codes, have any code routine for decrypting to original codes at run-time. Our analyzer is based on the special patterns of decryption routine codes. If there are non-malicious codes that have similar behaviors to malicious codes, it needs more time and analysis to classify those. The cases are out of scope for our analyzer targeting real-time detection.

The analyzer in Figure 1(a) detects the seed instructions that store the address value related to the current program counter into a stack memory. The address indicates the start address of encrypted codes, which is called the base address. If the seed instructions are detected, the analyzer generates a virtual stack to trace the operations of the stack with the *base* address and in (b) detects a register loading the base address. After that, (c) the analyzer traces the movements of the *base* address from the first register, and (d) checks whether the final register with the base address is used for accessing a

 ISBN: 978-1-61208-444-2

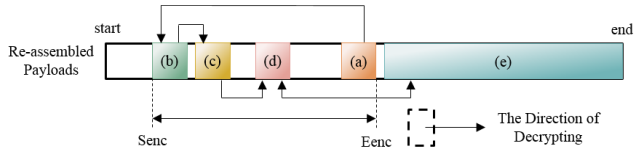


Figure 1. Encrypted code analyzer: (a) seed codes, (b) RLBV codes, (c) MUBV codes, (d) loop codes, and (e) encrypted codes.

valid memory address. The detected codes of (d) decrypt the encrypted codes of (e).

The seed detector finds the instructions shown in Figure 1(a) and creates a virtual stack. The instructions frequently used for a seed are call, fsave, fnsave, fstenv, and fnstenv. As an example, if fnstenv [esp-0c] is detected, the base address is on the top of the stack and is written on the created virtual stack. To decrypt the encrypted codes, the loop codes load the base address into a register using instructions such as pop esi. The RLBV detector traces the position of the base address stored on the real stack using a virtual stack that is operated by instructions such as push/pop, inc/dec/sub/add, and mov. Finally, the detector determines the last register using the base address.

A register with the base address is referenced for accessing the address range of the encrypted code. Attackers can move the base address to other registers to hide the memory accesses referenced by the register with the base address. The MUBV detector expresses the movements between registers as the connection graph for inspecting the register relations. Through this graph, the detector can determine a final register with the base address. The loop detector analyzes the instructions for accessing any range of memory with the base address included in the detected register. One case of instructions is xor byte ptr [ecx+esi-1],93, and our detector analyzes the validity of the address range. If the instruction accesses the memory range near the re-assembled payloads, the detector determines that the payloads are encrypted malicious codes and reports the start-position, Senc, and end-position, Eenc, to our signature generation system. Our previous work [1] described the encrypted code analyzer in greater detail.

III. EXPERIMENTS

Our previous work showed the detection rate, false positive rate, and performance of our encrypted code analyzer. For evaluating the detection rate, we used four kinds of polymorphic generation tools, and thirteen kinds of polymorphic generation engines. Our analyzer archived a 100% detection rate for all polymorphic codes that include disassembly thwarting and self-modifying code techniques.

Figure 2 shows several detection results against the encryption routines that were generated from the use of Linux/x86/shell_bind_tcp exploit. Moreover, the low instruction traversing counts indicate that the detection speed of our analyzer is similar to other static methods.

Figure 3 shows the processing overhead estimated under the system for a 3.2 GHz Pentium 4 processor with 4 GB of RAM on Cent OS (kernel version 2.6.9). The sample is network traffic captured as pcap files in a university that has

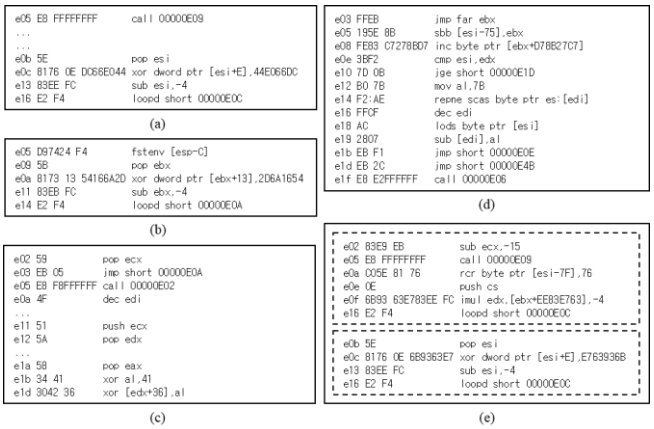


Figure 2. The decryption routines detected by the encrypted code analyzer: (a) Call4DwordXor, (b) FnstenvMov, (c) PexAlpha-Num, and (d) NonAlpha (e) Pex.

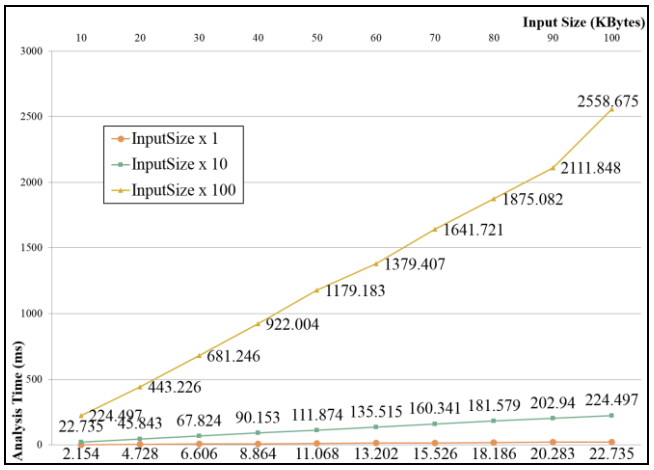


Figure 3. The processing overhead.

policies for clean network. The C function gettimeofday() was used for evaluating each processing overheads. The y-axis analysis time was calculated as the difference of full_time, which includes pcap_parsing+disassemble+detection, and disassemble_time, which includes pcap_parsing+disassemble. The analysis time is the detection time of our analyzer and it means the RLBV+MUBV+loop detector time except for the seed detector.

The results show a linear increasing trend similar to the static methods. Normally, exploit code size is small under a few tens of kilobytes. It means that the result of InputSize x 1 is useful to show a linear overhead. If the bad cases that suspicious traffic is continuously analyzed as back-to-back are considered, other two graphs are useful to show to maintain a linear increasing of analysis time. At present, we guess that the abnormal increase between the last two points, which are 2111.848 and 2558.675, on Input Size x 100 is occurred by any buffer problems between our analyzer and disassembler [5].

 ISBN: 978-1-61208-444-2

IV. CONCLUSIONS

For the detection of polymorphic codes, we have already proposed a new static analysis method for detecting self-contained polymorphic codes using static analysis resistant techniques. In this paper, we overviewed the main functions and presented experiments to show the real field effectiveness of the proposal.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R-20150518-001267, Development of Operating System Security Core Technology for the Smart Lightweight IoT Devices].

REFERENCES

[1] D. Kim, I. Kim, J. Oh, and H. Cho, "Lightweight Static Analysis to Detect Polymorphic Exploit Code with Static Analysis Resistant Technique," Proc. of IEEE ICC, June 2009, pp. 904-909.

[2] R. Chinchani and E. V. D. Berg, "A Fast Static Analysis Approach to Detect Exploit Code Inside Network Flows," Proc. of RAID, Sep. 2005, pp. 284-308.

[3] M. Polychronakis, K. Anagnostakis, and E. Makatos, "Emulation-based Detection of Non-self-contained Polymorphic Shellcode," Proc. of RAID, Sep. 2007, pp. 87-106.

[4] Q. Zhang et al., "Analyzing Network Traffic to Detect Self-decrypting Exploit Code," Proc. of ACM ASIACCS, Mar. 2007, pp. 4-12.

[5] Libdasm – A Disassembly Library. [Online]. Available from: https://code.google.com/p/libdasm/. 2015.06.16.

 ISBN: 978-1-61208-444-2

Testing Data Set for Analyzing Behaviors of Malicious Codes

Youngsoo Kim, Jungtae Kim and Ikkyun Kim
Cyber Security Research Laboratory
Electronics & Telecommunications Research Institute
Daejeon, Korea
e-mail: {blitzkrieg, jungtae_kim, ikkim21}@etri.re.kr

Abstract— Cyber targeted attack has sophisticated attack techniques using malwares to exploit vulnerabilities in systems and external Command & Control (C&C) servers are continuously monitoring and extracting data off a specific target. Since this attacking process is working continuously and uses diverse malicious codes and attacking routes, it is considered to be difficult to detect in advance. The paper proposes an indirect analytical method based on the Testing Data Set (TDS) that includes various malware behaviors for detecting cyber attacks. Especially, the proposed TDS contains both network and host dataset by running recently collected malwares in a secure testbed environment for collecting specific behaviors of the malware infections and activations. Such a combination of the self-generated datasets provides a valuable information source for the malware behavior analysis.

Keywords-Malicious Code; Behavior-based Analysis; Testing Data Set; Host-based Malicious Behavior; Network-based Behavior

I. INTRODUCTION

Recently, a common phenomenon of recently increasing cyber attacks tends to focus on a specific target with preparations for a long period. Such complicated and stealthy attacks, known as the Advanced Persistent Threat (APT), normally begin with a customized malware penetration for a target system infection in order to collect internal information of the target system. The malwares in the infected system continuously communicate with the Command & Control (C&C) servers to transmit the internal system information to hackers that was collected in advance. Then, the hacker can control the target system as well as other surround servers and resources by transferring additional malwares. Finally, the hacker achieves an intended purpose, such as the destruction of a target system, or to take the financial gains, with acquired controls over the target system and terminates the attack without leaving any evidences for the traces.

Many researches and studies are on-going to find and prevent such sophisticated APT attacks in advance that are found to be the main cause of the recent cyber incidents. Since most of the APT attacks begin with a malware infections, therefore studies for the detection of anomalies in the system caused by APT is also being actively conducted. Generally, those analysis methods used to predict the anomalies are being investigated through either the static or dynamic analysis of malwares. Also, there is an indirect analysis method, which is based on the data generated in a secure environment, for the malware infections and activations.

The paper proposes an indirect analytical method based on the Testing Data Set (TDS) that includes various malware behaviors. For the conventional Network TDS, it is not easy to get a useful dataset due to the different types of network attacks happening in the heterogeneous environments. Also, the Internet Server Providers (ISP) are not able to publically open the network TDS, that are collected for traffic analysis for the management purposes, due to the subscriber privacy policy. In case when the dataset is available, the network payload that contains personal and classified information such as IP address is removed or modified, which becomes an improper dataset for security analysis. Table I presents various types of the network TDS that are used for the attack analysis [1]-[5]. In case of the Host TDS, which obviously contains behavioral information for PC users including application usages and processes details, it was not easy to find a proper dataset to analyze. Consequently, the proposed testbed generates self-generated TDS which contains both network and host dataset by running collected malwares. The self-generated combination of both Host and Network TDS provides a valuable information source for the malware behavior analysis before the collected malware patches can be available to prevent collection of active behaviors of the malware.

TABLE I. NETWORK TDS

NAME	KDD CUP 1999	MIT LINCOLN LAB	NLANR	CAIDA	SONY MAWI
ATTACKS	DOS, BACKDOOR, BUFFER OVERFLOW	DOS, DDOS	SLAMMER, CODE-RED	WITTY	SLAMMER , WITTY
PACKET SIZE	HEADER LENGTH	NO LIMIT	HEADER LENGTH	N/A	96 BYTES
USAGE	PERFORMANCE EVALUATION FOR IDS		TRAFFIC COLLECTION, ANALYSIS		
COLLECTING TIME	1999	1998~2000	2001	2001	1999~PRESENT
AMOUNTS (MB)	17	150~200	25~40	1	100~150
FORMAT	PCAP	PCAP	TSH	TEXT	PCAP

 ISBN: 978-1-61208-444-2

Especially in the past, it was common to use a Virtual Machine (VM) to evaluate the runtime malware behaviors. As malwares were, discovered recently, programmed to circumvent the VM environment, therefore we apply a system reboot software to roll-back the system into an initial clean state when every time malwares are executed.

The rest of the paper consists of three sections. It begins with the Section II, which contains a proposed TDS generation environment in details including a network configuration with testbed components, detailed contents of the collected host and network TDS. And a method to collect the recent malwares and setup Host PCs to maximize the malware activation rate as well as a method of collecting and storing data generated by the actual executions of malicious codes will be described in the Section III. Finally, Section IV concludes a paper after reviewing and evaluating the resulting TDS.

II. TDS GENERATION ENVIRONMENT

We generated normal and abnormal data separately to get the TDS. The normal data could be a collection of host data generated from computers, e.g., name of process or triggering time of a specific event, while users doing their normal job with computers, without infection of malwares. It also includes the network traffic data generated from devices linked with network, e.g., source IP address, destination port, or starting time of a specific session, simultaneously with the above host data.

An abnormal data could be generated after activating malicious codes. We have gathered diverse malicious codes in advance and executed them automatically to get an abnormal data sets which includes both host data and network traffic data.

A. Network Architectures and Components

The TDS generation environment proposed has a different approach to get the both normal TDS and abnormal TDS. The following Figure 1 depicts network architecture of our testbed for collecting the normal and abnormal TDS. This testbed is designed as similar as to a general network architecture in real world. It includes general components, e.g., user computers, a web server, switches, routers, and a database server. It also contains a notebook for an attacker, C&C servers, a PMS (Patch Management System) server, and redirection server for support advanced attacks. To make this testbed to be the same as real enterprise network architecture, we include commercial security devices, e.g., IPS (Intrusion Prevention System), WAF (Web Application Firewall), and UTM (Unified Threat Management).

To collect TDS by generating two types of data, we need the following components including User Computers (UC), Host Data Collector (HDC), Network Data Collector (NDC), Network Traffic Collecting Device (NTCD), Servers for Network Drive (SND), and Malware Crawler (MC). We installed a specialized software on the Host Data Collector (HDC) at UCs and activated it to get the TDS of user PCs.

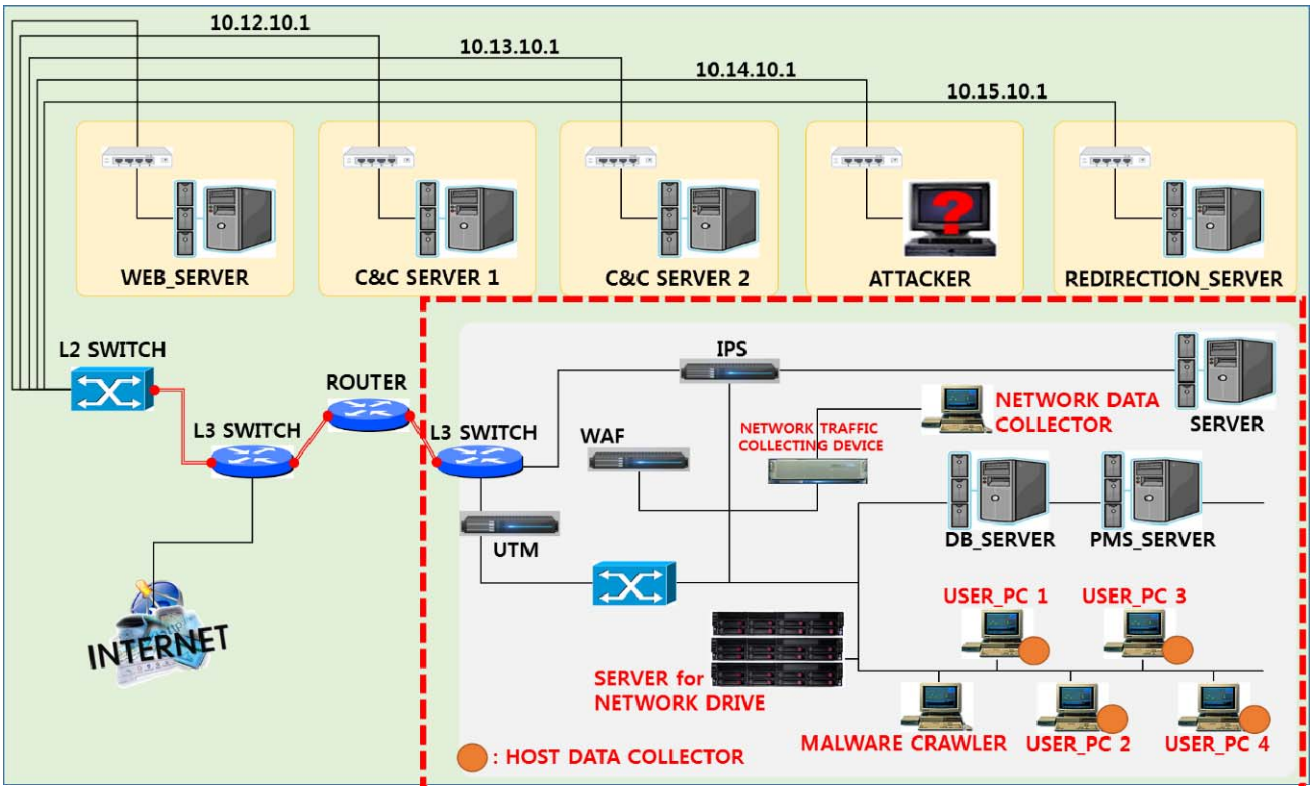


Figure 1. A Testbed Configuration of the TDS Generation Environment.

 ISBN: 978-1-61208-444-2

The UCs must be connected with the SND for transporting the collected host TDS. The HDC is an agent program for logging host data including the triggering time of a specific event, event user, process name, process identity, parent process identity, threat identity, event number, event name, API name, path of a specific process, parameters for calling APIs, etc. The logged data will be saved into a form of binary file (.DAT) once in a minute.

For the Host TDS, a set of binary files will be stored in a local directory temporally and moved to a specific folder of the network drive. We can select APIs for logging and filter some processes not to be saved selectively. The NDC collects and stores network data from the NTCD that tapped into a local switch. The network data includes Connection Start time, Connection End tine, Source IP Address, Destination IP Address, Source PORT, Destination PORT, Protocol, Inbound Flow Bytes, Outbound Flow Bytes, Numbers of Inbound Packets, Numbers of Outbound Packets, Service Name and Service Provider information and etc.

The NTCD includes DPI (Deep Packet Inspection) engines and flow collector which is based on multicore-processors. It can control asymmetric traffics using clustering technologies, support to set up controlling policies and to subscribe server-based session statistics [6]. The SND is a kind of storage server for the hosts TDS, which is stored in local directory of user's computer temporally. We can set up this network drive at the Windows Explorer of UCs. The MC is automated programs that create copies of malwares from some malware sample sites, e.g., *Vxvault* or *Malshare*. Resulting logs for visiting web sites are stored at *Mongodb* and malware information can be stored at *MySQL* [7][8].

B. Contents of TDS

The Testing Data Set (TDS) contains many useful data for the security analysis. We selected some items primarily for analyzing status of the host computer system and networks, e.g., malware behaviors or abnormal network flows. But, it can be applied to many fields for security analysis, since it includes the real case normal and abnormal host and network data which can be correlated in the both time and IP addresses. For example, an identified malware process with a PID, local IP and port information can help to find overall network connectivity of the malware to the external networks.

Following Table II and Table III show a set of collected items of each host and network data respectively [9]. Firstly, the collected items of the host data includes an Index Number of Event, Triggering Time of Event, Event User, Process Name, Process Identity, Parent Process Identity, Threat Identity, Event Number, Event Name, Windows API Name, Process Path, and Parameter of Calling APIs.

Also, the collected items of network data includes a Connection Start time, Connection End time, Source IP Address, Destination IP Address, Source PORT, Destination PORT, Protocol, Inbound Flow Bytes, Outbound Flow Bytes, Numbers of Inbound Packets, Numbers of Outbound Packets, Service Name and Service Provider information.

TABLE II. COLLECTED ITEMS OF HOST DATA

Collected items	Description
Index	Sequence number of a specific event
Time	Triggering time of a specific event
User ID	Event user
Process Name	Process name
PID	Process identity
PPID	Parent process identity
TID	Thread identity
Event Number	Event Number
Event Name	Event Name
API Name	Name of Windows API
Path	Path of a specific process
Parameter	Parameters for calling APIs

TABLE III. COLLECTED ITEMS OF NETWORK DATA

Collected items	Description
startTime	Starting time of a specific session
endTime	Terminating time of a specific session
Duration	Duration of a specific session
srcIp	Source IP address
destIp	Destination IP address
srcPort	Source port number
destPort	Destination port number
tcpFlag	TCP flag
protocol	Protocol name
inpkts	The number of input packets
outpkts	The number of output packets
inbytes	The amount of bytes for input packets
outbytes	The amount of bytes for output packets
service	Service name
device	Device name,
Sp	Name of service provider,
Status	Normal (0) / Abnormal behavior (malware name)

 ISBN: 978-1-61208-444-2

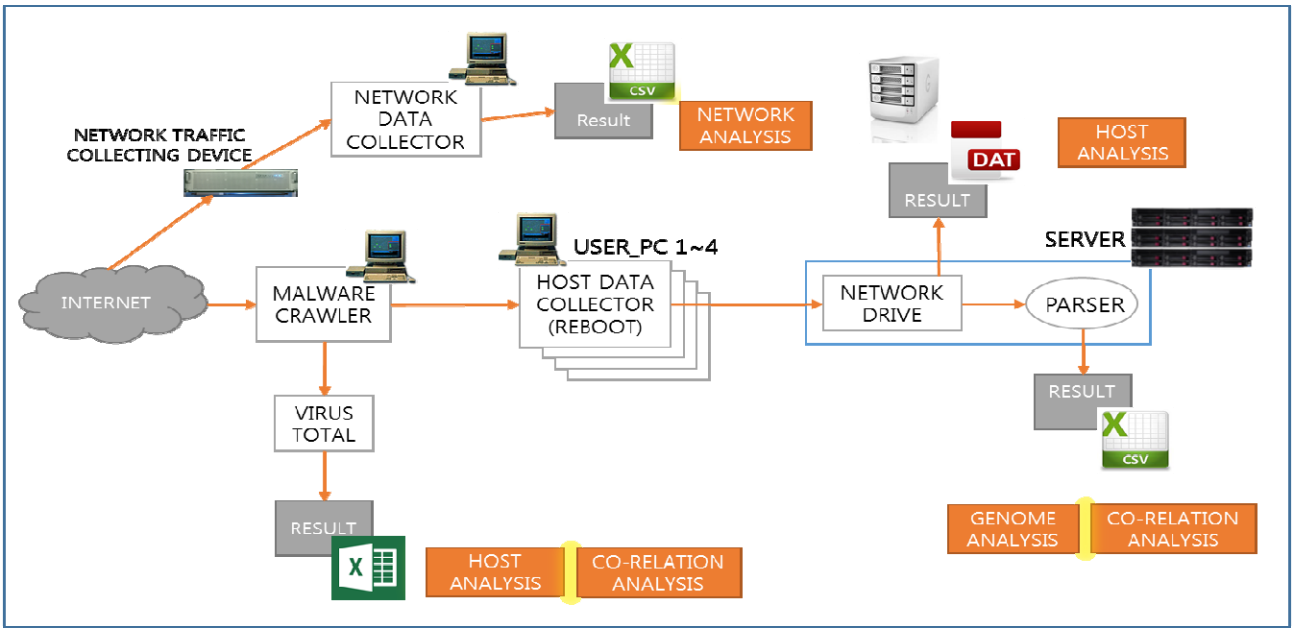


Figure 2. A Conceptual Structure for Collecting of Abnormal Data.

III. METHOD OF COLLECTING TDS

A. *Collection of Recent Malwares*

The malwares, recently founded, should be collected and managed to generate an abnormal data continuously. The MC creates copies of malwares from the malware sample sites including the *Vxvault* and *Malshare*, and stores them. It also manages malware related information using a database. The resulting logs for visiting the web sites are stored at the *Mongodb* and malware's information can be stored at the *MySQL*.

TABLE IV. TABLE STRUCTURE FOR STORING MALWARES' INFORMATION

Field	Type	Description
no	int(11)	Sequence number
type	text	Name of malware sample site
reg_time	text	Stored time
malware_name	text	Name of malware
download_link	text	Download URL
status	text	Status (Wait/Fetched/Completed/stop)
analysis_ip	text	IP address of performing analysis
start_time	text	Starting time of analysis
end_time	text	End time of analysis

The above Table IV shows a detailed information of the collected malware information with the DB table structure. After the MC copies the collected malwares from the malware sample sites, it calculates the hash values of those collected malwares and removes duplicated malwares by comparing each hash values. We use the SHA256 hash algorithm for hashing [10].

When a malware is registered for the first time, the initial status information is "Wait", and the client program in the HDC copy the target malware to activate, resulting status changed to the "Fetched" state. The status of malwares can be changed as "Completed" in case of finishing analysis or activation, and as "Stop" in case of being interrupted because of some errors.

B. *Collection of Abnormal Data*

Figure 2 depicts a conceptual structure for collecting the abnormal data. The HDC programs installed at 4 UCs are activated to get the TDS for host data. It copies malwares those status are in "Wait" state from the MC and activates them one by one. The logged data will be made into a form of binary file (.DAT) in every minute. The Host TDS, a set of binary files, will be stored in local directory temporally, and moved to a specific folder of the network drive.

The binary files are generally used for a host data analysis, but should be transformed to the CSV file format using a parser for the cyber-genome analysis and co-relation analysis. We set up 4 UCs with different environmental settings, as shown in the Table V. We selected applications for installation by referencing the CVE (Common Vulnerabilities and Exposures) lists [11].

The NTCD is positioned at a connection point between an external and internal network in order to detect malicious

 ISBN: 978-1-61208-444-2

behaviors which occur from the outside, and collects the network traffic data generated between the external and internal network.

The 20 data factors which are collected by the network data collector includes Connection Start time, Connection End time, Source IP Address, Destination IP Address, Source PORT, Destination PORT, Protocol, Inbound Flow Bytes, Outbound Flow Bytes, Numbers of Inbound Packets, Numbers of Outbound Packets, Service Name and Service Provider information.

TABLE V. DIFFERENT SETUPS OF 4 UCS

	User PC_1	User PC_2	User PC_3	User PC_4
OS	windows 7 SP1 32bit		windows 7 SP1 64bit	
Installed Applications	IE 10	IE 11	IE 10	IE 11
	Flash Player 14	Flash Player 15	Flash Player 16	Flash Player 17
	Acrobat Reader 10	Acrobat Reader 11	Acrobat Reader 10	Acrobat Reader 11
	.NET			
	Google Chrome 43.X			
	SDK			
	Silver Lite			
	MSOffice 2003 SP2 / MSOffice 2007 SP3 / MSOffice 2013 SP1			
	HWP 2010/2014, 7zip, Nateon, Alftp, Mplayer, Notepad, Putty, MS Outlook, Outlook Express, cmd, telnet, utorrent, gzip, vim, Wordpad, Kakaotalk, Facebook, windows media player, GOMaudio, Google Drive, gimp, Filezilla, Smplayer, Xmplay, pnotes, Naver Streaming Service, Stickies, Cpu-z, Freecommander, Apache, Unitaller			

After it collects the network flow data from the edge routers and it sends the data which should be included in payload defined UDP packet to NDC for a network topology processing. NDC makes and stores network TDS locally as a format of CSV file. It can be used for network analysis. By inputting a hash value of a specific malware to VirusTotal site, MC gets some diagnosis results that over 40 vaccine companies have [12]. It can be referred to host analysis or co-relation analysis.

For the evaluation purpose, we have collected and generated a TDS by executing 3,392 malwares during two weeks periods. Each host collector is configured to collect the process information for 5 minutes until malwares were properly executed, then the notification of malware process terminations were forwarded to the NDC.

For recursive testing of heterogeneous malwares, the Comback 7.0 was used to support automatic reboot of the system to its original state after collecting the malware process information [13]. For the network TDS, the collected binary data is converted to the CSV file of 434 MB for analysis, and 6.7 TB were collected for the binary type host TDS.

The host TDS were also converted to a CSV file type with a specific parser developed for easy data analysis to be used in the Cyber Gene and Correlation analysis. Additionally, the hash values of malwares names from the VirusTotal were generated as an Excel format for analysis.

Following Figure 3 and 4 represents a sample host and network TDS collected respectively.

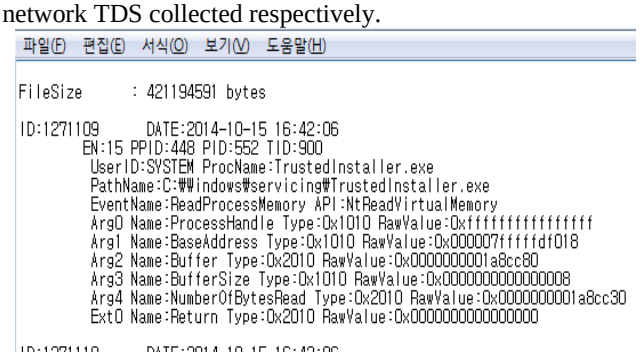


Figure 3. Host TDS

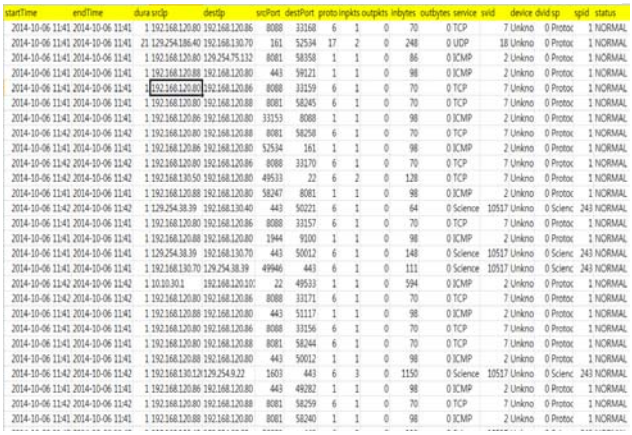


Figure 4. Network TDS

IV. CONCLUSION AND FUTURE WORK

To cope with the increasing advanced cyber attacks and to overcome the limitations of the conventional Network TDS, the paper proposed an indirect analytical method for detecting advanced cyber attacks (APT) by proposing a collection method of the Testing Data Set (TDS) that includes various malware behaviors. To do so, a testbed was designed to suit with the real network environments and various types of recent malware were collected for evaluating the resulting dataset collected.

The self-generated dataset collects predefined 12 and 17 detailed components information of the Host and Network TDS respectively. The combination of both Host TDS and Network TDS provides a valuable information source for the malware behavior analysis. For the future works, a TDS collection for the known malware with behavior information will provides a useful insight for malware analysis which helps to create categorized datasets based on the different types of the malware behaviors.

 ISBN: 978-1-61208-444-2

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.B0101-15-1293,Cyber-targeted attack recognition and traceback technology based on the long-term historic analysis of multi-source data).

REFERENCES

[1] The UCI KDD Archive, "KDD Cup 1999 Data," available at http://www.ics.uci.edu/~kdd/databases/kddcup99/kddcup99.html [retrieved: Oct, 2015]

[2] Lincoln Laboratory Massachusetts Institute of Technology, "MIT Lincoln Laboratory – DARPA Intrusion Detection Evaluation Data Sets," available at http://www.ll.mit.edu/IST/ideval/data/data_index.html [retrieved: Oct, 2015]

[3] NLANR Measurement and Network Analysis Group, "NLANR PMA," available at http://pma.nlanr.net [retrieved: Oct, 2015]

[4] Cooperative Association for Internet Data Analysis, "Cooperative Association for Internet Data Analysis (CAIDA)," available at http://www.caida.org [retrieved: Oct, 2015]

[5] MAWI Working Group, "MAWI Working Group Traffic Archive," available at http://tracer.csl.sony.co.jp/mawi/ [retrieved: Oct, 2015]

[6] PacketLiner EL480, http://sysmate.com [retrieved: Oct, 2015]

[7] Vxvault, http://vxvault.net/ViriList.php [retrieved: Oct, 2015]

[8] Malshare, http://malshare.com/ [retrieved: Oct, 2015]

[9] D. Moon, H. Lee, and I. Kim, “Host based Feature Description Method for Detecting APT Attack,” Journal of The Korea Institute Of Information Security & Cryptology, Vol. 24, No. 5, Oct. 2014, pp. 839-850, ISSN: 1598-3986.

[10] S. Lee, D. Choi, and Y. Choi, “Improved Shamir’s CRT-RSA Algorithm: Revisit with the Modulus Chaining Method,” ETRI Journal, Vol. 36, No. 3, Jun. 2014, pp.469-478, ISSN: 1225-6463.

[11] Common Vulnerabilities and Exposures, http://www.cvedetails.com/ [retrieved: Oct, 2015]

[12] Virus Total, https://www.virustotal.com/ [retrieved: Oct, 2015]

[13] Comback, http://www.wowcomback.com/comback/combackIntro.asp [retrieved: Oct, 2015]

 ISBN: 978-1-61208-444-2

FUZZBOMB: Autonomous Cyber Vulnerability Detection and Repair

David J. Musliner, Scott E. Friedman, Michael Boldt, J. Benton, Max Schuchard, Peter Keller
Smart Information Flow Technologies (SIFT)
Minneapolis, USA
email: {dmusliner,sfriedman,mboldt}@sift.net
Stephen McCamant
University of Minnesota
Minneapolis, USA
email: mccamant@cs.umn.edu

***Abstract*—Beginning just over one year ago, Smart Information Flow Technologies (SIFT) and the University of Minnesota teamed up to create a fully autonomous Cyber Reasoning System (CRS) to compete in the Defense Advanced Research Projects Agency (DARPA) Cyber Grand Challenge (CGC). Starting from our prior work on autonomous cyber defense and symbolic analysis of binary programs, we developed numerous new components to create FUZZBOMB. In this paper, we outline several of the major advances we developed for FUZZBOMB, and review its performance in the first phase of the CGC competition.**

Keywords-autonomous cyber defense; symbolic analysis; protocol learning; binary rewriting.

I. INTRODUCTION

In June 2014, DARPA funded seven teams to build autonomous CRSs to compete in the DARPA CGC. SIFT and the University of Minnesota (UMN) together formed the FUZZBOMB team, building on our prior work on the FUZZBUSTER cyber defense system [1] and the FuzzBALL symbolic analysis tool [2].

SIFT's FUZZBUSTER system automatically finds flaws in software using symbolic analysis tools and fuzz testing, refines its understanding of the flaws using additional testing, and then synthesizes *adaptations* (e.g., input filters or source-code patches) to prevent future exploitation of those flaws, while also preserving functionality. FUZZBUSTER includes an extensible plug-in architecture for adding new analysis and adaptation tools, along with a time-aware, utility-based meta-control system that chooses which tools are used on which applications during a mission [3]. Before the CGC began, FUZZBUSTER had already automatically found and shielded or repaired dozens of flaws in widely-used software including Linux tools, web browsers, and web servers.

In separate research, Prof. Stephen McCamant at UMN had been developing the FuzzBALL tool to perform symbolic analysis of binary x86 code. FuzzBALL combines static analysis and symbolic execution to find flaws and proofs of vulnerability through heuristic-directed search and constraint solving. On a standard suite of buffer overflow vulnerabilities, FuzzBALL found inputs triggering all but one, many with less than five seconds of search [2].

Together, FUZZBUSTER and FuzzBALL provided the seeds of a strategic reasoning framework and deep binary analysis methods needed for our FUZZBOMB CRS. However, many challenges still had to be addressed to form a fully functioning and competitive CRS. In this paper, Section II describes the CGC competition, Sections III and IV overview our prior components, Section V outlines several of the major advances we developed for FUZZBOMB, and Section VI reviews its performance in the first phase of the CGC competition.

II. DARPA'S CYBER GRAND CHALLENGE

Briefly, the CGC is designed to be a simplified form of Capture the Flag game in which DARPA supplies Challenge Binaries (CBs) that nominally perform some server-like function, responding to client connections and engaging in some behavioral protocol as the client and server communicate. The CBs are run on a modified Linux operating system called Decree, which provides a limited set of system calls. In the competition, CBs are provided as binaries only (no source code) and are undocumented, so the CRSs have no idea what function they are supposed to perform. However, in some cases a network packet capture (PCAP) file is provided, giving noisy, incomplete traces of normal non-faulting client/server interactions ("pollers"). Within each CB is one or more vulnerability that can be accessed by the client sending some inputs, leading to a program crash. To win the game, a CRS must find the vulnerability-triggering inputs (called Proofs of Vulnerability (PoVs)) and also repair the binary so that the PoVs no longer cause a crash, and all non-PoV poller behavior is preserved. The complex scoring system rewards finding PoVs, repairing PoVs, and preserving poller behavior, and penalizes increases in CB size and decreases in CB speed.

III. BACKGROUND: FUZZBUSTER

Since 2010, we have been developing FUZZBUSTER under DARPA's Clean-Slate Design of Resilient, Adaptive, Secure

 ISBN: 978-1-61208-444-2

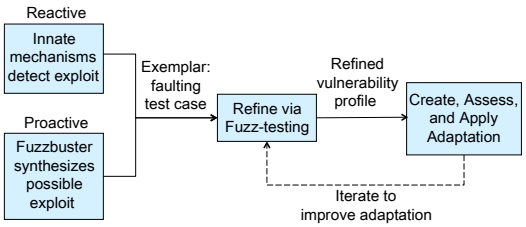


Figure 1. FUZZBUSTER refines both proactive and reactive fault exemplars into vulnerability profiles, then develops and deploys adaptations that remove vulnerabilities.

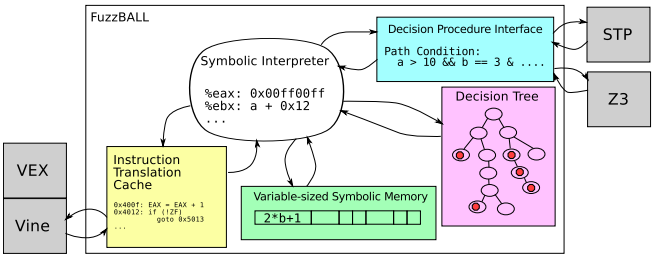


Figure 2. An overview of our FuzzBALL binary symbolic execution engine.

Hosts (CRASH) program to use software analysis and adaptation to defeat a wide variety of cyber-threats. By coordinating the operation of automatic tools for software analysis, test generation, vulnerability refinement, and adaptation generation, FUZZBUSTER provides long-term immunity against both observed attacks and novel (zero-day) cyber-attacks.

FUZZBUSTER operates both *reactively* and *proactively*, as illustrated in Figure 1. When an attacker deploys an exploit and triggers a program fault (or other detected misbehavior), FUZZBUSTER captures the operating environment and recent program inputs into a *reactive exemplar*. Similarly, when FUZZBUSTER's own software analysis and fuzz-testing tools proactively create a potential exploit, it is summarized in a *proactive exemplar*. These exemplars are essentially tests that indicate a (possible) vulnerability in the software, which FUZZBUSTER must characterize and then shield from future exploitation. For example, an exemplar could hold a particular long input string that arrived immediately before an observed program fault.

Starting from an exemplar, FUZZBUSTER uses its program analysis tools and fuzz-testing tools to refine its understanding of the vulnerability, building a *vulnerability profile* (VP). For example, FUZZBUSTER can use concolic testing to find that the long-string reactive exemplar is triggering a buffer overflow, and the VP would capture this information. Or, FUZZBUSTER can use delta-debugging and other fuzzing tools to determine the minimal portion of the string that triggers the fault.

At the same time, FUZZBUSTER tries to create software adaptations that shield or repair the underlying vulnerability. In the simplest case, FUZZBUSTER may choose to create a filter rule that blocks some or all of the exemplar input (i.e., stopping the same or similar attacks from working a second time). This may not shield the full extent of the vulnerability (or may be too broad, compromising normal operation), so FUZZBUSTER will keep working to refine the VP and develop more effective adaptations. Even symbolic analysis may not yield a minimal description of the inputs that can trigger a vulnerability: there may be many vulnerable paths, only some of which are summarized by a constraint description. Over time, as FUZZBUSTER refines the VP and gains a better understanding of the flaw, it may create more sophisticated and effective adaptations, such as filters that block strings based on length not exact content, or actual software patches that repair the buffer overflow flaw.

While FUZZBUSTER already had the coordination infrastructure and representation/reasoning to manage exemplars, VPs, and adaptations, many of the tools we had integrated could not apply to the CGC because they do not operate directly on binaries. To fill these gaps and support the full spectrum of vulnerability detection, exploitation, and repair needed for CGC, we integrated with UMN's FuzzBALL and also developed new components, as described below.

IV. BACKGROUND: FUZZBALL

FuzzBALL is a flexible engine for symbolic execution and automatic program analysis, targeted specifically at binary software. In the following paragraphs we briefly describe the concepts of symbolic execution and explain FuzzBALL's architecture, emphasizing its features aimed at binary code.

The basic principle of symbolic execution is to replace certain *concrete* values in a program's state with *symbolic variables*. Typically, symbolic variables are used to represent the inputs to a program or sub-function, and the symbolic analysis results in an understanding of what inputs can lead to different parts of a program. An interpreter executes the program, accumulating symbolic expressions for the results of computations that involve symbolic variables, and constraints (in terms of those symbols) that describe which conditional branches will occur. These symbolic expressions are valuable because they can summarize the effect of many potential concrete executions (i.e., many possible inputs). When a symbolic expression is used in a control-flow instruction, we call the formula that controls the target a *branch condition*. On a complete program run, the conjunction of the conditions for all the symbolic branches is the *path condition*. We can use an Satisfiability Modulo Theories (SMT) solver (such as Z3 [4]) on a path condition to find a set of concrete input values that would cause the corresponding path to be executed, or to determine what other paths might be feasible.

Many symbolic execution tools operate on program source code (e.g., KLEE, Crest), but FuzzBALL is differentiated by its focus on symbolic execution of binary code. At

 ISBN: 978-1-61208-444-2

its core, FuzzBALL is an interpreter for machine (e.g., x86) instructions, but one in which the values in registers and memory can be symbolic expressions rather than just concrete bit patterns. Figure 2 shows a graphical overview of FuzzBALL's architecture. As it explores possible executions of a binary, FuzzBALL builds a *decision tree* data structure. The decision tree is a binary tree in which each node represents the occurrence of a symbolic branch on a particular execution path, and a node has children labeled "false" and "true" representing the next symbolic branch that will occur in either case. FuzzBALL uses the decision tree to ensure that each path it explores is different, and that exploration stops if no further paths are possible.

We have used FuzzBALL on several CGC-relevant research projects, which typically build on the basic FuzzBALL engine by adding heuristics or other features specialized for a particular problem domain. For FUZZBOMB and the CGC, we integrated FuzzBALL with the FUZZBUSTER reasoning framework and significantly extended FuzzBALL's program analysis capabilities.

V. NEW DEVELOPMENTS

A. Hierarchical Architecture

We designed FUZZBOMB to operate on our in-house cluster of up to 20 Dell Poweredge C6100 blade chassis, each holding eight Intel XEON Harpertown quad-core CPUs. To allocate this rack of computers, we designed a hierarchical command-and-control scheme in which different FUZZBOMB agents play different roles. At the top of the hierarchy, several agents are designated as "Optimus", or leader agents. At any time, one is the primary leader, known as Optimus Prime (OP). All of the other Optimi are "hot backups," in case OP goes down for any reason (hardware failure, software crash, network isolation). All messages sent to OP are also sent to all of the other Optimi, so that their knowledge is kept up to date at all times. We enhanced our existing fault detection and leader election protocol methods to ensure that an OP is active in the cluster with very high reliability. We usually configure FUZZBOMB with three Optimi, each run on a different hardware chassis in the cluster.

Below OP, a set of "FUZZBOMB-Master" agents are designated, each to manage the reasoning about a single CB. OP's main job is allocating CBs to those Master agents and giving them each additional resources (other FUZZBOMBS, DVMs) to use to improve their score on a CB. A FUZZBOMB-Master's job is improve its score on its designated CB, using its allocated computing resources in the best way possible (whether that is analysis, rewriting, or testing/scoring). As progress is made on each CB, the responsible FUZZBOMB-Master will report that progress and the best-revised-CB-so-far back to OP.

OP's objective is to maximize the system's overall score, keeping in mind deadlines and other considerations. By design, OP should dynamically re-allocate the reasoning assets to the most challenging problems, to maximize the overall system's score. OP is also responsible for uploading FUZZBOMB's final best answers to the competition.

B. FuzzBALL Improvements

FUZZBOMB uses an improved FuzzBALL symbolic execution engine in an approach that combines ideas from symbolic execution and static analysis in order to find vulnerabilities in binary programs. A static-style analysis identifies parts of the program that might contain a vulnerability. Then a symbolic execution search seeks an execution path from the start of the program to the possible vulnerability point that constitutes a proof of vulnerability. Symbolic execution generates a number of input constraint sets, each set representing a family of related program execution paths. The symbolic execution engine uses these constraint sets to determine the inputs to the program that can reach the program vulnerability, offering a proof-of-concept exploit. While exploring this space, the symbolic execution engine will encounter many decision points (such as conditional branches). Each of these decision points branches off a new set of paths, leading to an exponentially growing number of paths. Exploring this search space of paths represents a significant computational effort. Scaling up the search in a way that mitigates this path explosion poses a key challenge. To overcome this problem, we applied parallelization techniques and heuristic search improvements, as well as other algorithmic changes.

1) Heuristic Guidance: Because the space of program executions is vast, even in the constraint-based representations of symbolic reasoning, heuristic guidance is essential. For the CGC, the key objective is to guide the search towards potential vulnerabilities. FUZZBOMB identifies potentially vulnerable instruction sequences and uses abstraction heuristics to focus the search towards those targets. Although a wide variety of source-level coding mistakes can leave a program vulnerable, these dangerous constructs are more uniform when viewed in terms of the binary-level capability they give to an attacker. For example, many types of source-code vulnerabilities create binary code in which the destination of an indirect jump instruction can be influenced by an attacker. The source-code and compiler details about why such a controllable jump arises are often irrelevant, and are not our focus. In particular, FUZZBOMB does not try to decompile a binary back to a source language, nor will it identify which particular source code flaw describes a vulnerability. FUZZBOMB's search guidance strategies target just these end-result capabilities, e.g., searching for an indirect jump that can be controlled to lead to attack code.

FUZZBOMB uses ***problem relaxation heuristics*** to reduce the search space of possible executions, drawing on recent advances in heuristic search techniques for directed symbolic execution and Artificial Intelligence (AI) planning.

 ISBN: 978-1-61208-444-2

To search through very large spaces, these techniques use rapid solutions to relaxed or approximate versions of their real problems to provide heuristic guidance. Over the last dozen years, research on relaxation heuristics has produced immense improvements in the scalability of AI planning and other techniques. For example, Edelkamp *et al.* [5] report up to four orders of magnitude reduction in nodes searched in model-checking. Similarly, AI planning systems have gone from producing plans with no more than 15 steps to plans with hundreds of steps (representing many orders of magnitude improvement in space searched). These techniques are only now being applied to directed symbolic execution to help find program paths to vulnerabilities (e.g., Ma *et al.* [6]).

For FUZZBOMB, the problem is to find a symbolic execution path through a program that leads to a vulnerability. One key research challenge is finding the best relaxation method for symbolic execution domains. We developed an approach using causal graph heuristics found in AI planning search [7] to direct symbolic execution, in a manner similar to call-chain backwards symbolic execution [6]. These heuristics use factorization to generate a causal model of subproblems, then "abstract away" interactions between the subproblems to create a relaxed version of the problem that can be solved quickly at each decision point during search. In symbolic execution, solving the relaxed problem determines:

- A reachability analysis to a vulnerability. If the relaxation of the program indicates a vulnerability is unreachable from a particular program decision point, then exploring from that point is fruitless.
- A distance estimate at each decision point that lets exploration proceed along an estimated shortest path.

To generate the relaxation heuristic, FUZZBOMB uses the causal model present within data-flow and control-flow graph (CFG) structures used in binary program analysis. For instance, in a CFG, nodes represent blocks of code and edges represent execution order. This provides a subproblem structure, allowing for bottom-up solving of each subproblem.

2) Other Improvements: The FuzzBALL approach to hybrid symbolic execution and static analysis needed many other improvements to work on the CGC CBs. Our major developments have included:

- Porting to Decree— We adapted FuzzBALL to handle the unique CB format, including emulating the restricted Decree system calls and handling the specific limitations of the CB binary format.
- Improving over-approximated CFG methods— Prior to symbolic analysis, FuzzBALL requires the control flow graph (CFG) of the target binary. Various existing methods are all imperfect at recovering CFGs, but some can be combined. We developed a new CFG-recovery tool that leverages prior work on recursive disassembly along with an updated over-approximation method that finds all of the bit sequences in a binary representing valid addresses/offsets within the binary and treats those as possible jump targets. While this overapproximation is extreme, FUZZBOMB uses heuristics to reduce the size of the resulting CFGs.
- Detecting input-controllable jumps— As FuzzBALL extends branch conditions forward through the possible program executions, whenever it reaches a jump it formulates an SMT query asking whether the CB inputs could force the jump to 42 (i.e., an arbitrary address). If so, a likely vulnerability has been identified.
- Detecting null pointer dereferences, return address overwrites, and various other vulnerable behaviors.
- Making incremental solver calls— We have enhanced FuzzBALL's SMT solver interface so that it can behave incrementally. For example, after querying if a jump target is input-controllable, it can retract that final part of the SMT query and the SMT solver can retain some information it derived during the prior solver call. Microsoft's Z3 SMT solver is state of the art and supports this type of incremental behavior.
- Handling SSE floating point (FP)— The original FuzzBALL implementation used a slow, emulation-based method to handle floating point calculations, and it could not handle the modern SSE FP instructions. We have recently completed major extensions that allow FuzzBALL to handle SSE FP instructions using Z3. We have switched over to using Z3 by default, and are collaborating with both the Z3 and MathSAT5 developers to fix bugs in their solvers and improve their performance.
- Implementing veritesting— David Brumley's group coined this term for a flexible combination of dynamic symbolic execution (DSE) and static symbolic execution (SSE) used to reason in bulk about blocks of code that do not need DSE [8]. We completed our own first version of this capability, along with associated test cases and SMT heuristic improvements. However, as noted below, this improvement was not used during the actual competition because its testing and validation was not complete.

Symbolic execution can be expensive because it is completely precise; this precision ensures that the approach can always create proofs of vulnerabilities. At the same time, it is valuable to know about potentially dangerous constructs even before we can prove they are exploitable. To that end, we modified FuzzBALL to run as a hybrid of static analysis and symbolic execution techniques.

C. Proofs of Vulnerability (PoVs)

We developed two ways of creating PoVs. First, when FuzzBALL identifies a vulnerability that can be triggered by client inputs, it will have solved a set of constraints on the symbolic input variables that describe a class of

 ISBN: 978-1-61208-444-2

PoVs for that vulnerability. Depending on the constraints, the PoV description may be more or less abstract (i.e., it may require very concrete inputs or describe a broad space of inputs that will trigger the vulnerability). For the concrete case, FUZZBOMB has a mechanism to translate FuzzBALL's constraints into the XML format required for a PoV.

Second, if a CB is provided with a PCAP file that illustrates how it interacts with one or more pollers, FUZZBOMB uses protocol reverse engineering techniques to derive an abstract description of the acceptable protocols for a CB. FUZZBOMB then feeds this protocol description into one or more fuzzing tools, to try to develop input XML files that trigger an unknown vulnerability.

We initially developed a protocol reverse engineering tool building on Antunes' ideas [9]. However, the techniques did not scale well to the large numbers of pollers present in the CGC example problems, and they are not robust to the packet loss present in the provided packet captures. We then developed a less elaborate protocol analysis tool which, while not providing a full view of the protocol state machine, allows FUZZBOMB to generate protocol sessions which are accepted by the CBs. This tool uses a heuristic approach, based on observations from prior work in the field [10], to identify likely protocol command elements, fields required for data delivery to the CBs (e.g. message lengths and field offsets), and message delimiters. Additionally, the protocol inference tool also attempts to identify session cookies and simple challenge/response exchanges that are required by the protocol. Significant effort was also required to process the DARPA-provided PCAP files because they contain unexpected packet losses and non-TCP-compliant behavior.

D. Binary Rewriting

We have developed a powerful binary rewriting tool suite that includes mechanism for rewriting instructions, relocating code, and inserting arbitrary code into binaries (if necessary, via trampolining) [11]. Building off of the conservative, over-approximated CFG, these rewriting tools can be used to perform a variety of proactive defensive rewrites as well as focused repairs. For example, FUZZBOMB can inject well-known techniques such as stack canaries that can protect against stack smashing and code injection attacks. We have developed a search-based method for trying different stack canary injection locations, trying to find a location for the canary and the canary-check that preserves known good functionality and defeats known PoVs.

We have also developed an "unstripping" tool that finds the unique bit patterns of the `libcgc` library calls in a binary. FUZZBOMB will use this information to identify which library calls are not used in a particular CB, and their space can be reclaimed for use by the binary rewriting tool (without expanding the size of the binary at all).

Initially, finding space to inject canaries and trampolining code was a major challenge. FUZZBOMB has three methods for finding space to use for rewriting:

- Hijacking program header segments.
- Using file space in between segments (which start on page boundaries).
- Extending a file's executable segment up to the next page boundary.

We have also developed purely defensive rewriting methods that can protect against flaws that FUZZBOMB has not yet identified. Currently, our defensive measures use heuristics to identify functions that receive input data and seem likely to contain a potential overflow flaw. The system then incrementally adds canary-based stack protection to those blocks, re-testing the resulting CB version to see if it still performs as expected. However, the performance (speed) impact of these changes is difficult to assess without many test cases. By default, FUZZBOMB chooses to add stack protection to just three target blocks, chosen heuristically.

VI. RESULTS AND CONCLUSIONS

The first year of CGC involved three opportunities to assess FUZZBOMB's performance: two practice Scored Events (SE1 and SE2) and the CGC Qualifying Event (CQE), which determined which competitors would continue to the second year of competition. In SE1, DARPA released fifteen challenge binaries, some of which had multiple vulnerabilities. At the time, FUZZBOMB had only recently become operational on our computing cluster, and it did not solve many of the problems. However, with access to the source of the SE1 examples and many bug fixes, some months later we had improved FUZZBOMB enough that it was able to find vulnerabilities in four of the problems, including at least one undocumented flaw. For each of those vulnerabilities, FUZZBOMB had a repair that was able to stop the vulnerability from being attacked while also preserving all of the functionality tested by up to 1000 provided test cases. FUZZBOMB also create defensive rewrites for all of the other binaries. In SE2, DARPA provided nine new challenge binaries in addition to the prior fifteen, giving a total of twenty-four. Each problem was supplied with either no PCAPs or a PCAP file containing up to 1000 client/server interactions. At the time of SE2, FUZZBOMB was only able to find two of the new vulnerabilities, but that performance was enough to earn fourth place, when the SE1 problems were included in the ranking.

Our progress in improving the system was slowed by major problems with the government-provided testing system: running parallel tests interfered with each other, and running batches of serialized tests could cause false negatives, hiding vulnerabilities. This meant we had to run tests one at a time, incurring major overhead and making test-running a major bottleneck (especially when given 1000 tests from PCAPs, or when FUZZBOMB created many tests itself). We finally resolved these issues by discarding the provided testing tool and writing our own. Our tool supported safe

 ISBN: 978-1-61208-444-2

parallel testing and increased testing speeds by at least two orders of magnitude. However, it took many weeks to come to that conclusion. Several key analysis functions were not completed, including handling challenge problems that had multiple communicating binary programs, complete support for SSE floating point instructions, and veritesting. We also were not able to build the ability to have the system reallocate compute nodes to different CBs or to different functions (DVM vs. running FuzzBALL). By the time of the CQE, in June 2015, FUZZBOMB was only able to fully solve seven of the twenty-four SE2 problems. If given the PoVs for the twenty-four problems, the repair system was able to fix twelve CBs perfectly, and the defense system earned additional points on the remaining CBs.

For CQE, DARPA provided 131 all-new problems to the twenty-eight teams who participated (out of 104 originally registered). Each problem was supplied with either no PCAPs or a single client/server interaction. Unfortunately, this singleton PCAP triggered an unanticipated corner case in FUZZBOMB's logic: the protocol analysis concluded that every element of the single client/server interaction was a constant, so the protocol had no variables to fuzz. And the default fuzz-testing patterns were not used because there *was* a protocol. Thus FUZZBOMB's fuzzing was completely disabled for those challenge problems. Also, because the reallocation functionality was not available, we had to preallocate the number of DVMs vs. FuzzBALL symbolic search engines. We chose to use 325 DVMs and only 156 FUZZBOMBS, because testing had been such a bottleneck. However, since there were almost no test cases provided in the PCAP files and fuzzing was disabled, FUZZBOMB had very few tests to run, and the DVMs were largely idle. With most CBs having only a single FuzzBALL search engine, there was little parallel search activity, and FUZZBOMB only found vulnerabilities in 12 CBs (some using prior SE2 PoVs). Of those, with the limited testing available, repair was only able to perfectly fix six (as far as our system could tell). Defense rewrote all of the remaining problems.

When the final CQE scores were revealed, FUZZBOMB came in tenth place and did not qualify to continue in the competition (only the top seven teams qualified). In addition to the singleton PCAP files and other issues, we learned of another "curveball" when the scores were released: among the 131 test cases, there were 590 known vulnerabilities, an average of more than 4.5 flaws per binary. In hindsight, FUZZBOMB's defensive system should have been much more aggressive in adding blind checks, to try to capture some points from all of those flaws. Our conservative rationale had been that retaining performance was more important, but with that many flaws per CB, the balance is changed. Even so, defensive rewriting earned FUZZBOMB more points than its active analysis and repair capability. This result supports our notion that CGC-relevant flaws boil down to a small number of patterns in binary, and can be addressed with a small number of repair/defense strategies.

Fortunately, the story is not over for FUZZBOMB; we have other customers who are interested in the technology, and we are actively pursuing transition opportunities to more real-world cyber defense applications.

ACKNOWLEDGMENTS

This work was supported by DARPA and Air Force Research Laboratory under contract FA8750-14-C-0093. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

REFERENCES

[1] D. J. Musliner et al., "Fuzzbuster: Towards adaptive immunity from cyber threats," in Proc. SASO-11 Awareness Workshop, October 2011, pp. 137–140.

[2] D. Babić, L. Martignoni, S. McCamant, and D. Song, "Statically-directed dynamic automated test generation," in Proceedings of the ACM/SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Toronto, ON, Canada, Jul. 2011, pp. 12–22.

[3] D. J. Musliner, S. E. Friedman, J. M. Rye, and T. Marble, "Meta-control for adaptive cybersecurity in FUZZBUSTER," in Proc. IEEE Int'l Conf. on Self-Adaptive and Self-Organizing Systems, September 2013, pp. 219–226.

[4] L. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in Tools and Algorithms for the Construction and Analysis of Systems (TACAS), ser. LNCS, vol. 4963. Springer, Apr. 2008, pp. 337–340.

[5] S. Edelkamp et al., "Survey on directed model checking," in Model Checking and Artificial Intelligence, 2008, pp. 65–89.

[6] K.-K. Ma, K. Y. Phang, J. S. Foster, and M. Hicks, "Directed symbolic execution," in Static Analysis Symposium (SAS), Venice, Italy, Sep. 2011, pp. 95–111.

[7] M. Helmert, "The fast downward planning system," Journal of Artificial Intelligence Research, vol. 26, no. 1, 2006, pp. 191–246.

[8] T. Avgerinos, A. Rebert, S. K. Cha, and D. Brumley, "Enhancing symbolic execution with veritesting," in Proceedings of the 36th International Conference on Software Engineering, 2014, pp. 1083–1094. [Online]. Available: http://doi.acm.org/10.1145/2568225.2568293

[9] J. Antunes, N. Neves, and P. Verssimo, "Reverse engineering of protocols from network traces," in Proc. 18th Working Conf. on Reverse Engineering (WCRE), 2011, pp. 169–178.

[10] W. Cui, V. Paxson, N. Weaver, and R. H. Katz, "Protocol-independent adaptive replay of application dialog," in NDSS, 2006.

[11] S. E. Friedman and D. J. Musliner, "Automatically repairing stripped executables with CFG microsurgery," in Adaptive Host and Network Security Workshop at the IEEE Int'l Conf. on Self-Adaptive and Self-Organizing Systems, 2015.

 ISBN: 978-1-61208-444-2

QoS in Peer to Peer Live Streaming through Dynamic Bandwidth and Playback Rate Control

Maria Efthymiopoulou, Nikolaos Efthymiopoulos
Department of Electrical and Computer Engineering, University of Patras
Patras, Greece
Email: mefthymiop@ece.upatras.gr, nefthymiop@ece.upatras.gr

***Abstract*—Current commercial live video streaming systems are based either on a typical client-server (cloud) or on a peer-to-peer (P2P) architecture. The former is preferred for stability and QoS while the latter is scalable with small bandwidth and management cost. In this paper, we propose a scalable and stable service management architecture for a cloud assisted P2P live streaming system. In order to achieve this we develop an analytical model and a hybrid control strategy that dynamically allocates from the cloud the exact amount of bandwidth that is required while simultaneously dynamically adapts the playback rate to the available bandwidth resources in order to guarantee the complete and on time stream distribution. To the best of our knowledge our proposed model is the first that copes up with a hybrid control strategy for simultaneous playback rate adaptation and auxiliary bandwidth allocation.**

Keywords - ***peer to peer; live streaming; control theory; QoS***

I. INTRODUCTION

Video streaming has become a dominant part of today's internet traffic. As Cisco analyzes in [1] between 2012 and 2013, the highest growth happened on the Internet side in online video with 16 percent year-over-year growth. On the other hand the tremendous number of users leads even the major streaming service providers (e.g., YouTube) to suffer from high bandwidth costs and scalability issues. P2P live streaming and P2P video on demand architectures as: [5][10]-[13][16] have received a lot of research attention in the past few years. In order to reveal the importance of our study we highlight the major requirements from P2P live streaming systems which are: i) **Efficiency** of the media distribution in terms of utilization of peers' upload bandwidth, in order to minimize any additional bandwidth contributed by media servers (cloud) and/or maximize the playback rate of the stream which the system is able to deliver, ii) **Stability** of the distribution which is defined as the uninterrupted and complete stream delivery in each participating peer even in the presence of dynamic conditions (e.g., unrelated network traffic, system bandwidth changes, peer arrivals and/or departures) that affect the amount of the available upload bandwidth in the system, iii) **Scalability** which is determined by the amount of resources (bandwidth, storage, processing overhead) that cloud, which manages the system, has to contribute in order to sustain the uninterrupted delivery of the stream, as the number of participating peers grows.

In the literature, there are two strategies in order to dynamically harmonize the relationship between the playback rate and the total upload bandwidth that participating peers and cloud contribute and enable in this way efficiency, stability and scalability. The first [6] is the dynamic allocation of upload bandwidth from auxiliary sources (e.g., clouds) while the second [16] is the dynamic adaptation of the playback rate according to the existing upload bandwidth of participating peers. The selection of a strategy has to do, except the technical issues, with the desire of system's users and the business model that the service provider wants to follow. In case that users and the service provider desire a costless live steaming the second has to be selected. In case that they desire a live streaming with high stream quality the first has to be selected. In this work we propose a hybrid strategy that enables a flexible tradeoff between the advantages of the two.

Towards the first strategy, the research community has proposed monitoring systems, such as [4][6], that use statistical methods for the scalable monitoring of the total available bandwidth resources in a P2P overlay. These systems are scalable but suffer from three drawbacks which are: i) stochastic methods are suitable only for specific upload bandwidth distributions among participating peers, ii) their efficiency in terms of peer bandwidth exploitation is low due to the low confidence interval, iii) they are not stable as they do not capture the system dynamics in cases of sudden disturbances (e.g., underlying network traffic changes and/or peers' arrivals-departures).

In [14], the problem of stability is recognized and it is studied the impact of flash crowds on the stability. In [15] is studied the stability of a real P2P live streaming system and is highlighted that the server plays an indispensable role in the stability. All these works highlight the problems that occur in P2P live streaming without a QoS enabling system.

Motivated by the lack of an analytical model and a holistic study in this area, and based in our previous work [2][3], we develop a control strategy for a non-linear system that is able to dynamically allocate from the cloud the exact amount of bandwidth that is required and simultaneously adapt the playback rate to the available bandwidth resources in order to guarantee the complete and on time stream distribution for each peer.

The reminder of this paper is structured as follows: Section II presents our P2P live streaming system's architecture which

 ISBN: 978-1-61208-444-2

is our background work. Section III presents the problems that we solve. In Section IV, is analyzed the proposed scalable bandwidth and playback rate control strategy and in Section V we conclude.

II. PROPOSED SYSTEM ARCHITECTURE

Our P2P live video streaming system consists of a media server in a cloud, (noted by S) and a set of peers (noted by N). S divides the stream into video blocks and is responsible for: i) the initial diffusion of blocks to a small subset of nodes among participating peers, ii) the tracking of the network addresses of a small set of participating peers in order to assist the bootstrap of the P2P overlay, iii) the dynamic and scalable monitor of the resources of participating peers, iv) the dynamic control of auxiliary bandwidth and playback rate.

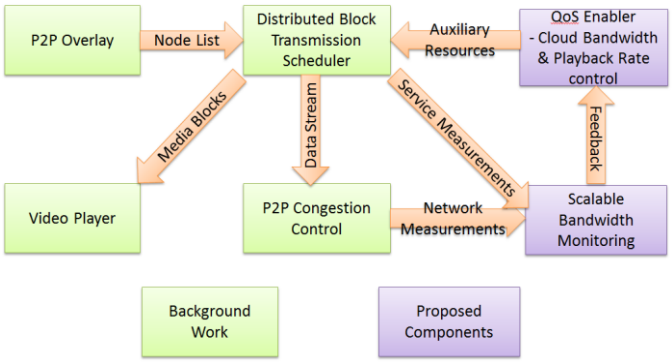


Figure 1. Proposed P2P live streaming architecture and major interactions

In order to allow peers to exchange video blocks, each peer in N maintains network connections with a small subset of other peers which will be noted as neighbors. The sets of these connections change dynamically and form a dynamic graph called the P2P overlay [2][3], which is a graph topology and P2P overlay management algorithms that each peer periodically executes. We use distributed optimization theory in order to dynamically ensure in a distributed (scalable) and dynamic fashion that: i) peers have connections proportional with their upload bandwidth, ii) peers have connections with other peers close to the underlying network, iii) our P2P overlay is adaptable to underlying network changes and peer arrivals and departures.

Distributed Block Transmission Scheduler (DBTS) [2][3] coordinates video block exchanges in a distributed fashion. In order to achieve this we developed a set of algorithms which executed by every peer in N which dynamically communicates with its neighbors in the P2P overlay. The major objective of DBTS is to ensure the timely delivery of every block to every peer by exploiting the upload bandwidth of participating peers and the additional bandwidth resources that media servers may contribute. DBTS sends the video blocks that have to be sent in the P2P congestion control component and the ordered stream with the blocks that it receives to DBTS.

These two components enhance our system with two properties that we exploit here. The first property (Property 1) is that if idle bandwidth exists it is derived from bandwidth surplus in the system and not from the inefficiency of the system to exploit it. In other words we guarantee that the presence of idle bandwidth implies the complete stream delivery. The second property is that the percentages of the idle resources among participating peers are almost equal (Property 2). We highlight here that in case of heterogeneous peer upload bandwidth various peers send with various bitrates (analog with their upload bandwidth capacity) but the percentage of their bandwidth utilization, and so the percentage of their idle time is very similar.

Our P2P congestion control mechanism [7] is able to manage sequential transmissions of video blocks to multiple locations that DBTS sends to it and to provide to the Scalable Bandwidth Monitoring the dynamic estimation of: i) the upload bandwidth capacity, ii) the idle bandwidth resources of each participating peer. In the rest of this paper, we analyze a Bandwidth Playback Rate Control (BPRC) component that acts as the QoS enabler of the P2P live streaming system. Its scalability properties are analyzed in detail in [16].

III. PROBLEM STATEMENT

We assume a set of peers N that receive the same video stream. In order to receive the stream all peers in N issue requests to their neighbors (a small subset of N) with a bit rate p_k which is the media object playback rate. The subscript k is an integer denoting the time instant.

The fulfillment of these requests generates the incoming flows. These requests are served from the same set of peers N which exploit their upload bandwidth. These are the outgoing flows in the system. By exploiting the outgoing flows P2P congestion control is able to calculate dynamically the upload bandwidth capacity, $u_{(i)k}$, and the idle percentage of the upload bandwidth capacity, $id_{(i)k}$, of each participating peer i in N.

The first problem is the development of an analytical model that connects, with an analytical relationship, the bandwidth that we have to allocate or release dynamically and the playback rate with the dynamic idle percentage of the upload bandwidth of the participating peers.

The second problem that we solve is to create a BPRC strategy with which we exploit our analytical model in order control $id_{(i)k}$ of each participating peer in N to a reference value id_{REF} by adapting dynamically, by the use of auxiliary resources (cloud), system's total upload bandwidth and playback rate. In this way, if the total upload bandwidth of the participating peers is greater than the required we dynamically estimate this surplus in order to be able to use it for other purposes (e.g., distribution of another stream) and/or increase p_k. Otherwise, if total system's upload bandwidth is less than the required, we dynamically estimate the amount of the deficit and we demand it from S in order to ensure the stability of the distribution and/or decrease p_k

IV. BANDWIDTH AND PLAYBACK RATE CONTROL (BPRC)

BPRC is a control functionality that is executed periodically, at a time instant k, with period T. It is executed in a centralized fashion by the server, S, who generates the media object that is streamed. Its objective is to set the idle time

 ISBN: 978-1-61208-444-2

percentage $id_{(i)k}$ of each peer i in N to a reference value id_{REF}, by periodically adjusting $U_{(S)k}$ and p_k. As $U_{(S)k}$ we define the total amount of upload bandwidth that should be added/removed from the P2P Overlay every time instant k that BPRC is executed. In the rest of this section we model this process analytically and we propose a control strategy with which we periodically calculate $U_{(S)k}$ and p_k. The symbols that we use are presented altogether in Table I below. Index i indicates a peer i that belongs to N and the index k represents a time instant k. In order to derive the system model we make two assumptions which are:

Assumption 1: According to Property 2 that we described in Section II we can write approximately $id_{(i)k}=id_k$, for each i belongs to N. We note that id_k represents the average $id_{(i)k}$ in N.

Assumption 2: Period T, with which BPRC is executed, is lower than the time interval that is needed for significant changes in the total upload bandwidth of participating peers. So we can do the approximation that total upload bandwidth remains similar between two consecutive executions of BPRC.

At any time instant k and in case that there are sufficient upload bandwidth resources (Property 1) is guaranteed the complete delivery of the stream to every peer in the set N and so the incoming flow to each participating peer is p_k. Thus the sum of the incoming flows of N peers is $N*p_k$. The sum of the incoming flows that peers receive is equal to the sum of the outgoing flows that peers in N contribute by using their upload bandwidth. The sum of the outgoing flows is the sum of their non-idle upload capacity $u_{(i)k}$ so we have:

TABLE I. NOTATION

Symbol	Definition
S	Generator (source) of the media object
N	Set of participating peers (in the equations below we use N as the number of participating peers)
p_k	Dynamic media playback rate at time instant k
$U_{(S)k}$	Amount of upload bandwidth that should be added/ removed from the P2P overlay at time instant k as it determined from BPRC
$u_{(i)k}$	Upload capacity (upper limit) of peer i at time instant k
$id_{(i)k}$	Idle time percentage of peer i at time instant k [0,1]
id_k	Average estimated idle time percentage of N peers at time instant k [0,1]
id_{REF}	Average idle time percentage reference value [0,1]
T	Period of execution of BPRC
m_k	System input that represents the change in the playback rate as determined from BPRC
m_{REF}	System input in the equilibrium point of BPRC

$$Np_k = \sum_{i \in N}(1 - id_{(i)k})u_{(i)k} \quad (1)$$

Under Assumption 1 we can rewrite (1) as:

$$Np_k = (1 - id_k)\sum_{i \in N} u_{(i)k} \quad (2)$$

Rewriting (2) for time instant k+1,we have:

$$Np_{k+1} = (1 - id_{k+1})\sum_{i \in N} u_{(i)k+1} \quad (3)$$

By definition at time instant k+1, total system's upload bandwidth resources, can be expressed as the sum of total system's upload bandwidth resources at time instant k plus $U_{(S)k}$. Thus, holds that:

$$\sum_{i \in N} u_{(i)k+1} = \sum_{i \in N} u_{(i)k} + U_{(S)k} \quad (4)$$

We now define the playback rate at time instant k+1, as the sum of the playback rate at time instant k p_k and w_k which is the difference that BPRC will introduce to the playback rate. Thus, holds that:

$$p_{k+1} = p_k + w_k \quad (5)$$

By using (4),(5) in (3) we have:

$$N(p_k + w_k) = (1 - id_{k+1})(\sum_{i \in N} u_{(i)k} + U_{(S)k}) \quad (6)$$

Now by dividing (2),(5) and by using Assumption 2 we have:

$$id_{k+1} = 1 + \frac{(id_k - 1)\sum_{i \in N} u_{(i)k}\left(p_k + w_k\right)}{\left(\sum_{i \in N} u_{(i)k} + U_{(S)k}\right)p_k} \quad (7)$$

We now set:

$$m_k = \frac{\sum_{i \in N} u_{(i)k}\left(p_k + w_k\right)}{\left(\sum_{i \in N} u_{(i)k} + U_{(S)k}\right)p_k} \quad (8)$$

From (7) by the use of (8) we have:

$$id_{k+1} = 1 + (id_k - 1)m_k \quad (9)$$

By setting $id_k=id_{k+1}=id_{REF}$ in (8) we obtain m_{REF} which is defined as the input, in the equilibrium point and is equal to 0. Thus, in this case arises that m_{REF} in the equilibrium point is equal to 1. In order to have a system which has as its equilibrium point (0,0) we now set:

$$x_k = id_k - id_{REF} \quad (10)$$

 ISBN: 978-1-61208-444-2

$$u_k = m_k - m_{REF} \quad (11)$$

The idle time percentage id_k belongs to the interval (0,1) by definition. Thus x_k ranges between ($-id_{REF}$, $1-id_{REF}$). By substituting (10),(11) in (9) we have:

$$x_{k+1} = 1 - id_{REF} + (x_k + id_{REF} - 1)(u_k + m_{REF}) \quad (12)$$

We observe that (12) is nonlinear for a linear closed loop system we use a feedback linearization [9] which introduces a state feedback such that the closed loop system becomes linear. To this end we select a control strategy u_k of the form:

$$u_k = \frac{(1-k_c)x_k}{k_c x_k + id_{REF} - 1} \quad (13)$$

In (12), k_c is a parameter that we choose. By combining now (10), (11) and (12) we have a system with eigenvalue k_c :

$$x_{k+1} = k_c x_k \quad (14)$$

In this way it is easy to observe from (13) that the series $\{x_k\}$ converges to 0, and so id_k converges to id_{REF} for any value k_c that belongs to (-1,1). Since k_c is a designer's choice we can explicitly set the eigenvalue of the system by just setting k_c. If we now combine (8),(10),(11) and (13) we have:

$$\frac{(1-k_c)(id_k - id_{REF})}{k_c(id_k - id_{REF}) + id_{REF} - 1} = \frac{\sum_{i \in N} u_{(i)k}\left(p_k + w_k\right)}{\left(\sum_{i \in N} u_{(i)k} + U_{(S)k}\right)p_k} - 1 \quad (15)$$

So each time that BPRC is executed, after measuring $u_{(i)k}$ and p_k, we have to select a pair of values for w_k and $U_{(S)k}$ such that (15) is satisfied. The selection of this pair has to do the desirable playback rate (p_k+w_k) and the auxiliary upload bandwidth that is allocated from the cloud or released from the P2P overlay which represented in (16). We keep the selection strategy open to any policy. As it can be seen from (15) a high playback rate will lead to a high auxiliary upload bandwidth and a low playback rate to a low auxiliary upload bandwidth.

$$\sum_{t \in (0,k-1)} U_{(S)t} + U_{(S)k} \quad (16)$$

The selection of id_{REF} has to do with the accuracy of our modeling (Assumption 1) and the adversity of the changes (disturbances) in the total upload bandwidth in the P2P overlay (Assumption 2). High inaccuracy and system disturbances need high id_{REF} (high degree of resource overprovisioning) that will guarantee uninterrupted stream delivery.

V. CONCLUSIONS

The proposed model is the first analytical model towards the simultaneous control of playback rate and auxiliary bandwidth provision in P2P live streaming. We leave the evaluation and a robust analysis of our model (to derive the minimum id_{REF} that guarantees uninterrupted stream delivery as a function of the model accuracy and the magnitude of disturbances) as future work.

ACKNOWLEDGMENT

This has been financed by STEER [7] which is an European Commission's FP7 project and by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

REFERENCES

[1] Cisco Visual Networking Index, "The Zettabyte Era—Trends and Analysis", 2015.

[2] N. Efthymiopoulos, A. Christakidis, S. Denazis, and O. Koufopavlou , "Liquidstream – network dependent dynamic P2P live streaming," Springer Peer-to-Peer Networking and Applications, vol. 4, pp. 50-62, Mar. 2011, doi: 10.1007 /s12083-010-0092-0.

[3] A. Christakidis, N. Efthymiopoulos, J. Fiedler, S. Dempsey, K. Koutsopoulos, et al., "VITAL++ a new communication paradigm: embedding P2P technology in next generation networks," IEEE Communications Magazine, vol. 49, pp. 84-91, Jan. 2011, doi: 10.1109/MCOM.2011.5681020.

[4] D. Ciullo, V. Martina, M. Garetto, E. Leonardi, and G. L. Torrisi, "Stochastic Analysis of Self-Sustainability in Peer-Assisted VoD Systems," IEEE INFOCOM, pp. 1539-1547, Mar. 2012, doi: 10.1109/INFCOM.2012.6195521.

[5] PPLive. [Online]. Available from: http://pplive.en.softonic.com/

[6] D. Ciullo, V. Martina, M. Garetto, E. Leonardi, and G. L. Torrisi, "Performance analysis of non-stationary peer-assisted VoD systems," IEEE INFOCOM, pp. 3001-3005, Mar. 2012, doi: 10.1109/INFCOM.2012.6195747.

[7] Social Telemedia Environment for Experimental Research. [Online]. Available from: http://www.steer.ics.ece.upatras.gr

[8] Opnet modeler. [Online]. Available from: www.opnet.com

[9] J.J. Slotine, and W. Li, Applied Nonlinear Control. Prentice Hall, 1991.

[10] GridCast. [Online]. Available from: http://www.gridcast.cn

[11] PPStream. [Online]. Available from: http://www.ppstream.com

[12] TVU. [Online]. Available from: http://www.tvunetworks.com

[13] SopCast. [Online]. Available from: http://www.sopcast.com

[14] Y. Chen, B. Zhang, C. Chen, and D. M. Chiu, "Performance modeling and evaluation of peer-to-peer live streaming systems under flash crowds," IEEE/ACM Networking, vol. 22, pp. 1106–1120, Aug. 2014, doi: 10.1109/TNET.2013.2272056.

[15] C. Wu, B. Li, and S. Zhao, "Diagnosing network-wide P2P live streaming inefficiencies," ACM Multimedia, Computing, and Communications, vol. 8, Article No 13, Feb. 2012, doi: 10.1145/2089085.2089090.

[16] M. Efthymiopoulou, N. Efthymiopoulos, A. Christakidis, N. Athanasopoulos, S. Denazis, and O. Koufopavlou, "Scalable playback rate control in P2P live streaming systems", Springer Peer-to-Peer Networking and Applications, pp. 1-15, Sept. 2015, doi: 10.1007/s12083-015-0403-6.

 ISBN: 978-1-61208-444-2

Motion Compensated Frame Rate Up-Conversion Using Adaptive Extended Bilateral Motion Estimation

Daejun Park
Department of Electronics and Computer Engineering
Hanyang University
Seoul, South Korea
e-mail: daejoon12@naver.com

Jechang Jeong
Department of Electronics and Computer Engineering
Hanyang University
Seoul, South Korea
e-mail: jjeong@hanyang.ac.kr

***Abstract*—In this paper, a novel frame rate up conversion (FRUC) algorithm using adaptive extended bilateral motion estimation (AEBME) is proposed. Conventionally, extended bilateral motion estimation (EBME) conducts dual motion estimation (ME) processes on the same region, therefore involves high complexity. However, in this proposed scheme, a novel block type matching procedure is suggested to accelerate the ME procedure. We calculate the edge information using sobel mask, and the calculated the edge information is used in block type matching procedure. Based on the block type matching, decision will be made whether to use EBME. Motion vector smoothing (MVS) is adopted to detect outliers and correct outliers in the motion vector field (MVF). Finally, overlapped block motion compensation (OBMC) and motion compensated frame interpolation (MCFI) are adopted to interpolate the intermediate frame in which OBMC is employed adaptively based on the frame motion activity. Experimental results show that the proposed algorithm has outstanding performance and fast computation comparing with EBME.**

Keywords- Frame rate up conversion; extended bilateral motion estimation; overlapped block motion compensation; block type matching; frame motion activity.

I. INTRODUCTION

Frame rate up conversion (FRUC) is used in various display devices with the purpose of increasing frame rates. Liquid crystal display (LCD) has annoying motion blur effect especially in sequences with dynamic motion [1]. This is due to its hold-type display characteristics which tend to sustain the light intensity for a longer moment than cathode ray tube (CRT). Viewers have difficulty in tracking a fast moving object in LCD because image from previous frame may still remain on the display. This results in annoying effect, which is called ghost effect. FRUC is the ideal technique used to counter this problem. This noticeable motion blur is resolved by doubling the frame rates. FRUC algorithm is also useful in limited bandwidth situation. In narrow bandwidth channel, encoder has to decrease transmission data rating. So encoder transfers either odd or even frames of sequence. At the decoder side, the removed frames are to be restored using FRUC technique.

Various FRUC algorithms have been developed [2]. Approaches that do not consider the motion of objects are the simplest in FRUC algorithms, e.g., frame repetition, frame averaging. These algorithms are easy to be implemented in software and hardware. However, they have problems such as motion jerkiness. To reduce these artifacts, motion compensation techniques can be applied. Such methods are called motion compensation interpolation (MCI). Motion-compensated frame rate up conversion (MC-FRUC) is a popular method in FRUC [3]. It consists of motion estimation (ME) and MCI. ME produces motion vectors (MVs) by using the block matching algorithm (BMA) for its low complexity and ease of implementation. However, BMA suffers from various artifacts, e.g., blocking artifact, halo effect.

MC-FRUC can be classified further into two approaches, true motion based type and non-true motion based type. Extended bilateral motion estimation (EBME) carries out full search on both original and intermediate grid and therefore it is classified as a non-true motion based approach [4]. EBME scheme is a slow algorithm and regarded as unsuitable for real-time applications. Nonetheless, it has the advantage of executing BME precisely twice, on two different overlapping grids. Moreover, its outlier detection and correction function is effective in smoothing out false motion vector.

The proposed adaptive extended bilateral motion estimation (AEBME) is an modified version of conventional EBME algorithm. The novel block type matching procedure is proposed to accelerate the ME procedure. We calculate the edge information using sobel mask, and the calculated edge information is used in block type matching procedure. Based on the block type matching, decoder will decide whether to use EBME. Motion vector smoothing (MVS) is adopted to detect outliers and correct outliers in the motion vector field (MVF). Overlapped block motion compensation (OBMC) technique is adopted during interpolation process to reduce the blocking artifacts that may occur due to irregularity of motion vector [5]. OBMC is employed adaptively based on frame motion activity. Finally, motion compensated frame interpolation (MCFI) is adopted to restore the missing frames.

The rest of the paper is organized as follows: Section II presents an overview of EBME algorithm. Section III describes our proposed algorithm in details. The experimental results and test conditions are provided in Section IV. Finally, we conclude the paper in Section V.

 ISBN: 978-1-61208-444-2

II. EXTENDED BILATERAL MOTION ESTIMATION

A. Bilateral Motion Estimation

The bilateral motion estimation (BME), as illustrated in Figure 1, is executed under the assumption that object motion is temporally symmetric from the intermediate frame's point-of-view.

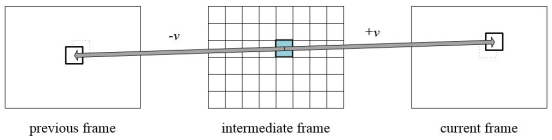


Figure 1. Illustration of the bilateral motion estimation.

The traditional BMA suffers from holes and occluded regions during compensation. However, bilateral motion estimation and compensation has no holes and occluded regions after reconstruction. In BME, the block in interpolated frame is regarded as the search center. The search is performed by comparing a block at a shifted position in the previous frame and another block at the opposite position in the current frame. We can compute the sum of bilateral absolute differences (SBAD) by (1) and find the MV which minimizes SBAD by (2).

$$SBAD(v_x, v_y) = \sum_{v_x \in SR} \sum_{v_y \in SR} \left| f_{n-1}(x - v_x, y - v_y) - f_n(x + v_x, y + v_y) \right| \quad (1)$$

$$v = \arg \min_{(v_x, v_y) \in SR} \{SBAD(v_x, v_y)\} \quad (2)$$

where (v_x, v_y) is the MV candidate, f_{n-1} and f_n are the previous and current frames, repectively. v is the selected MV, BLK is the block size, and SR is the search range.

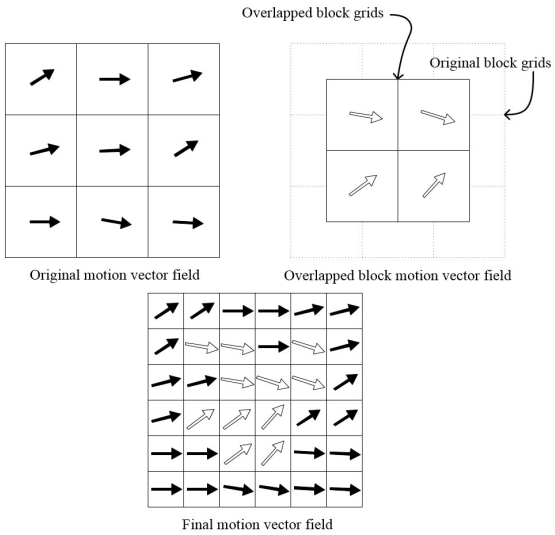


Figure 2. Illustration of the extended bilateral motion estimation.

B. Extended Bilateral Motion Estimation

The computational complexity of the BME is much lower than the quarter of the computational complexity of the BMA, because the search range of the BME is one quarter of that of BMA. However, when the motion trajectory of an object is not symmetrical from the intermediate frame's viewpoint, the true MV cannot be estimated. The EBME performs the BME for the overlapped blocks to search for a more accurate MV. Figure 2 illustrates how the EBME modifies the motion vector field, which is more precise than the original MVF. By comparing SBAD of the original block grids and the overlapped block grids, the final MVF will be decided.

C. Recursive Motion Vector Smoothing

An outlier causes the block artifact and degrades the image quality. MVS is adopted to detect outliers and eliminate outliers from the MVF.

$$v_m = \frac{1}{9} \sum_{i=1}^{9} v_i \quad (3)$$

$$D_i = abs(v_m - v_i) \quad (4)$$

$$D_n = \frac{1}{8} \sum_{i=2}^{9} D_i \quad (5)$$

In (3), v_m is an average MV of v_1 and all neighboring block's MVs surrounding v_1. In (4), D_i is the absolute difference between v_m and v_i, and D_n in (5) is the mean of the absolute difference between v_m and each of eight neighboring MVs. If $D_1 > D_n$, v_1 is an outlier.

After detecting all outliers, the smoothing process will correct them. The smoothing process is shown in Figure 3. First, it selects v_1 whose eight surrounding MVs are considered reliable. Median filtering will be employed to correct v_1. Next, it selects v_1, whose neighboring MVs contain one outlier. Median filtering will be conducted for reliable neighboring MVs. The smoothing process will be progressed by increasing the number of neighboring outliers one by one. If it cannot increase the number of neighboring outliers, the smoothing process will start over again.

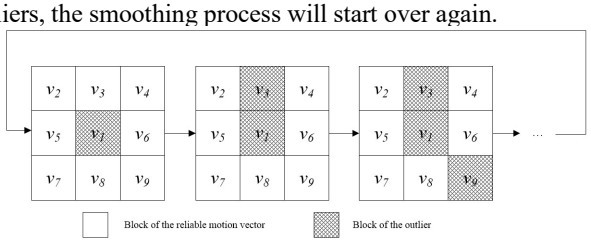


Figure 3. Recursive motion vector smoothing.

D. Overlapped Block Motion Compensation

OBMC process can drastically reduce blocking artifacts and provide a good visual quality in almost all sequences under an assumption that we have the accurate MVF. To enhance a visual quality, we employ bilinear window, illustrated in Figure 4, which is formulated as a linear estimator of pixel intensities given the limited block motion information. The coefficients of the filter are determined by (6).

$$w(u,v) = w_u \cdot w_v,\ w_u = \begin{cases} \frac{1}{N}(u+\frac{1}{2}) & for\ \ u = 0,...,N-1 \\ w_{(2N-1)-u} & for\ \ u = N,...,2N-1 \end{cases} \quad (6)$$

where N is the block size.

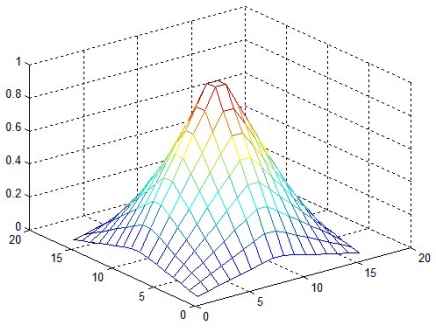


Figure 4. Bilinear window.

E. Motion Compensated Frame Interpolation

In order to construct the intermediate frame, MCFI is employed by using the final MVs. The intermediate frame is interpolated by (7). We select a block to which we want to apply MCFI, and enlarge block's size to the window size for OBMC process. Then, OBMC and MCFI are conducted.

$$f_{n-1/2} = \frac{1}{2}\{f_{n-1}(x - v_{x,final}, y - v_{y,final}) + f_n(x + v_{x,final}, y + v_{y,final})\} \quad (7)$$

where $(v_{x,final},\ v_{y,final})$ is the final MV, and $f_{n-1/2}$ is the intermediate frame.

III. The Proposed Algorithm

The flow chart of the proposed AEBME is shown in Figure 5. The proposed verification process is comprised of two components: block type matching and frame motion activity check.

In the previous work, we applied scene change detection algorithm and omitted motion vector smoothing algorithm [6]. But scene change detection algorithm shows better results only in specific case. So we omitted scene change detection algorithm in the proposed algorithm to make better AEBME algorithm.

In the previous work, we used CIF test sequences on experiment. But CIF size is not suited for the current imaging technology. So in the proposed algorithm, we used various test sequences used in HEVC test model.

A. Edge Detection

Sobel mask is used to calculate edge information [7]. The operator uses two 3x3 kernels to calculate approximations of the derivatives. The mask is slid over an area of the image. The edge magnitude $M(x,y)$ is calculated by (8).

$$M(x,y) = |g_x| + |g_y| \quad (8)$$

B. Block Type Matching

Non-true MV indicates different objects in the previous and the current frame. We need to check if the objects from MV are same. The edge information, which was calculated using sobel mask, is used to check block type of each object. First, we calculate the mean of edge values of each block by (9). If the edge value of pixel in each block is larger than the mean, the pixel is classified into the edge pixel. Otherwise, the pixel is classified into the flat pixel. If the percentage of edge pixels in the block is larger than T_1, the block is classified into the edge block. Otherwise, the block is classified into the flat block.

$$edge_{mean} = \frac{1}{W \times H}\sum_{x \in W}\sum_{y \in H} M(x,y) \quad (9)$$

After BME, if the block types of the reference blocks of the intermediate block are different, EBME is performed. This block type matching process can improve an accuracy of indicating same objects in the previous and current frame.

C. Frame Motion Activity

Static images have the zero MVs for most blocks. On the other hand, dynamic images have large MVs except background. If OBMC is applied in a static region, visual quality degradation is inevitable. So OBMC should be applied depending on the characteristic of frames after checking frame motion activity. First, we calculate the average MV of all MVs in a frame using (10) and (11). If the average of MVs is larger than T_2, the frame is classified into a dynamic frame. In the opposite case, the frame is classified into a static frame. And then, OBMC is applied only in dynamic frames.

$$v_i = |v_{x,i}| + |v_{y,i}| \quad (10)$$

$$v_{avg} = \frac{1}{(W/BLK) \times (H/BLK)}\sum_{i \in frame} v_i \quad (11)$$

where v_i is the magnitude of *i-th* block's MV, $(W/BLK)x(H/BLK)$ is the number of blocks in a frame, v_{avg} is an average MV of whole MVs in a frame.

 ISBN: 978-1-61208-444-2

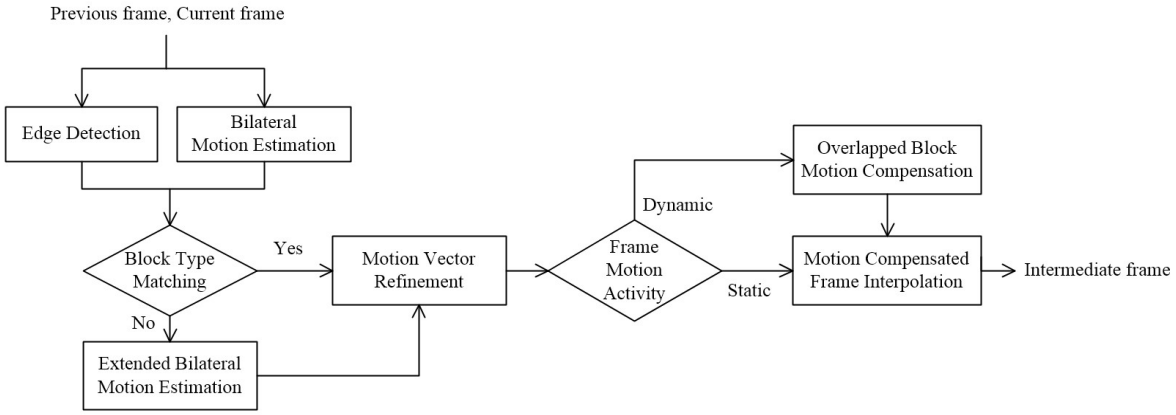


Figure 5. Flowchart of the proposed AEBME algorithm.

IV. EXPERIMENTAL RESULTS

The experiments are conducted using 25 odd frames of test sequences as input and 24 even frames are interpolated as a result. And original even frames are used as reference to calculate the peak signal to noise ratio (PSNR). The performance of the proposed AEBME algorithm has been evaluated through the objective evaluations. PSNR values of the intermediate frames are compared with EBME algorithm. In addition we have calculated the average number of the conducted EBME to show the result of computational complexity reduction.

For experiments, we set the original block size to 32x32 pixels and the search range to -16 +16. After BME, EBME and MVS, we set the block size to 16x16, OBMC filter size N to 32. The threshold T_1 in block type matching process is set to 0.6 and T_2 in checking frame motion activity process is set to 0.5. We used 11 test sequences which are the test sequences for HEVC.

PSNR is used as the metric for objective performance evaluation. The average PSNR values and computation times for the results are presented in Table I. The PSNR difference and computation time gain are also presented. The proposed AEBME algorithm has higher PSNR and comsumes less time than the anchor algorithm. These results are caused by skipping EBME process and OBMC process in static sequences.

TABLE I. PSNRS AND COMPUTATION TIMES OF TEST SEQUENCES.

Class	Sequence	EBME		EBME+MVS+OBMC		AEBME	
		PSNR(dB)	Time(s)	PSNR(dB)	Time(s)	PSNR(dB)	Time(s)
A	Traffic	44.03	120.11	44.13	122.43	45.21	67.14
	PeopleOnStreet	39.58	120	39.59	122.54	40.71	66.59
	Average	**41.81**	**120.06**	**41.86**	**122.49**	**42.96**	**66.87**
	Gain	**0**	**1**	**+0.05**	**0.98**	**+1.15**	**1.8**
B	Kimono1	41.25	55.57	41.32	57.08	41.69	32.52
	ParkScene	41.77	68.23	41.86	69.41	42.29	38.88
	Cactus	38.12	70.15	38.15	71.02	38.46	40.28
	BQTerrace	37.05	72.63	37.08	73.83	37.03	38.43
	BasketballDrive	36.8	59.47	36.87	61.98	37.25	33.49
	Average	**39**	**65.21**	**39.06**	**66.66**	**39.34**	**36.72**
	Gain	**0**	**1**	**+0.06**	**0.98**	**+0.34**	**1.78**
C	RaceHorses	34.12	11.25	34.31	11.65	34.56	6.52
	BQMall	41.06	12.86	41.18	13.05	41.62	7.47
	PartyScene	42.82	14.14	42.79	14.33	43.04	8.02
	BasketballDrill	40.2	13.2	40.17	13.48	40.9	7.52
	Average	**39.55**	**12.86**	**39.13**	**13.13**	**40.03**	**7.38**
	Gain	**0**	**1**	**-0.42**	**0.98**	**+0.48**	**1.74**

Table II shows the number of block type mismatch blocks where EBME algorithm is conducted.

TABLE II. THE AVERAGE NUMBER OF EBME PROCESS PERFORMED.

Class		EBME	EBME+MVS+OBMC	AEBME
A	Average	3871	3871	68.48
	Gain	1	1	56.53
B	Average	1888	1888	74.58
	Gain	1	1	25.32
C	Average	350	350	10.27
	Gain	1	1	34.08

V. CONCLUSIONS

This paper proposed AEBME algorithm, FRUC scheme that considers block type and frame motion activity. The novel block type matching algorithm is proposed to reduce additional BME process. We calculate the edge information using sobel mask, and the calculated edge information is used to decide whether to use EBME. MVS is adopted to detect and eliminate outliers in a MVF. OBMC is selectively applied by utilizing frame motion activity check. Finally, the missing frame are restored by adopting MCFI.

Experimental results show that the proposed algorithm has outstanding performance and fast computation comparing with the anchor algorithm.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1005) supervised by the IITP(Institute for Information&communications Technology Promotion).

REFERENCES

[1] S. H. Chan, T. X. Wu, and T. Q. Nguyen, “Comparison of wo frame conversion schemes for reducing LCD motion blurs,” IEEE Signal Processing Letters, vol. 17, pp. 782-786, 2010.

[2] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, “Motion-compensated frame rate up-conversion Part II: new algorithms for frame interpolation,” IEEE Transactions on Broadcasting, vol. 56, pp. 142-149, 2007.

 ISBN: 978-1-61208-444-2

[3] B. D. Choi, J. W. Han, C. S. Kim, and S. J. Ko, "Motion compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, pp. 407-416, 2007.

[4] S. J. Kang, K. R. Cho, and Y. H. Kim, "Motion compensated frame rate up-conversion using extended bilateral motion estimation," IEEE Transactions on Consumer Electronics, vol. 53, pp. 1759-1767, 2007.

[5] M. T. Orchard and C. J. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach," IEEE Transactions on Image Processing, vol. 9, pp. 1509-1521, 2007.

[6] D. Park and J. Jeong, "Motion Compensated Frame Rate Up-Conversion Using Modified Adaptive Extended Bilateral Motion Estimation," Journal of Automation and Control Engineering, vol. 2, no. 4, pp. 371-375, Dec. 2014.

[7] R. C. Gonzalez and R. E. Woods, "Digital image processing, 3rd edition," Prentice Hall, New Jersey, 2010.

 ISBN: 978-1-61208-444-2

Design of HACCP Communication Protocol

Sungyong Hyun, Seongwook Yoon, Kyung-Ae Cha, Won-Kee Hong
Dept. of Information and Communication Eng.
Daegu University
Gyeongsan, Republic of Korea
e-mail: Gustjddyd90@naver.com, ysw1859@gmail.com, chaka@daegu.ac.kr, wkhong@daegu.ac.kr

***Abstract*—Hazard Analysis Critical Control Point (HACCP) is a systematic preventive regulation to prevent several hazards in food manufacture and treatment processes. We proposed an intelligent HACCP system to put several check points into digital data in order to improve management of food sanitation. In this paper, application level network protocol is introduced for the server to control devices in the working field and for a lightweight device to transfer its check point results to a server.**

Keywords-HACCP; intelligent HACCP system; application level network protocol; food sanitation.

I. INTRODUCTION

The intelligent Hazard Analysis Critical Control Point (i-HACCP) system is a wirelessly connected client-server system for food sanitation management to minimize food safety accident and track the cause of outbreak [1]. Basically, it provides two kinds of lightweight computing devices, which are locked-up type and portable type, to gather several hazards occurred in food processing course. It guides nutritionists in workplace to check Critical Control Point (CCP) every day and record its results online in their tracks. Consequently, the computer-assisted management, such as record, collection and documentation of data inspecting CCPs would make the HACCP help decreasing the food safety accidents effectively.

In this paper, an application level communication protocol is proposed to support efficient information exchange between devices and server in the i-HACCP system. Check items at each CCP step, which nutritionists should inspect and transfer to the server are defined. Encode tables for menus and gradients are maintained between server and devices to minimize network traffic. Some entries in the encode table could be changed by server when a new menu or gradient is introduced. Table management protocol is also designed to inform devices to keep tables updated. The contribution of this paper is that it is the first approach to apply information technology to HACCP in order to enhance its effect on the food sanitation.

Section 2 introduces the i-HACCP System. The HACCP protocol is described in Section 3 and conclusion is made in Section 4.

II. I-HACCP SYSTEM

The i-HACCP system targets a school cafeteria where spaces for pretreatment, cook and food distribution are separated. The overall organization of the i-HACCP system is shown in Figure 1. There are two kinds of terminal devices deployed in the work place. One is a lockup device, which is anchored to a specific place, to get inputs of the examination results for CCP4, CP5, CCP6 and CCP7 by workers on the spot. It can be powered by a permanent and stable electric supplier or operated by a battery. The other one is a portable device with various sensors, such as thermal sensor, chlorine level sensor, iodine level sensor. It is used for recording the examination results for CCP4, CP5, CCP6, and CCP7. This system allows the inspection results to be gathered promptly. It results in increasing the reliability of the HACCP archives, since it minimizes the time interval between the inspection and the record.

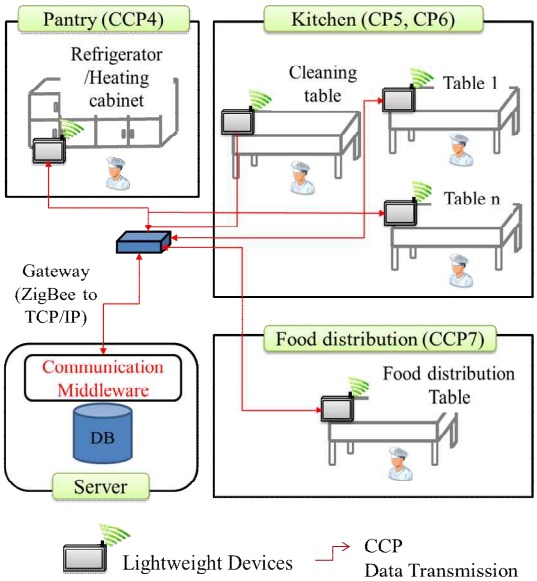


Figure 1. Organization of i-HACCP System.

 ISBN: 978-1-61208-444-2

The terminal device is constructed with a light weight processor, which requires low power and low to medium level processing power. In general, it will be 8 to 32 bit System On Chip (SOC) type microprocessors like ARM's Cortex-M3 or Atmel's ATmega series. The lockup device will use touch panel as an input device, while traditional input buttons will be carried with the portable device. They have 4.3" and 2.5" Liquid Crystal Display (LCD) panel respectively. The devices will use ZigBee Radio Frequency (RF) modem for low power and low data rate wireless communication.

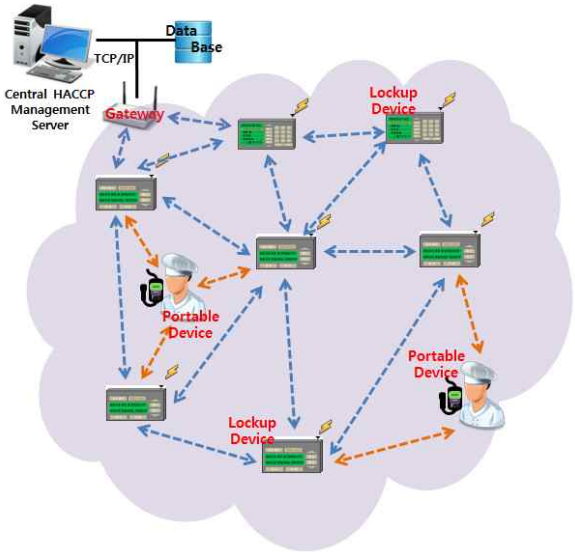


Figure 2. The Network Architecture for i-HACCP system.

Figure 2 shows the network architecture for i-HACCP system. The lockup device plays a role not only as an input terminal but also as a router to transmit data to the neighboring terminal. Basically, the network for i-HACCP system supports a multi-hop network routing protocol to find a path from a terminal device to the HACCP management server. It can be done statically with predefined routing table or dynamically with searching for a proper transmitter at every transfer. Ad-Hoc On-demand Distance Vector (AODV) protocol [12] is usually used for dynamic ad-hoc routing. Basically, the portable device selects one among neighboring lockup devices within its communication range for a transmitter based on signal strength.

III. HACCP PROTOCOL

Based on analysis of HACCP regulation, messages communicated between server and devices are classified into three types, such as request (REQ), response (RSP) and acknowledge (ACK) messages. REQ message are used when a device needs to ask server to register itself or send information, such as current time or today's menu. RSP message are sent only by server when a device sends REQ for current time or today's menu. ACK message indicates that receiver performs its duty to the corresponding REQ or RSP completely. Figure 3 shows flow of messages between client and server for various requests.

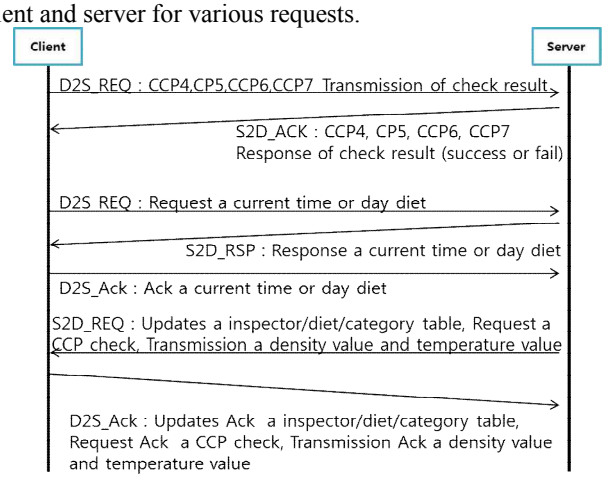


Figure 3. HACCP Communication Protocol.

TABLE I. MESSAGE FORMAT

Field	Description
STX	Start of message (0xAB1)
MSG Type	Message type (REQ, RSP, ACK)
IEEE Addr	IEEE address of device registered in Server
CCP/CP	Device's CCP/CP step and type
Payload Length	Length of payload transmitted
Payload	Data transmitted between client and server
SUM	Byte basis sum from IEEE field to Payload field
ETX	End of message (0xCD34)

The message format for the three types of messages is designed as shown in Table I. MSG type field contain message profile that identifies its meaning, as well as message type. Message types are subdivided into five types, such as D2S_REQ, S2D_REQ, D2S_ACK, S2D_RSP, S2D_ACK according to the transfer direction, i.e., D2S_REQ means a REQ message from device to server.

IV. CONCLUSION

The i-HACCP system collects data written in by nutritionists on the spot in a computerized way and builds HACCP archives automatically to enhance effectiveness of HACCP for a school cafeteria. The i-HACCP system is a multi-hop network architecture that connects several lockup devices and portable devices posted at a suitable place with a wireless modem. In this paper, we proposed a network protocol for i-HACCP system.

For future work, we will focus on the evaluation to measure the impacts of the encode tables, which are shared

 ISBN: 978-1-61208-444-2

between server and client devices and used to reduce the load of network traffic.

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Education (MOE) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation (No.2013028405).

REFERENCES

[1] W. -K. Hong, D. Lee, S. Yoon, and J. T. Ryu, "Intelligent HACCP System for School Foddservices," 2014 IIISC, Jan. 2014, pp. 164–166.

[2] W. H. Sperber and R. F. Stier, "Happy 50th Birthday to HACCP: Retrospective and Prospective," FoodSafety mgazine, Jan. 2010, pp. 44-46.

[3] FAO/WHO, "FAO/WHO guidance to governments on the application of HACCP in small and/or less-developed food businesses," Retrieved 14, Oct. 2007.

[4] D. McSwane and R. Linton, "Issue and concerns in HACCP development and implementation for retail food operations," Journal of Environmental health, Vol. 62, No.6, Jan./Feb. 2000, pp. 15-18.

[5] S. Y. Lee, Y. S. Jang, and H. J. Choe, "Current Status and Further Prospect on HACCP Implementation in Korea(Specially on Catering)," Food industry and nutrition, Vol. 4, No. 3, 1999, pp.14-26.

[6] J. M. Soriano, H. Rico, J. C. Molto, and J. Manes, "Effect of introduction HACCP on the microbiological quality of some restaurant meals," Food Control, Vol. 13, 2002, pp.253-261.

[7] S. Youn and J. Sneed, "Implementation of HACCP and prerequisite programs in school foodservice," Journal of the American Dietetic Association, Vol. 103, 2002, pp.55-60.

[8] Korean Ministry of Education, "Sanitation Management Guide for School Foodservice," Korean government publications 11-1340000-000185-14, 2010.

 ISBN: 978-1-61208-444-2

Block-based Error Compensation Method for Fast Thumbnail Generation in H.264/AVC Bitstreams

Kyung-Jun Lee and Je-Chang Jeong
Department of Electronics and Computer Engineering
Hanyang University
Seoul, Republic of Korea
e-mail: kjlee8812@gmail.com, jjeong@hanyang.ac.kr

***Abstract*—The thumbnail pictures can be generated from the frequency data, which is called H.264 advanced video coding (H.264/AVC) bitstreams, directly without inverse transform process. However, it makes some error caused by rounding of the floating point operation and it will be propagated to neighbor blocks. In this paper, we propose an error compensation method for fast thumbnail generation considering error propagation. It determines the block-based compensation value using the error distribution of intra prediction mode and gives the weighting factor to cover the error propagation.**

Keywords-thumbnail; H.264/AVC; error compensation.

I. INTRODUCTION

H.264/AVC is widely used for digital video processing including both of high and low bit rate applications, such as high definition television (HDTV), internet TV and digital multimedia broadcasting (DMB) services. Moreover, as the development of the personal devices, people can take video contents easily, so both of online and offline storage overflow with a lot of multimedia. In these reasons, some of generating reduced-size images which called thumbnail are needed for file searching and restoring operation. Because thumbnail can give intuitive information of video data, so it can be used for video searching, browsing, and displaying.

The thumbnail extraction from the frequency domain directly is normally used for fast generation. Because DC coefficient in the frequency domain block is considered as the representative value of the block. Therefore, the collection of the DC coefficients of entire images become a thumbnail. Yeo and Liu [1] proposed a method to make DC image consists of DC coefficients of a MPEG-1 frame, and it called DC sequence. Likewise, most thumbnail extracting method from the video streams of the MPEG-1/2 can make various sizes thumbnail images with reduced complexity [2]-[3].

H.264/AVC supports the intra prediction process which predicts and reconstructs the blocks of the image in spatial domain [4]. Chen *et al.* described an intra prediction process as matrix multiplication and proposed a frequency domain prediction method [5]. Kim *et al.* and Yu *et al.* proposed a fast thumbnail extraction method which calculate DC coefficient from the frequency domain directly using Chen's method [6]-[7]. Kim *et al.* proposed another fast thumbnail generation method by substituting multiplications to shifting operations [8]. Yoon *et al.* proposed an error compensation method for thumbnail images [9].

This paper proposes an enhanced error compensation method for thumbnail generation based on [9]. We focused on error propagation which is occurred because of rounding of the floating point operation. We collect the error distribution data from the thumbnail images and set the mean and deviation value. And considering the location of the blocks, we set the weighting factors to determine different compensation values.

This paper is organized as follows. In Section 2, related works are introduced. In Section 3, we describe how to determine the compensation value of the thumbnail image extracting from the bitstreams directly. Section 4 shows the experimental results and Section 5 concludes the paper.

II. RELATED WORKS

A. Fast Thumbnail Extraction

Chen described an intra prediction as a matrix multiplication for 9 modes [5]. Figure 1 shows a 4x4 block. It uses the pixel values in four neighboring blocks for intra prediction. The prediction block of current block $\mathbf{y}_p^m$ can be calculated with matrix form.

$$\mathbf{y}_p^m = \left(\sum_{n=1}^{3} \sum_{i=1}^{4} \mathbf{s}_i \mathbf{x}_n \mathbf{c}_{n,i}^m \right) + \sum_{i=1}^{4} \mathbf{c}_{4,i}^m \mathbf{x}_4 \mathbf{s}_i^T \qquad (1)$$

where m, $\mathbf{s}_i$, $\mathbf{x}_n$, $\mathbf{c}_{n,i}^m$ refers to the intra prediction mode, the shift matrix, the neighboring block of current block and the constant matrix, respectively. And prediction for next block, only the boundary pixels of the current block are needed. Therefore, we filtered the current prediction block in both of vertical and horizontal direction in advance. For this filtering process following **v**, **h** matrix will be used [6].

$$\mathbf{v} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{h} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad (2)$$

 ISBN: 978-1-61208-444-2

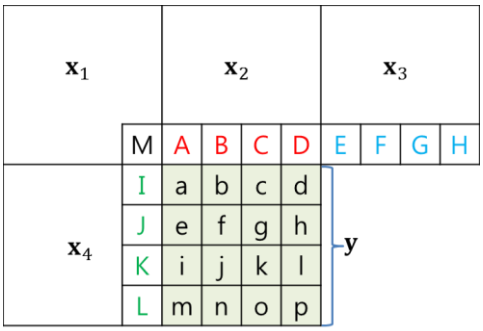


Figure 1. Current block y and the neighboring blocks $\mathbf{x_1}$ to $\mathbf{x_4}$ for intra prediction.

$$V(\mathbf{a}) = \mathbf{va} = \mathbf{a}_v \tag{3}$$

$$H(\mathbf{a}) = \mathbf{ah} = \mathbf{a}_h \tag{4}$$

Equation (3) and (4) are the vertical and horizontal filtering operator, respectively. By using this operator, we can get filtered prediction blocks.

$$\mathbf{y}_{p,v}^{m} = \left(\sum_{n=1}^{3}\sum_{i=1}^{4} V(\mathbf{s}_i\mathbf{x}_{n,v}\mathbf{c}_{n,i}^{m})\right) + \sum_{i=1}^{4} V(\mathbf{c}_{4,i}^{m}\mathbf{x}_{4,\mathrm{h}}\mathbf{s}_i^{T}) \tag{5}$$

$$\mathbf{y}_{p,h}^{m} = \left(\sum_{n=1}^{3}\sum_{i=1}^{4} H(\mathbf{s}_i\mathbf{x}_{n,v}\mathbf{c}_{n,i}^{m})\right) + \sum_{i=1}^{4} H(\mathbf{c}_{4,i}^{m}\mathbf{x}_{4,\mathrm{h}}\mathbf{s}_i^{T}) \tag{6}$$

After calculation, (5) and (6) can be simplified and it is defined as follows:

$$\mathbf{y}_{p,v}^{m} = \mathbf{x}_{1,v}\mathbf{c}_{1,4}^{m} + \mathbf{x}_{2,v}\mathbf{c}_{2,4}^{m} + \mathbf{x}_{3,v}\mathbf{c}_{3,4}^{m} + (\mathbf{p}_{4,v}^{m}\mathbf{x}_{1,h})^{T} \quad \text{where } \mathbf{p}_{4,v}^{m} = \textstyle\sum_{i=1}^{4}(\mathbf{s}_i V(\mathbf{c}_{4,i}^{m})) \tag{7}$$

$$\mathbf{y}_{p,h}^{m} = \mathbf{x}_{1,v}\mathbf{q}_{1,h}^{m} + \mathbf{x}_{2,v}\mathbf{q}_{2,h}^{m} + \mathbf{x}_{3,v}\mathbf{q}_{3,h}^{m} + \mathbf{c}_{4,4}^{m}\mathbf{x}_{4,h} \quad \text{where } \mathbf{q}_{4,h}^{m} = \textstyle\sum_{i=1}^{4}(H(\mathbf{c}_{n,i}^{m})\mathbf{s}_i^{T}) \tag{8}$$

To make DC coefficient, *UNI* operation is needed. It fills the whole components in the matrix with mean value of the block. For this process, following **u** matrix will be used.

$$\mathbf{u} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{9}$$

$$U(\mathbf{a}) = \frac{1}{N \times N}\mathbf{uau} \tag{10}$$

$$\mathbf{y}_{p,uni}^{m} = \sum_{n=1}^{3} \mathbf{x}_{n,v}\mathbf{c'}_{n,i}^{m} + (\mathbf{c'}_{4}^{m})^{T}\mathbf{x}_{4,h} \quad \text{where } \mathbf{c'}_{n}^{m} = \tfrac{1}{16}\textstyle\sum_{i=1}^{4}\mathbf{c}_{n,i}^{m}\mathbf{u} \tag{11}$$

For calculate in the frequency domain, (7), (8), and (11) are transformed by *HT* which is the modified DCT operator in H.264/AVC.

$$\mathbf{Y}_{p,v}^{m} = \mathbf{X}_{1,v}\mathbf{C}_{1,4}^{m} + \mathbf{X}_{2,v}\mathbf{C}_{2,4}^{m} + \mathbf{X}_{3,v}\mathbf{C}_{3,4}^{m} + (\mathbf{P}_{4,v}^{m}\mathbf{X}_{1,h})^{T} \tag{12}$$

$$\mathbf{Y}_{p,v}^{m} = \left(\sum_{n=1}^{3} \mathbf{X}_{n,v}\mathbf{Q}_{n,h}^{m}\right)^{T} + \mathbf{C}_{4,4}^{m}\mathbf{X}_{4,h} \tag{13}$$

$$\mathbf{Y}_{p,uni}^{m} = \sum_{n=1}^{3} \mathbf{X}_{n,v}\mathbf{C'}_{n}^{m} + (\mathbf{C'}_{4}^{m})^{T}\mathbf{X}_{4,h} \tag{14}$$

B. *Error Compensation for Thumbnail Image*

In H.264/AVC, some rounding errors are occurred because of transform, and quantization process. Yoon focus on statistical pattern of truncated errors and set a random variable *r* to compensate them [9]. The compensation value *s* is determined which makes minimum variance of *r*. The matrix form of *s* can be written as follows:

$$\mathbf{s}^{m} = \begin{bmatrix} E[r_{0,0}] & E[r_{0,1}] & E[r_{0,2}] & E[r_{0,3}] \\ E[r_{1,0}] & E[r_{1,1}] & E[r_{1,2}] & E[r_{1,3}] \\ E[r_{2,0}] & E[r_{2,1}] & E[r_{2,2}] & E[r_{2,3}] \\ E[r_{3,0}] & E[r_{3,1}] & E[r_{3,2}] & E[r_{3,3}] \end{bmatrix} \tag{15}$$

In the same way, the representative value which is the mean of distributed error can be written in the matrix form as follows:

$$\mathbf{D}^{m} = \begin{bmatrix} E[D_{0,0}] & E[D_{0,1}] & E[D_{0,2}] & E[D_{0,3}] \\ E[D_{1,0}] & E[D_{1,1}] & E[D_{1,2}] & E[D_{1,3}] \\ E[D_{2,0}] & E[D_{2,1}] & E[D_{2,2}] & E[D_{2,3}] \\ E[D_{3,0}] & E[D_{3,1}] & E[D_{3,2}] & E[D_{3,3}] \end{bmatrix} \tag{16}$$

This predicted error value also will be filtered in vertical and horizontal direction and then it compensates the thumbnail image.

III. PROPOSED METHOD

We propose an enhanced error compensation method using error distribution in each intra prediction modes and set the weighting factors by considering the location of the blocks. When the current block is predicted, it will be filtered for next blocks' prediction. Therefore, the errors also influence next blocks and it will cumulate. For this reason, we must set different compensation value. Each intra prediction modes have a characteristic error distribution, so we use this information to compensate the difference. We calculate the average and the deviation as a representative value of the block in each intra prediction mode.

$$V_{ij} = \mu^{m} + \omega_{k}^{m}\sigma^{m}, if\ i \in b_k \tag{17}$$

 ISBN: 978-1-61208-444-2

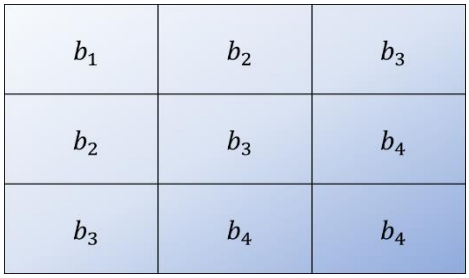


Figure 2. Partitioned image for different weighting factors.

Equation (17) determine the compensation values. The $\boldsymbol{\mu^m}$, $\boldsymbol{\sigma^m}$ are the average value and the deviation of mode *m*, respectively and we can get this factor experimentally. In order to consider the error distribution, we divide an image into blocks as shown in Figure 2. Finally we calculate the compensation values V_{ij} by using weighting factors ω_k^m.

IV. EXPERIMENTAL RESULTS

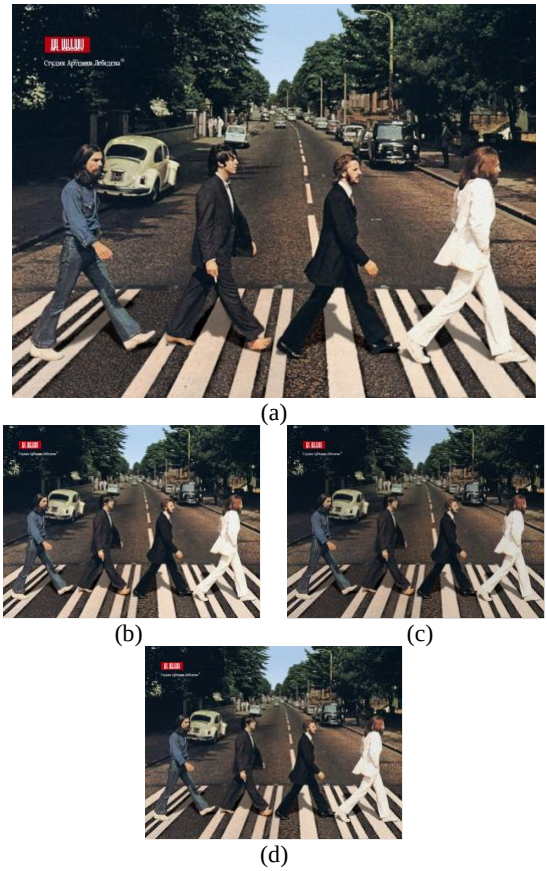
The proposed method was evaluated by the following conditions. We randomly choose the test images and the seven unofficial JPEG images (*Album1*, *Album2*, *Hirmer*, *Soccer*, *Beatles*, *TVshow*, and *Flowershop*) were used. The size of the images is shown in Table I. We generate three thumbnail images. Frequency domain (FD) makes the thumbnail by extracting DC coefficients and fast generation method from frequency domain (FFD) makes it by using [6]. Block-based FD (BFD) is the result of proposed method. The weighting factors related to each prediction mode and location of block are set experimentally. We were doing the experiment on the MATLAB program for simple comparison.

TABLE I. THE SIZE OF THE TEST IMAGES.

Image	Size
Album1	600×600
Album2	640×640
Hirmer	720×540
Soccer	900×656
Beatles	1024×768
TVshow	1280×720
Flowershop	4272×2848

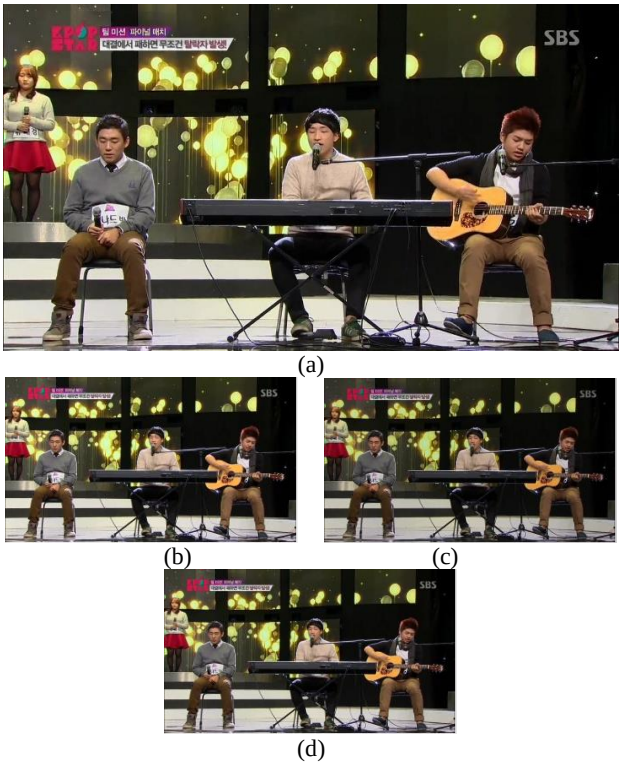
Figure 3 and 4 show the comparison of subjective quality of the thumbnail image. It shows almost same at first sight but it has certain difference in detail of the objects. FFD method shows a little bit dark compare to FD result because of errors. Especially, due to error propagation, it become darker when it goes to bottom-right part of the image than upper-left part. BFD method compensates the degraded part effectively. So, it looks more similar to FD result than FFD method.

Table II shows the average peak signal-to-noise ratio (PSNR) of each method. We compute the PSNR of the FFD and BFD with reference to the thumbnail image generated by FD. As shown in this table, proposed method can make higher PSNR for all test images. Especially, *Flowershop* image shows the outstanding improvement.



(a) (b) (c) (d)

Figure 3. The *Beatles* image (a)Original image, (b)FD, (c)FFD, (d)BFD.



(a) (b) (c) (d)

Figure 4. The *TVshow* image (a)Original image, (b)FD, (c)FFD, (d)BFD.

TABLE II. COMPARISON RESULT OF THE PSNR (dB).

Image	FFD [6]	BFD	Δ
Album1	34.24	36.01	+1.77
Album2	33.30	34.98	+1.68
Hirmer	33.69	35.08	+1.39
Soccer	33.74	35.10	+1.36
Beatles	32.45	34.45	+2.00
TVshow	34.45	36.27	+1.82
Flowershop	24.84	28.75	+3.91

V. CONCLUSIONS

We have presented the block-based error compensation method for thumbnail extraction. It uses the error distribution data of each intra prediction modes. It determines the compensation value with the mean and the deviation value of the errors. Also, we divide the image into some blocks and adaptively set the weighting factors by considering the location of the pixels. Thus, it have lower value when it goes to upper-left and higher value when it goes to bottom-right. By using the proposed method, we can successfully compensate the error of thumbnail which is extracted by fast extraction algorithm. It achieves better result both of subjective and objective comparison.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1005) supervised by the IITP(Institute for Information&communications Technology Promotion)

REFERENCES

[1] B. Yeo and B. Liu, “Rapid Scene Analysis on Compressed Video,” IEEE Trans. Circuits and Syst. Video Technol., vol. 5, no. 6, pp. 533-540, Dec. 1995.

[2] J. H. Song and B. L. Yeo, “Fast Extraction of Spatially Reduced Image Sequences from MPEG-2 Compressed Video,” IEEE Trans. Circuits Syst. Video Technol., vol. 9, no. 7, pp. 1100-1114, Oct. 1999.

[3] S. J. Suh, S. S. Chun, M. H. Lee, and S. H. Sull, “Efficient Image Down-Conversion for Mixed Field/Frame-mode Macroblocks,” IEEE Electron. Lett., Vol. 39(6), pp. 514-515, Mar. 2003.

[4] ISO/IEC JTC1/SC29/WG11 (MPEG), “Coding of audio-visual objects-Part 10: Advanced Video Coding,” International Standard 14496-10, ISO/IEC, 2004.

[5] C. Chen, P. H. Wu, and H. Chen, “Transform-Domain Intra Prediction for H.264,” IEEE ISCAS, pp. 1497-1500, May. 2005.

[6] E. S. Kim, T. W. Um, and S. J. Oh, “A Fast Thumbnail Extraction Method in H.264/AVC Video Streams,” IEEE Trans. Consumer Electronics, vol. 55, no. 3, pp. 1424-1430, Aug. 2009.

[7] S. J. Yu, M. K. Yoon, E. S. Kim, C. B. Sohn, D. G. Sim, and S. J. Oh, “An Efficient Thumbnail Extraction Method in H.264/AVC Bitstreams,” THE KOREAN SOCIETY OF BROADCAST ENGINEERS, vol. 13, no. 2, pp. 222-235, Mar. 2008.

[8] M. H. Kim, H. J. Lee, and S. H. Sull, “Fast Thumbnail Generation in Integer DCT Domain for H.264/AVC,” IEEE Trans. Consumer Electronics, vol. 57, no. 2, pp. 589-596, May. 2011.

[9] M. K. Yoon, Y. S. Lee, C. B. Sohn, H. C. Park, C. B. Ahn, and S. J. Oh, “An Efficient Error Compensation Method for Thumbnail Extraction in H.264/AVC Bitstrems,” THE KOREAN SOCIETY OF BROADCAST ENGINEERS, vol. 13, no. 5, pp. 622-635, Sep.

 ISBN: 978-1-61208-444-2

Using High Performance Parallel Data Warehouse (HPDW) Big Data Analytical Platform for Big Data Analysis

Boon Keong Seah
MIMOS
Technology Park Malaysia
KL, Malaysia
seahbk2006@yahoo.com

***Abstract*—Data warehouse has been traditionally implemented in Relational Database Management System (RDBMS) from operational data store up until the data marts and OLAP (online analytical processing) cubes for data analysis. However, the process of analyzing big data based on RDBMS is a time consuming process. In addition, with the advent of IoT, social media and other means of big data incorporation, the challenge pose to process the enormous streaming data with the need to obtain the data analysis at hand with near real time requires a need of new platform to address this. Big data incorporation for data analysis is important as it will enlarge the scope of analysis such as weather, devices information, real-time data for data correlation with existing historical data. Presently, RDBMS is not developed for handling large data set and also with ability to perform join queries between historical and streaming data for more data insight. In this paper, we introduce HPDW appliance which is a new big data platform encompassing from stream and batch data process and data query through JDBC, ODBC and integrated multi-data source BI dashboarding and data scientist tool. As it is an appliance, the nodes and all respective components required are pre-configured, hence data scientist or BI analysis will focus on using the big data for analysis and not on the setup of the big data infrastructure which will be time consuming. HPDW appliance is developed based on Massive Parallel Process (MPP) to achieve the in-memory speed it requires which uses Hadoop Distributed File System (HDFS) as the storage layer and high network speed Infiniband for node connectivity. In this paper, we describe experimental results related with the performance of its query processing. We compare the performance results on a physical cluster between RDBMS against HPDW system by varying the size of the data warehouse for fact table queries ranging from 7GB to 23GB data size. Our experiment results show that HPDW system can process the same SQL query with respect to RDBMS much faster, up to 11-200 times faster. In addition, we also show the data analysis results and data mining that can be performed on HPDW.**

Keywords—big data; hadoop; hive; parallel process; infiniband; RDBMS; MPP; streaming data; data warehouse; cube; Extract Transform Load (ETL); OLAP; business intelligence; data marts

I. INTRODUCTION

Data warehouse has been used actively in various industries for data analysis and decision making by the management. Traditionally data warehouse has been developed using combination of ETL tool, BI analysis and presentation layer as well as RDBMS for the data storage and processing. Data warehouse requires a number of predefined stages [1][2] and for handling large data set or Big Data such as Petabyte data warehouse, it is not common to hear RDBMS being used for this task [7][8]. Big Data is perceived as a new enablement for competitive advantage [9]. Big Data has the characteristics of high Volume, high Variety and high Velocity with information to be delivered very quickly [10]. By analyzing the relationship between the data in the combination of Big Data, we are able to gain competitive advantage [11][12]. Hence, by bringing Big Data to the data warehouse we are enriching the data further such as combining existing data with new data set of Big Data such as weather and social media to provide an even better analysis.

However, bringing a combination of Big Data to data warehouse is a challenge as existing RDBMS technology is not built for handling large data set [7][8] and in addition is the ability to perform joins queries between historical and streaming data.

In this paper, we introduce HPDW which is a new Massive Parallel Process (MPP) where it uses Hadoop Distributed File System (HDFS) as the storage layer and its own parallel query execution engine in combination with high network speed Infiniband integration into the clusters. In this paper, we first, set up a star schema health data warehouse on the HPDW for performing online analysis with HPDW Data Analysis which is a BI tool. In addition, we also use commercial database client tool with HPDW JDBC to access the HPDW system for query performance analysis. We compare the performance results on a physical cluster between RDBMS against HPDW system by varying the size of the data warehouse for fact table queries ranging from 7GB to 23GB data size. Our experiment results show that the HPDW system can process the same SQL query with respect to RDBMS much faster, up to 11-200 times faster. We also introduce HPDW ability to perform data mining on streamed data stored in HPDW. In addition, we also show ability of HPDW to perform unify query between batch and stream data for further analysis required.

This paper is organized as follows. Section II describes the background of this work. Section III describes the HPDW system overview. Section IV describes the approach of our data warehouse system and schema design. We also show our experimental evaluation result about HPDW vs RDBMS on SQL query performance in Section V. In this section, we also show the output using charts from HPDW Data Analysis and also using of python script for retrieving streaming data for

 ISBN: 978-1-61208-444-2

data mining. A brief conclusion and future works about this paper are made in Section VI.

II. RELATED WORKS

Traditionally data warehouse has been implemented using RDBMS. In order to implement high performance in RDBMS, parallel query processing has been implemented by RDBMS to speed up query process. Several RDBMSs [19][28][29] support parallel queries, where data can be partitioned across several nodes and accessed simultaneously.

However, handling of large data set or Big Data such as Petabyte data warehouse, it is not common to hear RDBMS being used for having the ability to store in large data set and perform data analysis efficiently [22][23]. In order to facilitate data analysis on these large dataset, there are two possibilities of addressing these, e.g. using massive parallel processing (MPP) system such as Teradata [18][28], Vertica [29] and Greenplum [19] or massive scale data processing platform such as MapReduce [20], Hadoop [21], and Dryad [22]. Each system is equipped with a high-level language (e.g., SQL [23], Hive [16][17], Pig Latin [25], or Sawzall [26]). Programs written in these languages are compiled into a graph of operators called a plan. The plan is then executed as a parallel program distributed across a cluster.

In the massive scale data processing, MapReduce, Hadoop, Hive and Pig are commonly being used. MapReduce [20] is a programming model for processing massive-scale data sets in large shared-nothing clusters. Users specify a map function that generates a set of key-value pairs, and a reduce function that merges or aggregates all values associated with the same key. A combination of map function and reduce function is called a job. In SQL(Structured Query Language), a MapReduce job can be expressed as an aggregation query. Hadoop [21] is an open-source implementation of MapReduce written in Java. Hadoop consist of HDFS which provides high scalability to store big data and MapReduce which presents an efficient programming model in processing HDFS. However, for data analyst, usual form of analyzing big data is through SQL query rather than having to develop MapReduce program code which is a heavy task. As a result, Facebook developed and published Hive [16][17][27] in order to resolve this problem. Hive processes the query of big data distributed and stored in Hadoop by providing an interface significantly similar to SQL called HiveQL(Hive Query Language). However, since Hive underlying framework runs the MapReduce of Hadoop, it does not have a performance advantage as a relational database. Fig. 1 and Fig. 2 show the overall flow and architecture of MapReduce and Hive/Hadoop respectively.

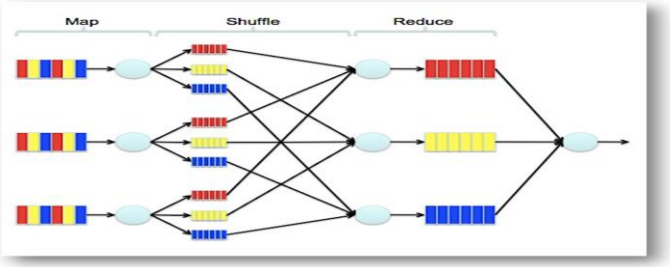


Figure 1. Processing Flow of MapReduce

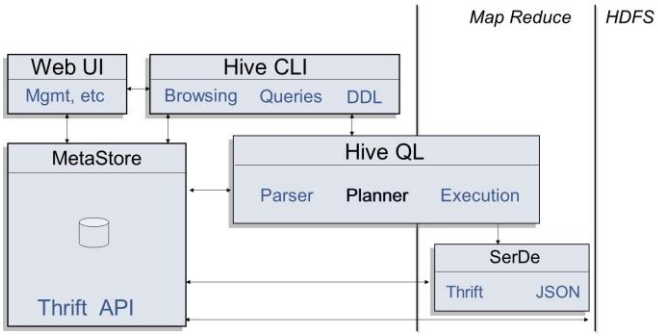


Figure 2. Hive Architecture Overview

In the Massive Parallel data processing in a shared-nothing architecture there have been many research conducted [32][33][34][35][36][37][38]. In addition, there are some high-end commercial MPP products are on the market today [19][28][29][30][31] that is a shared-nothing MPP. One of the well-known commercial MPP is Greenplum database which is a shared-nothing MPP architecture. It is a MPP database [20] infrastructure coupled with computational capabilities to provide faster querying. Some of these MPP commercial databases presently are supporting Hadoop. Nevertheless, they still require the migration from Hadoop into MPP database. Our system differs from this where we uses HDFS as the data store without having to migrate.

We also perform background study on whether the implementation should be performed on virtual or physical cluster [14]. As our implementation stress on performance, we found that for the same number of nodes, physical server perform at least twice the speed of virtual server on the big data set. Hence, our implementation work is performed on physical cluster.

III. HPDW SYSTEM OVERVIEW

HPDW Big Data Analytical Platform consists of 4 major sections: Data Streaming, Data Platform, Data Exploration and Analytics. In Fig. 3, an overview of HPDW Big Data Analytical Platform is shown:

 ISBN: 978-1-61208-444-2

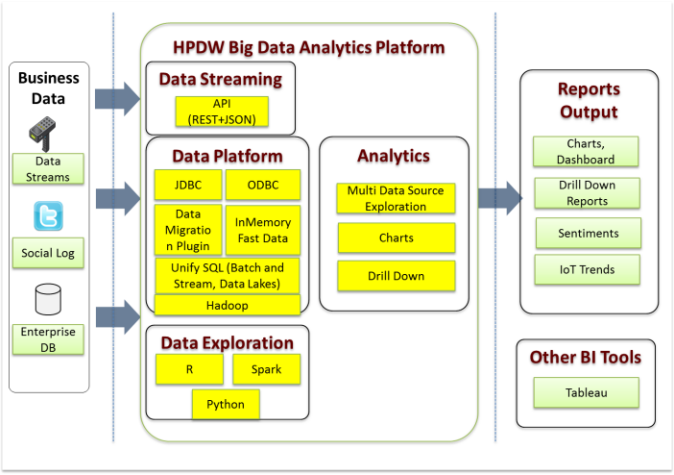


Figure 3. HPDW Data Platform Architecture Overview

HPDW Data Streaming provides a RESTful JSON service to accept continuous data streaming in any JSON format. The service uses Apache Kafka as the data streaming queue to accept high-load of data stream. Streaming Data are stored in HPDW Data Platform section for historical data query.

HPDW Data Platform is based on shared-nothing MPP architecture where each node will use the local disks, memory, etc. HPDW however is based on shared-nothing MPP architecture where each node will use the local disks, memory, etc. Basically HPDW allows for commodity based servers to be connected on a dedicated high speed network with its own parallel query execution engine and memory where these nodes will be known as worker nodes which can perform the reduce steps and then pass on the result back to the master nodes which will aggregate all the nodes results to the requester. All the aggregation processes are done in memory for speed optimization. The parallel query execution engine is responsible for converting SQL into a physical execution plan by performing a query profiling and choosing the query plan based on cost-based optimization. The query execution will then be divided to all the nodes so that it can be executed in parallel. The connection amongst the nodes are provided through HPDW interconnect which uses Infiniband. In the HPDW storage layer, it consists of HDFS which is based on Hadoop cluster with MapReduce engine. HDFS basically stores the data and the metadata of tables loaded from HDFS. Fig. 4, shows the overview of the processing steps of HPDW.

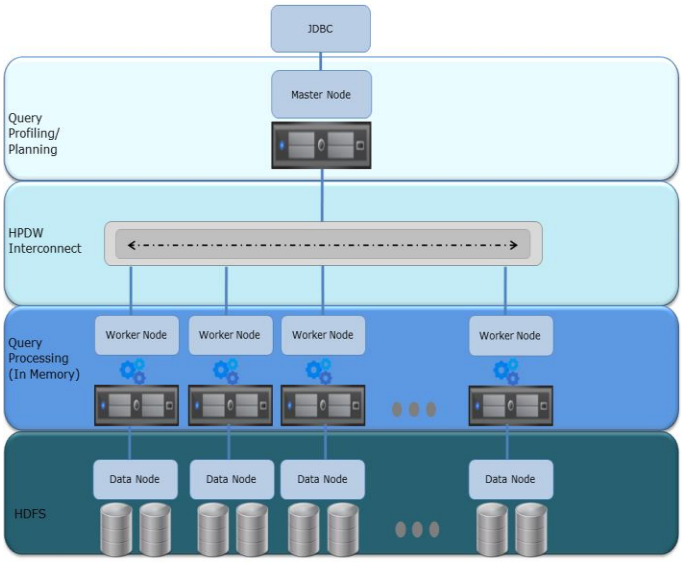


Figure 4. HPDW Data Platform Architecture Overview

HPDW Data Exploration is based on python to provide data scientist the ability to perform data mining on both stream and batch data using R, Spark and Python. The scripts can connect to HPDW Data Platform for data mining of the data stored there such as historical data stream.

HPDW Data Analytics is a web-based multi-data source BI tool to provide analysis with dashboard, charts, GIS to further understand the data. Presently the type of data sources supported includes Twitter, Facebook, HPDW Data Platform, PostgreSQL, Oracle, MySQL, CSV, SOLR, IMPALA, Hive2, MongoDB, RESTful JSON. The charts enable dynamic drill down, dynamic filtering, and dynamic chart changing to enable easier visual analysis of data.

IV. DATA WAREHOUSE IMPLEMENTATION

As we have previously implemented a healthcare data warehouse in PostgreSQL which encompasses from data source ingestion until data marts for dashboards, we have come across issues of performance with PostgreSQL. Hence, performance comparison of data warehouse migrated from RDBMS to HPDW big data is required to validate whether the migration of the data warehouse is worthy to overcome the performance issues we faced during the ETL process in RBMS which is slow. We analyse the query performance between HPDW and RDBMS (PostgreSQL) where we have implemented the health data warehouse onto both HPDW and RDBMS with the same set of data and data model. A star schema is used to model the data warehouse in which facts and dimensions are relate through their respective entity keys to form a join table.

Fig. 5, 6 and 7, shows the star schema model in conceptual, logical and physical data model representation.

 ISBN: 978-1-61208-444-2

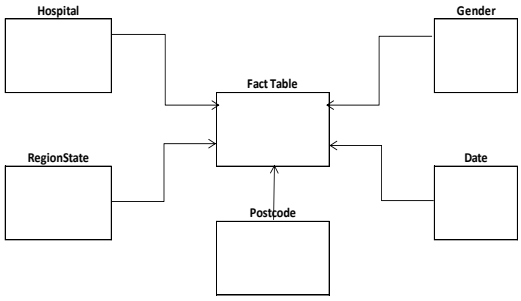


Figure 5. Conceptual Data Model of Data Warehouse implemented with HPDW

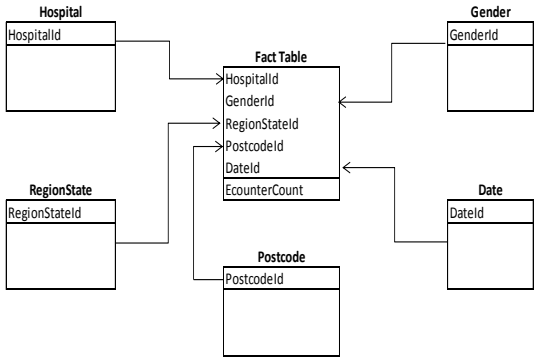


Figure 6. Logical Data Model of Data Warehouse implemented with HPDW

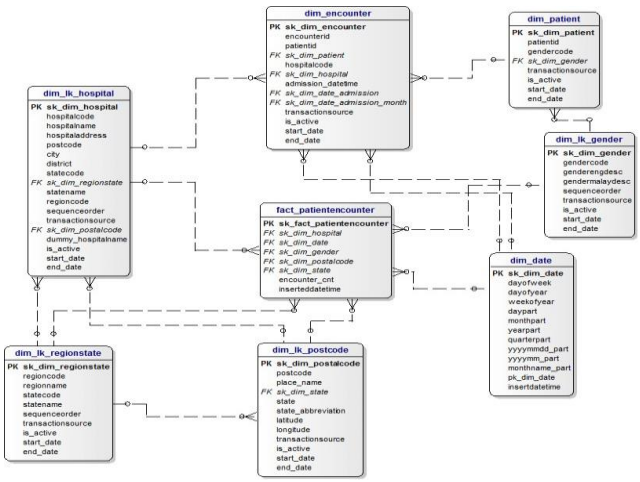


Figure 7. Physical Data Model of Data Warehouse implemented with HPDW

V. EXPERIMENT AND RESULT

In this section we present the performance results on a physical cluster between RDBMS against HPDW system by varying the size of the data warehouse for fact table queries ranging from 7GB to 23GB data size. We compare the results of the two systems to demonstrate the benefit of using HPDW for processing very large data set.

A. *Experiment*

In the experiment, we pre-generated a number of random data for the fact tables ranging from 100M to 300M rows. We have implemented the HDFS using Hadoop 2.4 and distribute the data to 4 physical nodes where one is serving the Master Node and Hadoop Name Node and the others as the Worker Node and Hadoop Data Nodes. Each node has a 128 GB memory in which the details are given in Table I.

TABLE I. HARDWARE SPECIFICATION FOR HPDW AND POSTGRESQL

	HPDW	**PostgreSQL**
Nodes	4 x Physical Nodes	N/A
CPU	Intel Xeon Ten-Core E5-2660v3 2.60Ghz processors – 20 Cores	Intel Xeon Ten-Core E5-2660v3 2.60Ghz processors – 20 Cores
RAM	128 GB	96 GB
Storage	HDD 4 TB (RAID 10)	HDD 4 TB (RAID 5)
OS	Ubuntu (64 bits)	Ubuntu (64 bits)

We compare the performance of a set of SQL queries given in Tab. II, IV and VI which runs on PostgreSQL and our prototype system HPDW (MPP with HDFS). To capture the SQL queries performance, we use Aqua Data Studio [15] as the database client tool for both HPDW and PostgreSQL query analysis. In order for the database tool to access the HPDW system, a HPDW JDBC is provided.

B. *Results*

Fig. 8 shows the summary of the execution time of SQL queries compared between HPDW and PostgreSQL on various data set sizes. The experiment results show that HPDW is more than 11-200 times faster than PostgreSQL. With the speed it achieved, it will enable data analysis at a much faster rate rather than the norm of data warehouse where it needs to go through a number of stages and days to aggregate the data.

 ISBN: 978-1-61208-444-2

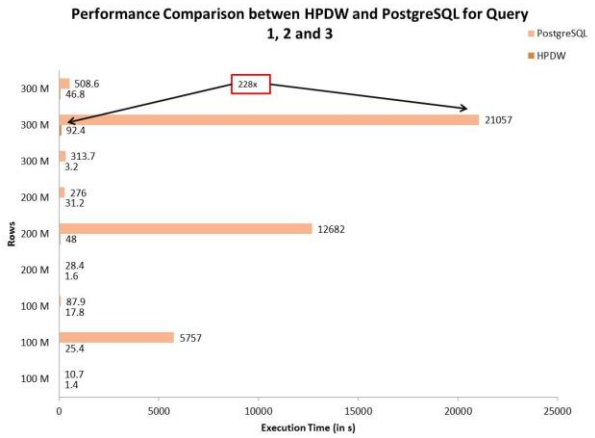


Figure 8. Performance comparison between HPDW and PostgreSQL for Q1, Q2, Q3 for data size between 7-23GB on 100M, 200M and 300M rows respectively.

Following are the list of queries test we have conducted on both HPDW and PostgreSQL for the same set of queries and data size.

TABLE II. QUERY FOR TOTAL NUMBER OF PATIENTS

Number of records	**Query 1 Test Case:** Total number of patients
100 million	select count (*) from **fact_100m**
200 million	select count (*) from **fact_200m**
300 million	select count (*) from **fact_300m**

TABLE III. EXECUTION TIME FOR TOTAL NUMBER OF PATIENTS

Numbers of records (In millions)	Execution Time (In s)										
	HPDW						PostgreSQL				
	1	2	3	4	5	Average	1	2	3	4	5	Average
100	3	1	1	1	1	1.4	101.6	11.1	10.7	10.7	10.7	29
200	3	1	2	1	1	1.6	208.2	130.4	28.3	28.4	28.4	84.7
300	4	3	4	3	2	3.2	432.6	423.6	345.6	315.1	313.7	366.1

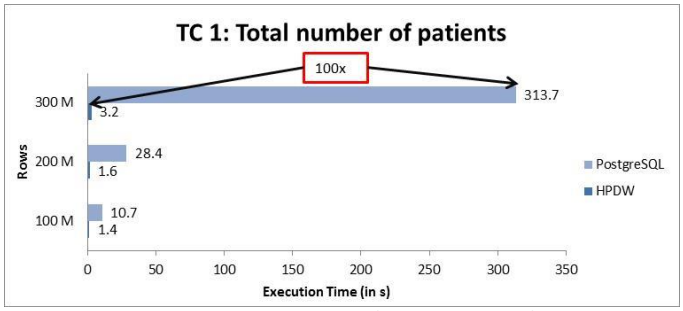


Figure 9. Execution time comparison for total number of patients

For Q1 test case: Total numbers of patients, PostgreSQL takes about 313.7 seconds to execute 300 M rows of records. HPDW just takes 3.2 seconds. It is 100 times faster than PostgreSQL.

TABLE IV: QUERY FOR TOTAL ENCOUNTERS BY MONTH & YEAR, SERVICETYPE, NATIONALITY, AGEGRP, GENDER

No. record	**Query 2 Test Case:** Total Encounters by month & year, servicetype, nationality, agegrp, gender.
100 million	SELECT d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR) AS mth_yr ,st.svctype,n.nationaltype,ag.agegrp,g.gender,SUM(f.enc_cnt) AS enc_cnt FROM **fact_100m** f JOIN dim_lk_servicetype st on st.sk_dim_servicetype=f.sk_dim_servicetype JOIN dim_lk_agegrp ag ON ag.sk_dim_agegrp=f.sk_dim_agegrp JOIN dim_lk_gender g ON g.sk_dim_gender=f.sk_dim_gender JOIN dim_lk_nationality n ON n.sk_dim_nationality=f.sk_dim_nationality JOIN dim_date d ON d.sk_dim_date=f.sk_dim_date where d.yr_pt=2013 GROUP BY d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR) ,st.svctype,n.nationaltype,ag.agegrp,g.gender
200 million	SELECT d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR) AS mth_yr,st.svctype,n.nationaltype,ag.agegrp, g.gender,SUM(f.enc_cnt) AS enc_cnt FROM **fact_200m** f JOIN dim_lk_servicetype st on st.sk_dim_servicetype=f.sk_dim_servicetype JOIN dim_lk_agegrp ag ON ag.sk_dim_agegrp=f.sk_dim_agegrp JOIN dim_lk_gender g ON g.sk_dim_gender=f.sk_dim_gender JOIN dim_lk_nationality n ON n.sk_dim_nationality=f.sk_dim_nationality JOIN dim_date d ON d.sk_dim_date=f.sk_dim_date where d.yr_pt=2013 GROUP BY d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR), st.svctype,n.nationaltype, ag.agegrp,g.gender
300 million	SELECT d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR) AS mth_yr,st.svctype,n.nationaltype,ag.agegrp,g.gender,SUM(f.enc_cnt) AS enc_cnt FROM **fact_300m** f JOIN dim_lk_servicetype st on st.sk_dim_servicetype=f.sk_dim_servicetype JOIN dim_lk_agegrp ag ON ag.sk_dim_agegrp=f.sk_dim_agegrp JOIN dim_lk_gender g ON g.sk_dim_gender=f.sk_dim_gender JOIN dim_lk_nationality n ON n.sk_dim_nationality=f.sk_dim_nationality JOIN dim_date d ON d.sk_dim_date=f.sk_dim_date where d.yr_pt=2013 GROUP BY d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR), st.svctype, n.nationaltype,ag.agegrp,g.gender

TABLE V. EXECUTION TIME FOR TOTAL ENCOUNTERS BY MONTH & YEAR, SERVICETYPE, NATIONALITY, AGEGROUP, GENDER

Numbers of records (in millions)	Execution Time (in s)											
	HPDW						PostgreSQL					
	1	2	3	4	5	Average	1	2	3	4	5	Average
100	26	25	25	26	25	25.4	5757					5757
200	47	46	51	49	47	48	12682					12682
300	92	103	89	89	89	92.4	21057					21057

For Q2 test case: Total Encounters by month & year, servicetype, nationality, agegrp, gender , PostgreSQL

takes about 21057 seconds to execute 300 M rows of records. HPDW just takes 92.4 seconds. It is 228 times faster than PostgreSQL.

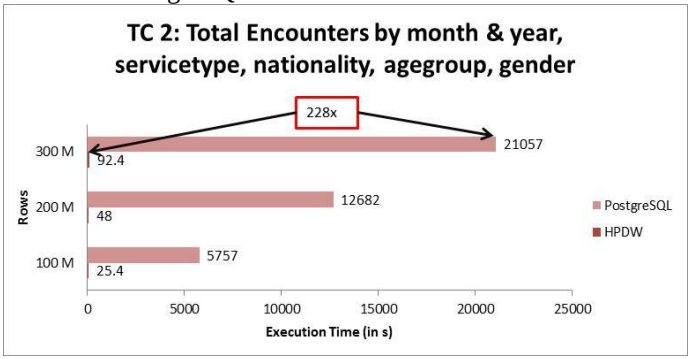


Figure 10. Execution time comparison for Total Encounters by month & year, servicetype, nationality, agegrp, gender

TABLE VI. QUERY FOR TOTAL ENCOUNTERS BY MONTH & YEAR, REFERENCE HOSPITAL, AGEGRP, GENDER

No. record	**Query 3 Test Case:** Total Encounters by month & year, reference hospital, agegrp, gender
100 million	SELECT d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR) AS mth_yr,r.reference, ag.agegrp,g.gender,SUM(f.enc_cnt) AS enc_cnt FROM **fact_100m** f JOIN dim_lk_ref r on r.sk_dim_ref=f.sk_dim_ref JOIN dim_lk_agegrp ag ON ag.sk_dim_agegrp=f.sk_dim_agegrp JOIN dim_lk_gender g ON g.sk_dim_gender=f.sk_dim_gender JOIN dim_date d ON d.sk_dim_date=f.sk_dim_date where d.yr_pt=2013 GROUP BY d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR), r.reference, ag.agegrp,g.gender
200 million	SELECT d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR) AS mth_yr,r.reference, ag.agegrp,g.gender,SUM(f.enc_cnt) AS enc_cnt FROM **fact_200m** f JOIN dim_lk_ref r on r.sk_dim_ref=f.sk_dim_ref JOIN dim_lk_agegrp ag ON ag.sk_dim_agegrp=f.sk_dim_agegrp JOIN dim_lk_gender g ON g.sk_dim_gender=f.sk_dim_gender JOIN dim_date d ON d.sk_dim_date=f.sk_dim_date where d.yr_pt=2013 GROUP BY d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR), r.reference, ag.agegrp,g.gender
300 million	SELECT d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR) AS mth_yr,r.reference, ag.agegrp,g.gender,SUM(f.enc_cnt) AS enc_cnt FROM **fact_300m** f JOIN dim_lk_ref r on r.sk_dim_ref=f.sk_dim_ref JOIN dim_lk_agegrp ag ON ag.sk_dim_agegrp=f.sk_dim_agegrp JOIN dim_lk_gender g ON g.sk_dim_gender=f.sk_dim_gender JOIN dim_date d ON d.sk_dim_date=f.sk_dim_date where d.yr_pt=2013 GROUP BY d.mth_pt||'-'||CAST(d.yr_pt AS VARCHAR), r.reference, ag.agegrp,g.gender

TABLE VII. EXECUTION TIME FOR TOTAL ENCOUNTERS BY MONTH & YEAR, REFERENCE HOSPITAL, AGEGRP, GENDER

Numbers of records (in millions)	Execution Time (in s)											
	HPDW						PostgreSQL					
	1	2	3	4	5	Average	1	2	3	4	5	Average
100	17	17	17	18	20	17.8	125.1	87.7	87.8	87.9	87.9	95.28
200	35	34	32	30	25	31.2	277	285	279	277.7	276.6	279.06
300	49	44	46	47	48	46.8	509.6	507.8	507.9	508.8	508.6	508.5

For Query 3: Total Encounters by month & year, reference hospital, agegrp, gender, PostgreSQL takes about 508.6 seconds to execute 300 M rows of records. HPDW just takes 46.8 seconds. It is 11 times faster than PostgreSQL.

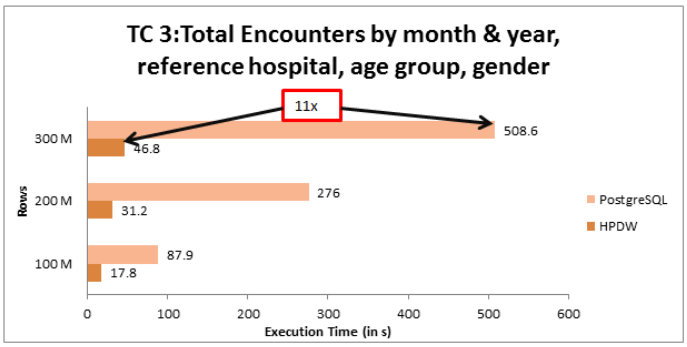
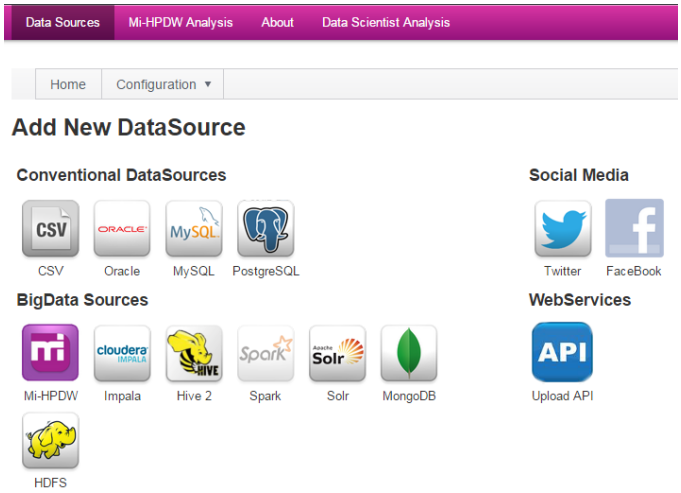
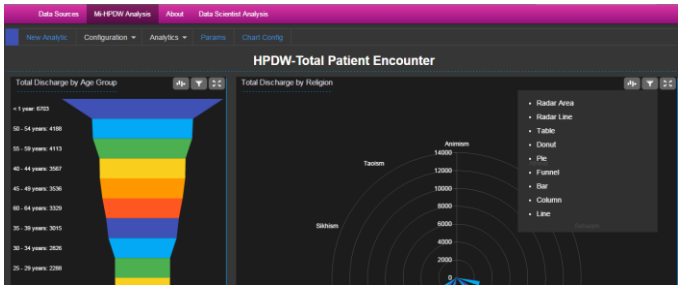


Figure 11. Execution time comparison for Total Encounters by month & year, reference hospital, agegrp, gender

Overall, we can see HPDW outperforms PostgreSQL greatly. In addition, we also perform queries utilizing HPDW Data Analysis which is a multi-data source data analysis tool we have developed. We developed analytical charts against both HPDW and PostgreSQL using Q2 100M records as shown in Fig. 12. With the HPDW Data Analysis tool, we are able to perform data exploration onto HPDW but not with PostgreSQL data source as the query execution in PostgreSQL requires at least 95 minutes for execution. The database connectivity will time out after a period of 5 minutes.





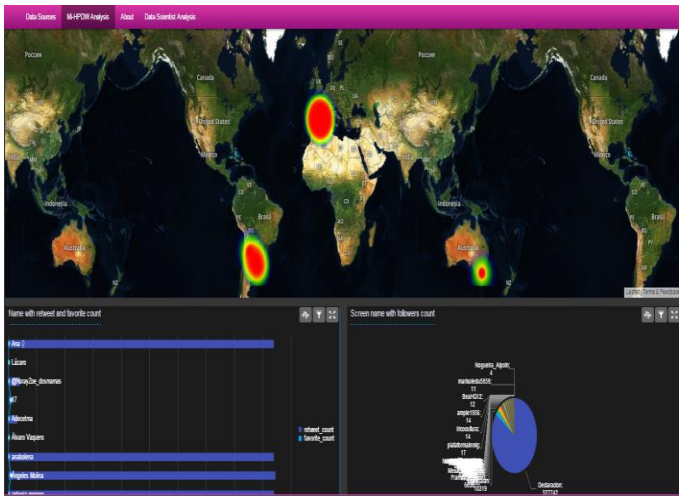


Figure 12. HPDW Data Analysis for multi data source exploration with drill down. Shown here is data analysis on HPDW data source.

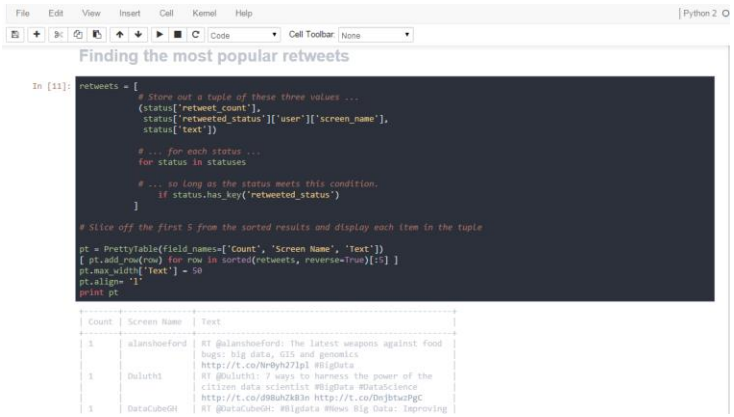


Figure 13. HPDW Data Exploration for data mining of data

VI. CONCLUSION

In this paper, we introduce HPDW Big Data Analytical Platform which is a new big data analytical platform that can provide end-to-end solution for both storing and analyzing of historical and streaming data. It can be used to analyse multi data source of data and unify the data for further data exploration. In order to achieve the speed it requires, HPDW uses InMemory for data process and Infiniband as the high network speed to interconnect all the data nodes. HPDW also incorporates RESTful JSON for easy stream data insertion. Historical data stream can then be analyzed through HPDW data analysis web system or scripts as shown in Fig. 13. We also provide JDBC and ODBC connection for further 3rd party tool integration such as Tableau.

In this paper, we evaluated the performance of a commercial RDBMS (PostgreSQL) and HPDW on health data warehouse for fact table queries ranging from 7GB to 23GB data size. Our tests indicate that overall HPDW outperforms RDBMS (PostgreSQL) for large data sets in the range of 11-200 times. In future, we will further improve HPDW by having more SQL query commands to be supported. This will enable more support for data analysis of the data stored in HPDW. In addition, more types of data sources for the HPDW Data Analysis will be supported such as OData, Excel, Spark and others. Hence, this big data analytical platform is able to provide data scientist and BI analysis a piece of mind as it reduces the effort requires to setup, developed and configure the big data storage, speed and streaming required.

ACKNOWLEDGMENT

This work is funded by MOSTI under HPDW Techfund.

REFERENCES

[1] Boon Keong Seah, "An application of a healthcare data warehouse system," Innovative Computing Technology (INTECH), 2013 Third International Conference on , vol., no., pp.269-273, 29-31 Aug. 2013.

[2] Boon Keong Seah; Selan, N.E., "Design and implementation of data warehouse with data model using survey-based services data,"

 ISBN: 978-1-61208-444-2

Innovative Computing Technology (INTECH), 2014 Fourth International Conference on , vol., no., pp.58-64, 13-15 Aug. 2014

[3] Apache Hadoop," http://hadoop.apache.org/".

[4] Apache Hive," https://hive.apache.org/".

[5] "SQL.Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/SQL.

[6] "PostgreSQL," http://www.postgresql.org/.

[7] A.Thusoo, et. al. Hive : a warehousing solution over a map-reduce framework. Facebook Data Infrastructure Team. 2009.

[8] Q. Wang, et.al. On The Correctness Criteria of Fine-Grained Access Control in Relational Databases. In Proceedings of VLDB, 2007.

[9] McGuire T., Manyika J., Chui M., July / August 2012, "Why Big Data is the New Competitive Advantage", Ivey Business Journal, www.iveybusinessjournal.com/topics/strategy/why-big-data-is-thenew-competitive-advantage

[10] Mark B., "Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data". Gartner, June 27, 2011, http://www.gartner.com/newsroom/id/1731916

[11] Johnson E. J., July/August 2012, "Big Data + Big Analytics = Big Opportunity", Journal of Financial Executive, pp. 1-4.

[12] Nichols, W., March 2013, "Advertising Analytics 2.0", Harvard Business Review, 91(3): 60-68.

[13] Thusoo, Ashish, et al, "Hive: a warehousing solution over a map-reduce framework," Proceedings of the VLDB Endowment, vol.2, no.4, pp.1626-1629, 2009.

[14] B. Arres, N. Kabbachi and O. Boussaid, "Building OLAP cubes on a Cloud Computing environment with MapReduce", Computer Systems and Applications (AICCSA), ACS International Conference, 27-30 May, 2013.

[15] "Aqua Data Studio," http://www.aquafold.com/.

[16] SHVACHKO, Konstantin, et al, "The hadoop distributed file system," in: Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, IEEE, pp. 1-10,2010.

[17] Thusoo, Ashish, et al, "Hive: a warehousing solution over a map-reduce framework," Proceedings of the VLDB Endowment, vol.2, no.4, pp.1626-1629, 2009.

[18] "Teradata," http://www.teradata.com/.

[19] "Greenplum database," http://www.greenplum.com/.

[20] Dean, Jeffrey, and S. Ghemawat., "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol.51, no.1, pp.107-113, 2008.

[21] "Hadoop," http://hadoop.apache.org/.

[22] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad:Distributed data-parallel programs from sequential building blocks," in Proc. of the European Conference on Computer Systems (EuroSys), 2007, pp. 59–72.

[23] ISO/IEC 9075-*:2003, Database Languages - SQL. ISO, Geneva, Switzerland.

[24] Y. Yu, M. Isard, D. Fetterly, M. Budiu, U. Erlingsson, P. K. Gunda, and J. Currey, "DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language," 2008.

[25] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig latin: a not-so-foreign language for data processing," in Proc. of the SIGMOD Conf., 2008, pp. 1099–1110.

[26] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan, "Interpreting the data: Parallel analysis with Sawzall," Scientific Programming, vol. 13, no. 4, 2005.

[27] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, R. Murthy, "Hive - A Warehousing Solution Over a Map-Reduce Framework," In Proc. of Very Large Data Bases, vol. 2 no. 2, August 2009, pp. 1626-1629.

[28] C. Ballinger, "Born to be parallel: Why parallel origins give Teradata database an enduring performance edge," http://www.teradata.com/t/page/87083/index.html.

[29] "Vertica, inc." http://www.vertica.com/.

[30] "IBM zSeries SYSPLEX," http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=/com.ibm.db2.doc.admin/xf6495.htm.

[31] A. Pruscino, "Oracle RAC: Architecture and performance," in Proc. of the SIGMOD Conf., 2003, p. 635.

[32] H. Boral, W. Alexander, L. Clay, G. Copeland, S. Danforth, M. Franklin, B. Hart, M. Smith, and P. Valduriez, "Prototyping Bubba, a highly parallel database system," IEEE TKDE, vol. 2, no. 1, pp. 4–24, 1990.

[33] L. Chen, C. Olston, and R. Ramakrishnan, "Parallel evaluation of composite aggregate queries," in Proc. of the 24th International Conference on Data Engineering (ICDE), 2008.

[34] A. Deshpande and L. Hellerstein, "Flow algorithms for parallel query optimization," in Proc. of the 24th International Conference on Data Engineering (ICDE), 2008.

[35] D. DeWitt and J. Gray, "Parallel database systems: the future of high performance database systems," Communications of the ACM, vol. 35, no. 6, pp. 85–98, 1992.

[36] D. J. Dewitt, S. Ghandeharizadeh, D. A. Schneider, A. Bricker, H. I. Hsiao, and R. Rasmussen, "The Gamma database machine project," IEEE TKDE, vol. 2, no. 1, pp. 44–62, 1990.

[37] G. Graefe, "Encapsulation of parallelism in the Volcano query processing system," SIGMOD Record, vol. 19, no. 2, pp. 102–111, 1990.

[38] Y. Xu, P. Kostamaa, X. Zhou, and L. Chen, "Handling data skew in parallel joins in shared-nothing systems," in Proc. of the SIGMOD Conf., 2008, pp. 1043–1052.

 ISBN: 978-1-61208-444-2

StayActive: An Application for Detecting Stress

Panagiotis Kostopoulos, Tiago Nunes, Kevin Salvi, Mauricio Togneri and Michel Deriaz

Information Science Institute, GSEM/CUI
University of Geneva
Geneva, Switzerland
Email: {panagiotis.kostopoulos, tiago.nunes, kevin.salvi, michel.deriaz}@unige.ch, mauricio.togneri@gmail.com

***Abstract*—In today's society, working environments are becoming more stressful and people working in these environments become prone to various illnesses. But, work should be a source of health, pride and happiness, in the sense of enhancing motivation and strengthening personal development. In this work, we present StayActive, a system which aims to detect stress and burn-out risks by analyzing the behaviour of the users via their smartphone. In particular, we collect data from people's daily phone usage gathering information about the sleeping pattern, the social interaction and the physical activity of the user. We assign a weight factor to each of these three dimensions of wellbeing according to the user's personal perception and build a stress detection system. We evaluate our system in a real world environment and in a daily-routine scenario. This paper highlights the architecture and model of this innovative stress detection system.**

Keywords*—*Stress Detection; Smartphone; Sleeping Pattern; Social Interaction; Physical Activity.

I. INTRODUCTION

Today stress is omnipresent as never before and it is one of the major problems in modern society. Detecting stress in natural environments is beneficial to avoid developing burn-out situations and illness.

The most common method to quantify stress is to simply ask people about their mood filling in questionnaires. There are standard methods for doing so like the Perceived Stress Scale questionnaire [1]. Questions in the perceived stress scale (PSS) assess to what degree a subject feels stressed in a given situation.

Nowadays wearable devices such as mobile phones and wearable sensors are ubiquitous in our lives. Several researchers have tried to understand personality from mobile phone usage [2][3]. Our stress detection system aims to use technology to recognize stress levels using data from the devices that users always carry and wear.

Sleeping patterns, social life and physical activity are connected with the presence of stress in people's lives [4]. We take into account these three dimensions for building our stress detection system.

The rest of this paper is organized as follows. In Section II, our designed stress detection system is described in detail. Experimental results using real data are reported and discussed in Section III. Future work to be done on StayActive is presented in Section IV. Finally, a brief conclusion is drawn in Section V.

II. SYSTEM DESIGN

StayActive is an Android application running on a smartphone. We have chosen the Android based solution because it is an open source framework designed for mobile devices. The Android Software Development Kit (SDK) provides the Application Programming Interface (API) libraries and developer tools necessary to build, test and debug applications for Android. We implemented the prototype in Java using the Android SDK API 19.

A. System overview

Although there are still several open questions regarding the links between the behaviour of a person and their stress level, in StayActive we take a pragmatic approach and build an initial stress detection system which can be extended and refined.

The general architecture of our stress detection system is given in Figure 1.

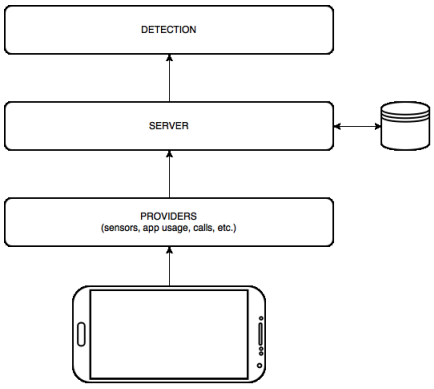


Figure 1. StayActive system architecture.

1) Providers: The first layer is the one that collects and provides the data to upper layers. The provider module contains all the implemented data providers, which are responsible for collecting a specific type of data from the device. They are free to implement the data monitoring behaviour as they wish. The currently implemented providers collect the following type of data: type of physical activity, calls and SMS, ambient light and temperature, location, battery level, screen on/off intervals, Wi-Fi, step counter, number of screen touches and finally type

 ISBN: 978-1-61208-444-2

of applications launched. We give some examples of the results of these providers in Figures 2- 6.

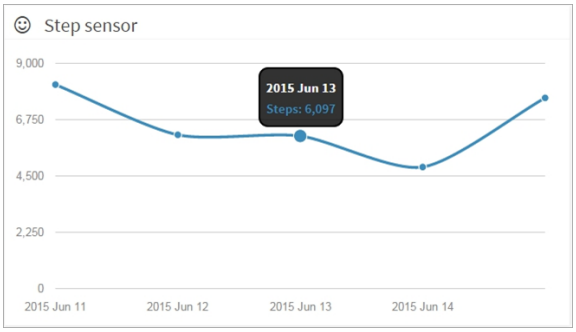


Figure 2. Step counter provider.

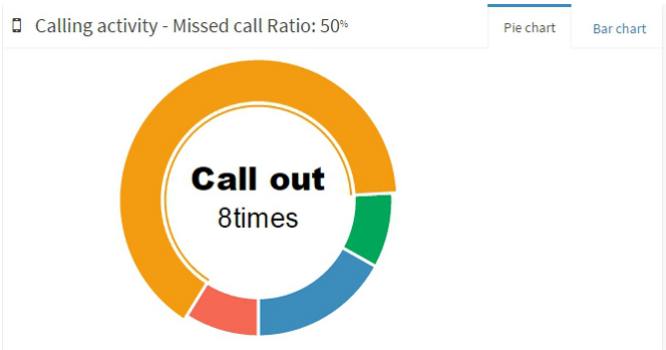


Figure 3. Call provider.

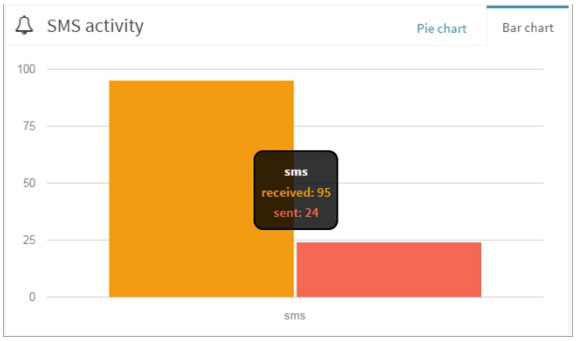


Figure 4. SMS provider.

2) Server: The server module is responsible for receiving data from the mobile devices and storing it in a database. We aggregate all the data and we process it in order to extract a relaxation score for each user as explained in the next section.

3) Detection: This module contains analyzers for each data provider, which extract useful information and patterns from the raw data to output a partial relaxation score. The core detector module will aggregate the results of these individual analyzers and compute a final stress level, as explained in the next section.

B. Stress detection

Simply collecting the patterns of people's behaviour is insufficient for helping them improve their personal wellbeing. It

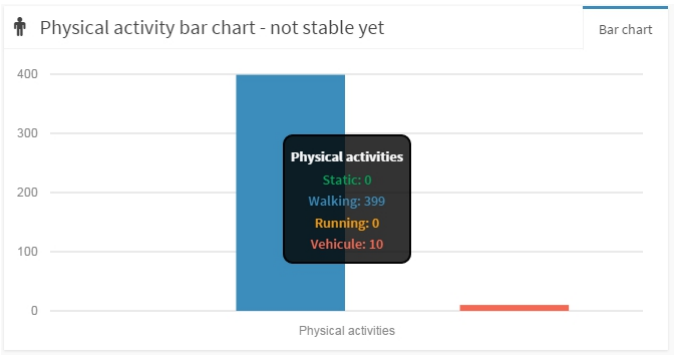


Figure 5. Physical activity provider.

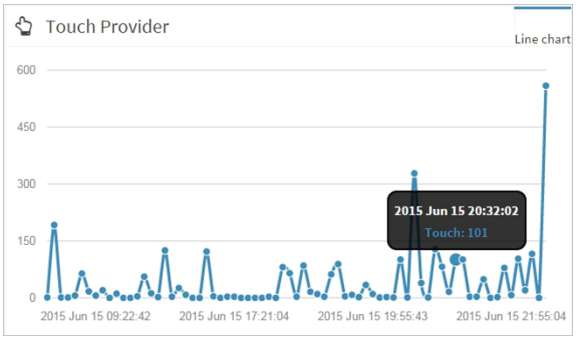


Figure 6. Screen touch provider.

is important to use different dimensions of people's wellbeing and compute their stress level. That way, we will be able to help them by giving advice for reducing their stress level and therefore improving their quality of life. Our stress detection module takes into account three main dimensions of wellbeing: the sleeping pattern of the users, their social interaction and their physical activity as reported in Table I.

TABLE I. Factors measuring stress.

Sleeping pattern	Social interaction	Physical activity
sleeping hours/day	touches of the screen/day	number of steps per/day

1) Sleeping pattern: There is a big body of research work which analyzes the link between sleep hygiene and the mood of people [5][6]. People usually exchange sleep for additional working hours as a coping mechanism for busy lifestyles. In our stress detection module we take into account the user's duration of sleep. We set the number of normal sleeping hours at 8 and penalize insufficient sleep and oversleeping. We set the lower threshold of normal sleeping hours at 7 and the upper threshold at 9 hours according to [7]. For any extra missing or more hours of sleep we penalize the behaviour of the user with a weight factor per hour. In order to compute the sleeping pattern of the user we take into account the Screen analyzer. Between 6 p.m. and 10 a.m. we compute the biggest time interval that the user did not touch his screen and we compute the duration of his sleep. An example of the sleeping pattern of a user for some days is depicted in Figure 7.

2) Social interaction: The daily social interaction of people has a serious impact on many dimensions of wellbeing [6].

 ISBN: 978-1-61208-444-2

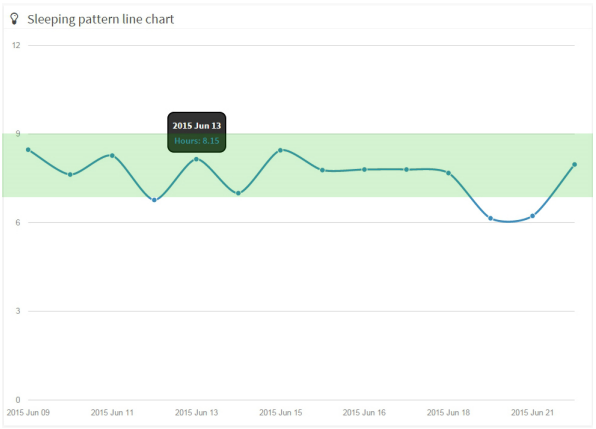


Figure 7. Sleeping pattern.

People who maintain dense social connections are more likely to have resilient mental health. They tend to be able to cope with stress and often are better able to manage chronic illness.

On the other hand regarding communication, researchers are hypothesizing that perhaps people become so used to and even dependent on receiving constant messages, emails, and tweets, that the moment they do not receive one, their anxiety increases. People feel compelled to check their phone constantly, which can then lead to disappointment when there are no new messages, and increased stress about why no one is messaging them, or when the next message might come.

Repetitive checking of mobile phones is considered a compulsive behaviour [8]. People who are highly dependent on the Internet for interaction act impulsively, avoid emotions, and fail to keep up a proper planning or time management [9]. We identify features which are relevant for detecting problematic phone usage and therefore increase the stress level of the user.

For the moment in our system we take into account the number of touches of the screen as a factor for the social interaction of the users using their smartphones. The accumulated result per day is multiplied with the corresponding weight factor and therefore it is accumulated in the total relaxation score. In the next StayActive version we plan to take into account per day, the number of calls and SMS's that the person received, the number of touches of the screen, the number of times the screen was turned on and off and the number of social applications the person used.

3) Physical activity: Several studies have linked exercise to improved depression, self-esteem and stress [10][11]. Our system monitors the physical activity of the user, making the distinction between the type of activity (e.g. walking, running, bicycling). We have also implemented a step counter which gives us the opportunity to find the number of steps that each user took per day. The American Heart Association uses the 10,000 steps metric as a guideline to follow for improving health and decreasing risk of heart disease, the leading cause of death in America. 10,000 steps a day is a rough equivalent to the Surgeon Generals recommendation to accumulate 30 minutes of activity most days of the week.

In our model we assign the maximum value of wellbeing, and therefore the lowest stress level, when reaching the goal of 10,000 steps per day. If someone reaches less than this number we penalize (decrease relaxation factor) with a weight factor per 1,000 steps.

III. Evaluation with real data

For the evaluation of our data, we followed an empirical model. We monitored the behaviour of the user in the above mentioned three dimensions (sleeping pattern, social interaction and physical activity) collecting data for a week.

A. Relaxation score

At first we compute a relaxation score for each individual user for every day of the monitoring week. The relaxation score is in the scale of [0-10] where the more stressed you are, the lower your score will be (so the more relaxed you are the higher your relaxation score). The idea of the scoring procedure is the following. We assign a weight factor to each of the three dimensions of wellbeing that we have taken into account in our study. This factor is based on the response of the participants to the following question which was asked in the beginning of the experiment. Which of the three dimensions do they personally consider as the most important for their wellbeing? To the most important dimension we assign a weight of $w_1 = 0.4$ and to the rest we assign a weight of 0.3 respectively ($w_2 = w_3 = 0.3$), so that $w_1+w_2+w_3 = 1$. Based on these factors we are able to calculate the per day relaxation level of each person as depicted in Figure 8 according to the Equation 1. Therefore we compute a result per dimension and adding them we calculate the final daily relaxation factor of the user. For each of the three dimensions we normalize the results in the scale of [0-10] and then multiply each of them with the respective factor. Adding the three results per user, per day we extract the daily relaxation level of each user.

$$relaxation \quad score = w_1 * dm_1 + w_2 * dm_2 + w_3 * dm_3 \tag{1}$$

TABLE II. Dimensions of Equation 1.

Dimension 1 ($\mathbf{dm_1}$)	**Dimension 2 ($\mathbf{dm_2}$)**	**Dimension 3 ($\mathbf{dm_3}$)**
sleeping hours/day	touches of the screen/day	number of steps/day

B. Preliminary results

The three participants of our first tests were young adult members from our research group. The evaluation of the results takes place by asking the people who participated in the experiment how they felt on each day corresponding to the monitoring week when data was collected, without knowing the outcome. Then, we compare their personal perception with the relaxation score that we have computed using the StayActive application for each individual day. The more score you have the less stressed you are. This is the first step of evaluating the accuracy of the relaxation score that we produced through our empirical model. Secondly, we extract

 ISBN: 978-1-61208-444-2

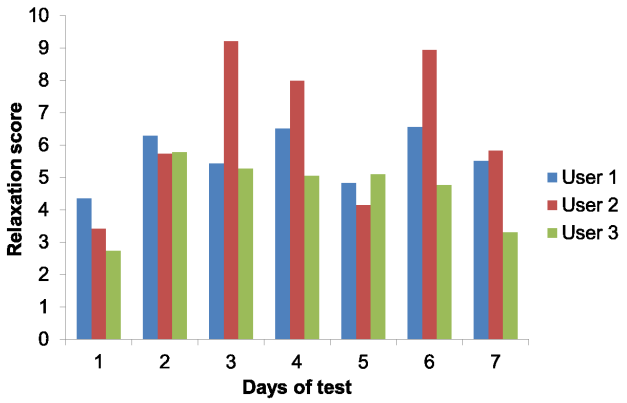


Figure 8. Relaxation scores.

a pattern for the behaviour of the user based on the data that we collected during the testing week and then we compare this pattern with the average daily activity of the user for this week. We calculate the deviation from the normal behaviour that we have extracted from the seven-day experiment and based on that we characterize the user as stressed or not. We also calculate the mean of the stress factor for each person during a week in order to have more robust and accurate data.

IV. Future work

This is a first model of our stress detection system. We are still enhancing and improving it. The immediate steps after the work that has been presented are the following. Further use of the application collecting data for a month and comparison of the different stress results of each individual day with the means that we have extracted from the initial testing period. In the long term, we are targeting a final machine learning approach which will take more features into account in order to improve the accuracy of stress detection. We will create a pattern per user based on his behaviour for a month and a possible deviation from this pattern because of the sleeping hours, the social interaction or the physical activity of the user should agree with the respective decrease in the relaxation score (more stressed). The long term idea of StayActive is to provide older adults with a personalized, adaptable tool which can also monitor some changes to biological signals like skin conductance and heart rate, using wearable sensors and link them to a low relaxation score (increased stress level). Then it will recommend and present various relaxation activities just-in-time in order to allow the users to carry out and solve everyday tasks and problems at work.

V. Conclusions

Stress detection is a research field that has a big impact on the improvement of people's daily life. In this paper we present a first prototype which takes into account three main dimensions of wellbeing. The sleeping pattern, the physical activity of the users and their social interaction are accumulated with different weight factors and give an estimation of the daily stress level of the user. To the best of our knowledge, this is the first system that computes a relaxation score based on different dimensions of human wellbeing.

Acknowledgements

This work was co-funded by the State Secretariat for Education, Research and Innovation of the Swiss federal government and the European Union, in the frame of the EU AAL project StayActive (aal-2013-6-126).

References

[1] S. Cohen, T. W. Kamarck, and R. Mermelstein, "A global measure of perceived stress," in *Journal of Health and Social Behavior*, 1983, pp. 1027–1035.

[2] A. Sano and R. W. Picard, "Stress recognition using wearable sensors and mobile phones," in *Humaine Association Conference on Affective Computing and Intelligent Interaction*, 2013, pp. 671–676.

[3] A. Muaremi, B. Arnrich, and G. Trster, "Towards measuring stress with smartphones andwearable devices duringworkday and sleep," in *BioNanoSci*, 2013, pp. 172–183.

[4] R. Norris, D. Carroll, and R. Cochrane, "The effects of physical activity and exercise training on psychological stress and well-being in an adolescent population," in *Journal of Psychosomatic Research*, 1992, pp. 55–65.

[5] S. Moturu, I. Khayal, N. Aharony, W. Pan, and A. Pentland, "Sleep, mood and sociability in a healthy population," in *33rd Annual International Conference of the IEEE EMBS*, 2011, pp. 5267–5270.

[6] N. L. et al., "Bewell: A smartphone application to monitor, model and promote wellbeing," in *5th ICST/IEEE Conference on Pervasive Computing Technologies for Healthcare IEEE Press*, 2011, pp. 23–26.

[7] G. Alvarez and N. Ayas, "The impact of daily sleep duration on health: A review of the literature," in *Progress in Cardiovascular Nursing*, 2004, pp. 56–59.

[8] A. Oulasvirta, T. Rattenbury, L. Ma, and E. Raita, "Habits make smartphone use more pervasive," in *Personal and Ubiquitous Computing*, 2012, pp. 105–114.

[9] S. Li and T. Chung, "Internet function and internet addictive behavior," in *Computers in Human Behavioor*, 2006, pp. 1067–1071.

[10] K. Fox, "The influence of physical activity on mental well-being," in *Public Health Nutritionl*, 1999, pp. 411–418.

[11] A. W. R.S. Paffenbarger, R. Hyde and C. Hsieh, "Physical activity, all-cause mortality, and longevity of college alumni," in *New England journal of medicine*, 1986, pp. 605–613.

 ISBN: 978-1-61208-444-2

Smart Position Selection in Mobile Localisation

Carlos Martínez de la Osa, Grigorios G. Anagnostopoulos, Michel Deriaz

Information Science Institute
GSEM/CUI
University of Geneva
Geneva, Switzerland
Email: {carlos.martinez, grigorios.anagnostopoulos, michel.deriaz}@unige.ch

***Abstract*—Which technology should be used in order to be able to locate oneself in any kind of scenario? This has been a recurrent question in the last years. It has become evident that, until now, there is no dominant indoor positioning solution based on a single technology. Outdoors, positioning systems based on satellites have given excellent results. However, a global solution for both kinds of scenarios does not exist. In our study, this problem is dealt with by creating an algorithm able to evaluate positions received from different technologies and choose the most trustworthy one. As a result, we are able to improve the overall accuracy of the user's position estimation, compared to the ones the different technologies would have given if used independently. In this way, the user is offered a simple solution to have an accurate position in all environments, in a transparent way. The main challenge of using different technologies at the same time is usually the battery consumption. A solution for dealing with this aspect is also proposed in this document. This research has been done in the context of the Ambient Assisted Living (AAL) Enhanced Daily Living and Health (EDLAH) project, where older people can track their lost objects, which requires them to be positioned in a very accurate way.**

***Keywords*—Indoor localisation; Outdoor localisation; Position selection; Heterogeneous positioning; Battery saving.**

I. INTRODUCTION

The ability to position people indoors has become a very important focus of research in the last years. An example of this are the requirements given in the European project EDLAH, that motivates this research, where the goal is for users to be able to locate some of their lost objects in a map of their house. It is also required to position these users in a very accurate way using their mobile devices.

Outdoor positioning is now excellent with the establishment of Global Positioning System (GPS), but the number of applications that demand positioning abilities in all environments is increasing rapidly. On the other hand, it appears that, until now, there is not a dominant solution based on a single technology able to offer better results than the rest for these cases.

One of the commonly used technologies for positioning in indoor environments is the Wi-Fi signal [1][2]. This approach takes advantage of the fact that most buildings have several Wi-Fi access points, in order to provide Internet access, so the hardware required is already installed. On the other hand, usually the access point network is not dense enough to facilitate a satisfactory precision of localisation. Another technology widely used during the last years is the Bluetooth Low Energy (BLE) technology [3][4]. It has a low energy consumption, while maintaining a communication range similar to that of its predecessor, Classic Bluetooth. Some other approaches combine these methods with the inertial sensors of the device used to improve the accuracy and the experience of the user in between position estimation receptions [5][6].

An important challenge for applications that need to offer positioning globally, both indoors and outdoors, is to have an efficient mechanism that decides which position provider should be used. In our study, we face this problem by creating an algorithm able to gather positions received from different technologies, evaluate them and choose the most trustworthy one, therefore improving the overall accuracy of the user's position estimation. A similar approach to this solution can be found in [7], where the concept "Quality of Position" is presented.

This is one of the problems that must be faced in the EDLAH project, where the users must be located with high precision in their own flat, as well as outdoors in a garden or common area, in order to find their lost objects. These objects have been previously identified with a BLE beacon that allows a mobile device to compute the distance to them, as described in [8].

In this work, the position providers offered by Google in the Android operative system have been used, both GPS and Cell-ID based position provider [9]. Also, the BLE positioning solution presented in [3], where a grid of BLE beacons is deployed. This allows the position of the user to be inferred from a weighted average of the Received Signal Strength Indication (RSSI) values, which were received from the different beacons in range. In this case, the position estimation is limited to the area that is defined by the polygon that the beacons' placement creates. This way if the beacons are placed in an indoors environment, the position estimation will only be calculated indoors.

There exist also several studies about power saving in mobile positioning, giving an overview of current localisation technologies and a classification of techniques for improving the energy efficiency by evaluating some of the most promising approaches [10][11][12].

 ISBN: 978-1-61208-444-2

The rest of this paper is organised as follows. In Section II, we present the position selection algorithm and the concepts of position estimation and position trust. Experimental results and their corresponding analysis are shown in Section III. In Section IV, we present a theoretical model for battery saving. Finally, future work directions along with conclusions drawn are presented in Section V.

II. Position selection

The core of this research is based on the existence of position providers. Conceptually, position providers constitute the lowest layer of a location based service. They transform raw sensor data into position estimations. In our system, these estimations are sent to higher level layers.

A. Position estimation

Firstly, it is necessary to establish the attributes that a position estimation must have in order to be suitable for the algorithm. These are:

- Latitude
- Longitude
- Accuracy
- Provider name
- Timestamp

The two basic parameters that a position must have are the latitude and longitude coordinates, as they allow the identification of a specific point in the geographic coordinate system.

As the position is not exact, but an estimate, it is needed to have an idea of the quality of that estimation. This is given by the dynamic accuracy estimation (referred in this paper as accuracy), which is generally described as the radius of 68% confidence of the position. In other words, if a circle centred at the position's latitude and longitude is drawn, and with a radius equal to the accuracy, then there is a 68% probability that the true position is inside the circle. This is because it is assumed that location errors are random with a normal distribution, so the 68% confidence circle represents one standard deviation. In practice though, location errors do not always follow such a simple distribution.

Moreover, the name of the provider that estimated the position must be delivered in order to give the user and the system information about the technology used. This is specially important for the battery saving algorithm, as it makes it possible to differentiate between different providers. Finally, the timestamp of when the position was recorded is also required, which is basic to have an idea of how recent each estimation is.

Additionally, the position estimation might also contain information about the altitude, the speed, the bearing of the user, etc. These are given to upper layers, but are out of the scope of this research.

B. Position trust

We define the position trust as an internal parameter of the algorithm utilized to determine which of the available positions is the best at each moment. The calculation of the trust is based on three parameters:

- Accuracy: as stated before, it is described as the radius of 68% confidence of the position, in meters. As explained, this error is also an estimation, because the system does not know what the exact real position of the person is. The accurate modelling of the accuracy estimation is vital for the correct behaviour of our approach. Generally, this estimation is based on some of the received characteristics of the raw data of the technology used. In GPS and the Cell-ID provider it is given by Android, while in the BLE provider we base this estimation on the strength of the received Bluetooth signal and the number of Bluetooth beacons in sight.
- Recency: in seconds, the difference between the timestamp obtained when the position was estimated and the actual timestamp at that moment.
- Priority: optionally, a value from 1 to 10, with 1 being the highest priority that can be assigned by the user to the different technologies used when initialising the algorithm. This is utilised in case the user prefers specific providers above others, even if the algorithm would choose a different position estimation if this priority would not exist. If priorities are not assigned, this parameter will not be taken into account in the selection process.

The trust of the position will be inversely proportional to the accuracy, which is given as the estimated error committed in the measurement, in meters. Therefore, the smaller the error, the bigger the trust. The trust will also be inversely proportional to the recency of the position, thus the newer the position, the higher the trust. If the user has given priorities to the different providers, this value will also be taken into account when calculating the trust. This will be inversely proportional to the value of the priority. Following, in (1), (2) and (3), the way that the trust is calculated, as well as the restrictions of the weight values are presented.

$$trust = \frac{w_1}{accuracy} + \frac{w_2}{recency} + \frac{w_3}{priority} \quad (1)$$

$$w_1 + w_2 + w_3 = 1 \quad (2)$$

$$w_i \geq 0\ ,\ i\,\epsilon\{1,2,3\} \quad (3)$$

The values w_1, w_2 and w_3 correspond to the weight, or the importance, given to the accuracy, the recency of the position and the priority of the provider used, respectively. These weights can be tuned, following (2) and (3), in order to obtain the most adequate results for each scenario or preference of the user.

 ISBN: 978-1-61208-444-2

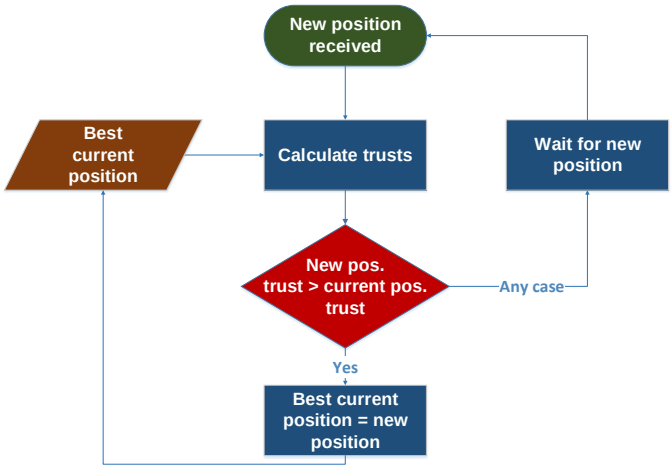


Figure 1. Flow chart of the position selection algorithm

C. Position selection

Each time that one of the position providers sends a new position estimation, its trust will be calculated and compared with the recalculated trust of the last best position estimation. If the trust of the new position is better, the algorithm will save it as the best possible position estimation at the moment and it will be returned to the user as an output. If the previous position estimation still has a better trust, then no update occurs. It is important to note that the trust of the last best position estimation will be recalculated every time a provider offers a new position. This is done because the recency of the previous position estimation must be updated and, therefore, its trust will decrease. The procedure's logic can be visualised in Figure 1.

III. Results

We will now proceed to show the results achieved by using the position selection algorithm in our test environment at the University of Geneva. Concretely, the algorithm is tested using three different position providers: GPS, Cell-ID positioning, and a BLE provider. The first two providers are the ones provided by Google in Android mobile devices, while the BLE one has been developed by our group. The position selection algorithm, in the following example, has been configured using $w_1 = w_2$ and $w_3 = 0$ in (1). This means that, when calculating the trust of a position estimation, one meter in the accuracy is penalized the same way as one second in the recency of the estimation. This setup would be the logical solution for a user that is moving at a speed of one meter per second. In this example, the priorities are not taken into account. As an example, a position that was taken one second ago with an accuracy of one meter, will have the same trust as a position calculated right now with an accuracy of two meters. In this situation, the algorithm will choose the newer position estimation.

To measure the results of the algorithm, we have created a tool that allows us to record the actual real position of a user, as he or she is moving, at the same time that the estimated positions are being calculated. Therefore, we can later calculate the error committed in the position estimations of a moving or static user. In our case, we have selected a path that mixes different types of scenarios. The user started the trip in an indoors area with no BLE coverage (and being indoors the GPS coverage is almost non-existent), followed by an indoors zone with BLE coverage. Later, the path continues outdoors with no BLE coverage and finishes again indoors with BLE coverage. The goal of this procedure is to test how the algorithm handles the changes between areas where one technology has much better accuracy than others.

TABLE I. Result for different position providers

Provider name	*Mean error (m)*	*SD (m)*	*SR*
GPS	13.7	11.68	41.01%
Cell-ID	31.69	20.07	62.96%
BLE	6.51	15.2	21.78%
GPStoBLE	5.76	7.53	39.57%
Our solution	4.84	6.42	45.34%

A list of the results can be seen in Table I . These have been extracted using a Samsung Galaxy S4 device. The results have been taken from five different position providers: standalone GPS, standalone Cell-ID position provider, standalone BLE position provider, GPStoBLE and our solution. GPStoBLE is a position provider previously created in our research group, that is specifically designed to switch between the BLE provider and GPS. It takes into account the specific characteristics of both providers and waits until several good readings of one of the providers are received to decide which of the two will be used.

In Table I, for each of these providers, the mean error committed in the estimations is shown, measured as the distance between the estimated position and the real one, in meters. Additionally, the standard deviation of the error is also specified, in order to offer a better idea about the dispersion of the results. Lastly, a parameter defined as Success Rate (SR), can be observed related to the estimated accuracy claimed by the providers. It indicates the percentage of the times that the real position was inside the area delimited by the circle with the estimated position as a center and a radius equal to the accuracy. As described in Section II, this value is expected to be close to 68%. Nevertheless, it is observed that this is not true for most of the providers, which means that the estimation of the accuracy should be improved in these cases.

Looking at the data obtained, it is directly seen that the standalone solutions are more inaccurate than the other two. Furthermore, the BLE and GPS solutions do not offer coverage everywhere. For example, even if the total error committed by the BLE position provider is relatively low, the error outdoors is unaccounted for as the provider is not estimating positions. Nevertheless, the user would not be able to position himself at that moment. It is notable that our solution, which is not provider specific, has a better performance compared to the one specifically designed for BLE and GPS, due to the introduction of a third position provider (Cell-ID) in areas

 ISBN: 978-1-61208-444-2

where none of the previous ones have coverage. The superior performance is also due to the removal of bad estimations, because when a new position is received with a bad accuracy, the algorithm will most likely keep using the previous position estimation as it would have more trust. The visual difference between the real path followed by the user and the one estimated by our solution can be observed in figures 2 and 3.

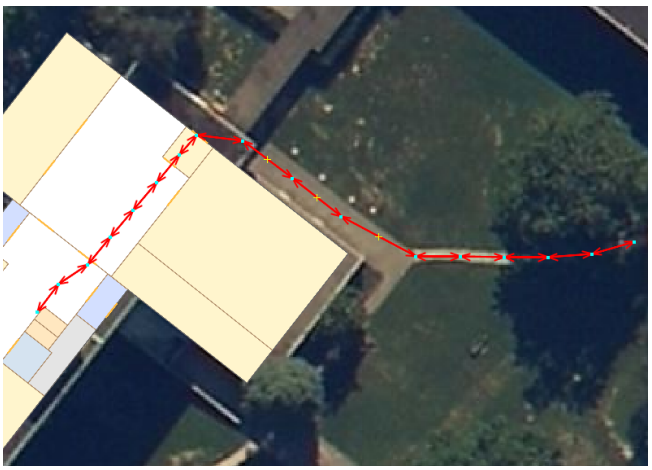


Figure 2. Real path followed by the user

Figure 3. Estimated path by our solution

Lastly, a parameter optimization step is made, in order to find a better parameter tuning for our algorithm. In Table II the results of five different setting of w_1 and w_2, are exemplified. The parameter w_3, representing the weight of the priority given to each provider, is again set to zero. Values were given following (2) and (3). The values vary from 0.3 to 0.7 for both parameters. Higher or lower than that, the algorithm starts giving poor results. If the accuracy has a weight lower than 0.25 the algorithm will start giving only new positions, taking very little into account the estimated error committed. Similarly, if w_2 is very low, the algorithm will offer a position update only when it receives a more accurate position, even if the saved one was taken a long time ago.

TABLE II. RESULT FOR DIFFERENT ALGORITHM WEIGHTS

w_1	w_2	*Mean error (m)*	*SD (m)*	*SR*
0.3	0.7	4.84	6.37	46.28%
0.4	0.6	4.81	6.36	47.44%
0.5	0.5	4.84	6.42	45.35%
0.6	0.4	4.90	6.55	40.46%
0.7	0.3	4.92	6.68	38.60%

It is appreciated how slightly better results are obtained for $w_1 = 0.4$ and $w_2 = 0.6$. This means that the estimations are closer to the real path of the user when the recency of the updates is given slightly higher importance than the estimated accuracy. This result might also be due to the accuracy estimation not having a 68% confidence as it should, but significantly lower in most of the cases.

IV. BATTERY SAVING

One of the downsides of this approach and, in general, of all heterogeneous positioning solutions, is that the device needs to have activated all technologies and be subscribed to their corresponding position providers at all times, which translates into an elevated battery consumption. For this reason, it is here presented a theoretical model on how to apply battery saving techniques to the solution presented in this paper. This model is based on controlling the switching, on or off, of the different position providers depending on how they are needed. There are no experimental results offered for this model yet, as it is an ongoing work in our group.

The algorithm applied in this case checks, on every position update, the trust of the position estimations offered by the different active providers. The main idea of the algorithm is to classify providers as reliable when they offer a number of trustworthy positions in a row, and, in a similar way, classify them as unreliable if they give a number of untrustworthy position estimations in a row.

The algorithm is iterated every time there is a position update. It checks the trust of the position received, if this trust is higher than a predefined trust value, a specific counter for this provider is increased. The counter is reset to zero every time the trust is lower than this value. When the algorithm detects that the counter is higher than a confidence threshold, it means that the provider has given several trustworthy positions in a row, so it is classified as a reliable position provider. If the system detects that there are several reliable providers at the same moment, it deactivates the ones with the highest power consumption.

Similarly, when a reliable provider gives a series of untrustworthy positions in a row, it is classified as not reliable and, if it is the only active provider at that moment, the rest of the providers will be reactivated again in order to find positions with higher trust. This logic can be visualised in Figure 4.

Besides, if the only provider used at a given moment consumes a high amount of power, a timer will be activated so that every given amount of time, providers that have a lower battery consumption are reactivated to check if they became

 ISBN: 978-1-61208-444-2

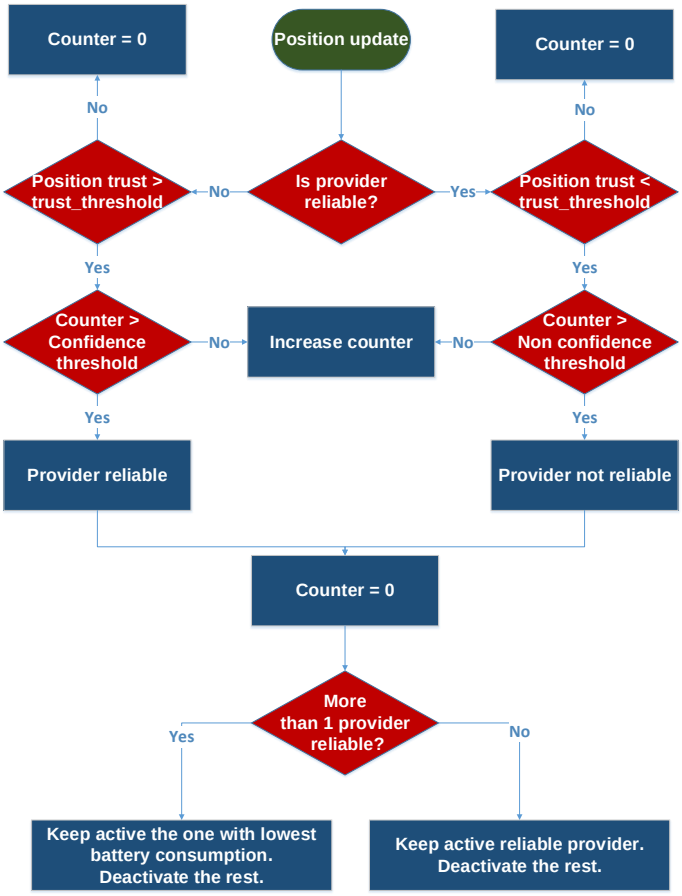


Figure 4. Flow chart of the battery saving algorithm

trustworthy and use one of them instead of the one with high battery consumption demand.

In order for this algorithm to be effective, it is important that, when defining a new provider, a parameter which indicates the energy consumption rate of this technology is specified. This parameter does not need to be a quantitative measure, but may just be a qualitative description. As an example, in our system, GPS is marked as High Consumption, while the BLE position provider is marked as Low Consumption.

The confidence thresholds that define when a provider is reliable or not, as well as the timer set for high power consumption providers, are meant to be tuned by the user of the algorithm. It is evident that there exists a trade-off between battery consumption and position accuracy. Having very low values on these parameters will imply a high amount of activating and deactivating providers, but it will also mean that more position estimations of different providers are received, improving the chances of getting more accurate estimations, but also increasing the battery consumption. On the other hand, higher values will imply less changes of providers, less battery consumption, and, most likely, less accuracy.

V. Conclusion and Future Work

A switching algorithm between different mobile position providers has been presented along with a battery saving logic. It has been demonstrated how our solution has been able to improve the results previously achieved in our group, which used to rely on position providers based on a single technology, or a switching logic designed for specific providers. The proposed solution is technology independent, since the algorithm simply requires some basic parameters in the position estimation, which offers substantial flexibility for the future addition of new position providers based on other technologies.

The requirements for the project EDLAH have been fulfilled, as the accuracy is improved in all environments, according to the available technologies. This way, the object localization module has a more reliable position estimation input.

Future work in this area involves testing other configurations of the algorithm by adjusting the different weights and thresholds involved. One of the possibilities is the addition of machine learning techniques to optimise the algorithm parameters trying to minimise the average error in the estimation. Additionally, the battery saving theoretical model will be implemented in order to extract results and conclusions from its use.

Acknowledgements

This work was co-funded by the State Secretariat for Education, Research and Innovation of the Swiss federal government and the European Union, in the frame of the EU AAL project EDLAH (aal-2012-5-062).

References

[1] S. Mazuelas *et al.*, "Robust indoor positioning provided by real-time rssi values in unmodified wlan networks." *J. Sel. Topics Signal Processing*, vol. 3, pp. 821–831, 2009.

[2] M. Lee and D. Han, "Voronoi tessellation based interpolation method for wi-fi radio map construction." *IEEE Communications Letters*, vol. 16, no. 3, pp. 404–407, 2012.

[3] G. G. Anagnostopoulos and M. Deriaz, "Accuracy enhancements in indoor localization with the weighted average technique," in *SENSORCOMM 2014*, 2014, pp. 112 – 116.

[4] Z. jianyong *et al.*, "Rssi based bluetooth low energy indoor positioning," in *IPIN 2014*, 2014.

[5] L. Liew and W. S. H. Wong, "Indoor positioning method based on inertial data, rssi and compass from handheld smart-device," 2014, pp. 48 – 52.

[6] U. Shala and A. Rodriguez, "Indoor positioning using sensor-fusion in android devices," 2011.

[7] E. Metola and A. Bernardos, "Poster an embedded fusion system for location management," vol. 104, pp. 233–237, 2012.

[8] G. Ionescu, C. M. de la Osa, and M. Deriaz, "Improving distance estimation in object localisation with bluetooth low energy," in *SENSORCOMM 2014*, 2014, pp. 45 – 50.

[9] Google, "Android Location Strategies," http://developer.android.com/guide/topics/location/strategies.html, 2015, [Online; accessed 24-June-2015].

[10] T. Graf, "Power-efficient positioning technologies for mobile devices," *Berlin University of Technology, Jul*, 2012.

[11] T. Nakagawa *et al.*, "Variable interval positioning method for smartphone-based power-saving geofencing," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, Sept 2013, pp. 3482–3486.

 ISBN: 978-1-61208-444-2

[12] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy aware localization for mobile devices," in *ACM MobiSys 2010*, 2010.

 ISBN: 978-1-61208-444-2

A Simple and Efficient Method for Computing Data Cubes

Viet Phan-Luong
Université Aix-Marseille
LIF - UMR CNRS 6166
Marseille, France
Email: viet.phanluong@lif.univ-mrs.fr

***Abstract*—Based on a construction of the power set of the fact table schemes, this paper presents an approach to represent and to compute data cubes using a prefix tree structure for the storage of cuboids. Though the approach is simple, the experimental results show that it is efficient in run time and storage space.**

Keywords-Data warehouse; Data cube; Data mining.

I. INTRODUCTION

In data warehouse, a data cube of a fact table with n dimensions and m measures can be seen as the result of the set of the Structured Query Language (SQL) group-by queries over the power set of dimensions, with aggregate functions over the measures. The result of each group-by query is an aggregate view, called a cuboid, of the fact table. The concept of data cube represents important interests to business decision as it provides aggregate views of measures over multiple combinations of dimensions. As the number of cuboids in a data cube is exponential to the number of dimensions of the fact table, when the fact table is big, computing a cuboid is critical and computing all cuboids of a data cube is exponentially cost in time [1][5]. To improve the response time, the data cube is usually precomputed and stored on disks [6]. However, storing all the data cube is exponential in space. Research in Online Analytical Processing (OLAP) focuses important effort for efficient methods of computation and representation of data cubes.

A. Related work

There are approaches that represent data cubes approximately or partially [9][10][16][18]. The other approaches search to represent the entire data cube with efficient methods to compute and to store the cube [2][7][21]. The computing time and storage space can be minimized by reducing redundancies between tuples in cuboids [20] or based on equivalence relations defined on aggregate functions [11][15] or on the concept of closed itemsets in frequent itemset mining [14] or by coalescing the storage of tuples in cuboids [19].

In the approaches to efficiently compute and store the cube, the computation is usually organized over the complete lattice of subschemes of the fact table dimension scheme, in such a way the run time and the storage space can be optimized by reducing redundancies [2][8][11][12][13][15][17]. The computation can traverse the complete lattice in a top-down or bottom-up manner. [7][16]. For grouping tuples to create cuboids, the sort operation can be used to reorganize tuples: tuples are grouped over the prefix of their scheme and the aggregate functions are applied over the measures. By grouping tuples, the fact table can be horizontally partitioned, each partition can be fixed in memory, and the cube computation can be modularized.

To optimize the cuboid construction, in top-down methods [7][20], the cuboids over the subschemes on a path from the top to the bottom in the complete lattice can be built in only one lecture of the fact table sorted over the largest scheme of the path. An aggregate filter is used for each subscheme. The filter contains, at each time, only one tuple over the subscheme with the current value of aggregated measures (or a non aggregated mark). When reading a new tuple of the fact table, if over the subscheme, the new tuple has the same value as the filter, then only the value of the aggregated measures is updated. Otherwise, the current content of the filter is flushed out to disk, and before the new tuple passes into the filter, the subtuple over the next subscheme (next on the path from the top to the bottom) goes into the next subscheme filter.

To minimize the storage space of a cuboid, only aggregated subtuples with aggregated measures are directly stored on disk. Non-aggregated subtuples are not stored but represented by references to the (sub)tuples where the non aggregated tuples are originated.

The bottom-up methods [11][13][15][20] walk the paths from the bottom to the top in the complete lattice, beginning with the empty node (corresponding to the cuboid with no dimension). For each path, let T_0 be the scheme at the bottom node and T_n the scheme at of the top node of the path (not necessary the bottom and the top of the lattice, as each node is visited only once). These methods begin by sorting the fact table over T_0 and by this, the fact table is partitioned into groups over T_0. To minimize storage space, for each one of these groups, the following depth-first recursive process is applied [20].

If the group is single, then the only element of the group is represented by a reference to the corresponding tuple in

 ISBN: 978-1-61208-444-2

the fact table, and there is no further process: the recursive cuboid construction is pruned.

Otherwise, an aggregated tuple is created in the cuboid over T_0 and the group is sorted over the next scheme T_1 in the path (with larger scheme) to be partitioned into subgroups. The creation of a real tuple or a reference in the cuboid corresponding to each subgroup over T_1 is similar to what we have done when building the cuboid over T_0.

When the recursive process is pruned at a node $T_i, 0 \leq i \leq n$, or reaches to T_n, it resumes with the next group of the partition over T_0, until all groups of the partition are processed. The construction resumes with the next path, until all paths of the complete lattice are processed, and all cuboids are built.

Note that in the above optimized bottom-up method, in all cuboids, if references exist, they refer directly to tuples in the fact table, not to tuples in other cuboids. This method, named Totally-Redundant-Segment BottomUpCube (TRS-BUC), is reported in [20] as a method that dominates or nearly dominates its competitors in all aspects of the data cube problem: fast computation of a fully materialized cube in compressed form, incrementally updateable, and quick query response time.

B. Contribution

This paper presents a simple and efficient approach to compute and to represent data cube without sorting the fact table or any part of it, neither partitioning the fact table, nor computing the complete lattice of subschemes, nor sophisticated techniques to implement direct or indirect references of tuples in cuboids. The efficient representation of data cube is not only a compact representation of all cuboids of the data cube, but also an efficient method to get the original cuboids from the compact representation. The main ideas of the proposed approach are:

1) Among the cuboids of a data cube, there are ones that can be easily and rapidly get from the others, with no important computing time. We call the latters the prime and next-prime cuboids. These cuboids will be computed and stored on disks.

2) The prime and next-prime cuboids are computed and stored on disk using a prefix tree structure for compact representation. To improve the efficiency of research through the prefix tree, this work integrates the binary search tree into the prefix tree.

3) To compute the prime and next-prime cuboids, this work proposes a running scheme in which the computation of the current cuboids can be speeded up by using the previously computed cuboids.

The paper is organized as follows. Section 2 introduces the concept of the prime and next-prime schemes and cuboids. Section 3 presents the structure of the integrated binary search prefix tree used to store cuboids. Section 4 presents the running scheme to compute the prime and next-prime cuboids and shows how to efficiently get any other cuboids from the prime and next-prime cuboids. Section 5 reports the experimentation results. Finally, conclusion and further work are in Section 6.

II. Prime and next-prime cuboids

This section defines the main concepts of the present approach to compute and to represent data cubes.

A. A structure of the power set

A data cube over a scheme S is the set of cuboids built over all subsets of S, that is the power set of S. As in most of existing work, attributes are encoded in integer, let us consider $S = \{1, 2, ..., n\}$, $n \geq 1$. The power set of S can be recursively defined as follows.

1) The power set of $S_0 = \emptyset$ (the empty set) is $P_0 = \{\emptyset\}$.

2) For $n \geq 1$, the power set of $S_n = \{1, 2, ..., n\}$ can be defined recursively as follows:

$$P_n = P_{n-1} \cup \{X \cup \{n\} \mid X \in P_{n-1}\} \quad (1)$$

P_n is the union of P_{n-1} (the power set of S_{n-1}) and the set of which each element is built by adding n to each element of P_{n-1}. Let us call P_{n-1} the *first-half power set* of S_n and the second operand the *last-half power set* of S_n.

Example: For $n = 3$, $S_3 = \{1, 2, 3\}$, we have:
$P_0 = \{\emptyset\}$, $P_1 = \{\emptyset, \{1\}\}$, $P_2 = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$,
$P_3 = \{\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.
The last-half power set of S_3 is:
$\{\{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.

B. Last-half data cube and first-half data cube

Consider a fact table R (a relational data table) over a dimension scheme $S_n = \{1, 2, ..., n\}$. In view of the first-half and the last-half power set, suppose that $X = x_1, ..., x_i \subseteq S_n$ is an element of the first-half power set of S_n. Let Y be the smallest element of the last-half power set of S_n that contains X. Then, $Y = X \cup \{n\}$. If the cuboid over Y is already computed in the attribute order $x_1, ..., x_i, n$, then the cuboid over $X = x_1, ..., x_i$ can be done by a simple sequential reading of the cuboid over Y to get data for the cuboid over X. So, we call:

– A scheme in the last-half power set a *prime scheme* and a cuboid over a prime scheme a *prime cuboid*. Note that all prime schemes contain the last attribute n and any scheme that contains attribute n is a prime scheme.

– For efficient computing, the prime cuboids are computed by pairs with one dataset access for each pair. Such a pair is composed of two prime cuboids. The scheme of the first one has attribute 1 and the scheme of the second one is obtained from the scheme of the first one by deleting attribute 1. We call the second prime cuboid the next-prime cuboid.

– The set of all cuboids over the prime (or next-prime) schemes is called the last-half data cube. The set of all remaining cuboids is called the first-half data cube. In this

TABLE I. FACT TABLE R1

RowId	A	B	C	D	M
1	a2	b1	c2	d2	m1
2	a3	b2	c2	d2	m2
3	a1	b1	c1	d1	m1
4	a1	b1	c2	d1	m3
5	a3	b3	c2	d3	m2

approach, the last-half data cube is computed and stored on disks. Cuboids in the first-half data cube are computed as queries based on the last-half data cube.

III. INTEGRATED BINARY SEARCH PREFIX TREE

The prefix tree structure offers a compact storage for tuples: the common prefix of tuples is stored once. So, there is no redundancy in storage. Despite the compact structure of the prefix tree, if the same prefix has a large set of different suffixes, then the search time in the set of suffixes can be important. To tackle it, this work proposes to integrate the binary search tree into the prefix structure. The integrated structure, called the *binary search prefix tree (BSPT)*, is used to store tuples of cuboids. With this structure, tuples with the same prefix are stored as follows:

– The prefix is stored once.

– The suffixes of those tuples are organized in siblings and stored in a binary search tree.

Precisely, in C language, the structure is defined by :

```
typedef struct bsptree Bsptree; // Binary search prefix tree
struct bsptree{ Elt data;       // data at a node
                Ltid *ltid;     // list of RowIds
                Bsptree *son, *lsib, *rsib; };
```

where son, $lsib$, and $rsib$ represent respectively the son, the left and the right siblings of nodes. The field $ltid$ is reserved for the list of tuple identifiers ($RowId$) associated with nodes. For efficient memory use, $ltid$ is stored only at the last node of each path in the BSPT. With this representation, each binary search tree contains all siblings of a node in the normal prefix tree. For example, we have:

– Table I represents the fact table R1 over the dimension scheme $ABCD$ and a measure M.

– Figure 1 represents the BSPT of the tuples over the scheme $ABCD$ of the fact table R1, where we suppose that with the same letter x, if $i < j$ then $xi < xj$, e.g., $a1 < a2 < a3$. In Figure1, the continue lines represent the son links and the dash lines represent the lsib or rsib links.

The BSPT is saved to disk with the following format:

$level > suffix : ltids$

where

– level is the length of the prefix part that the path has in common with its left neighbor,

– suffix is a list of elements, and

– ltids is a list of tuple identifiers (RowId).

Cuboids are built using the BSPT structure. The list of RowIds associated with the last node of each path allows the aggregate of measures. For example, with the fact table in Table I, the cuboid over $ABCD$ is saved on disk as the following.

$0 > a1\ b1\ c1\ d1 : 3$
$2 > c2\ d1 : 4$
$0 > a2\ b1\ c2\ d2 : 1$
$0 > a3\ b2\ c2\ d2 : 2$
$1 > b3\ c2\ d3 : 5$

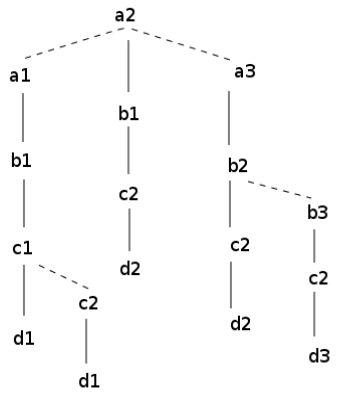


Figure 1. A binary search prefix tree

A. Insertion of tuples in a BSPT

Algorithm Tuple2Ptree: Insert a tuple into a BSPT.
Input: A BSPT represented by node P, a tuple $ldata$ and its list of tids lti.
Output: The tree P updated with $ldata$ and lti.
Method:

```
If (P is null) then
  create P with P->data = head(ldata),
  P->son = P->lsib = P->rsib = NULL;
  if queue(ldata) is null then P->ltid =lti
  else P->son = Tuple2Pree(P->son, queue(ldata), lti);
Else if (P->data > head(ldata)) then
        P->lsib=Tuple2Ptree(P->lsib, ldata,lti);
else if (P->data < head(ldata)) then
      P->rsib=Tuple2Ptree(P->rsib, ldata,lti);
else if queue(ldata) is null then
           P->ltid = insert(P->ltid, lti);
else P->son = Tuple2Ptree(P->son, queue(ldata), lti);
return P;
```

In algorithm Tuple2Ptree, $head(ldata)$ returns the first element of $ldata$ and $queue(ldata)$ returns the queue of $ldata$ after removing $head(ldata)$.

B. Grouping tuples using binary prefix tree

Algorithm Table2Ptree: Build a BSPT for a relational table.
Input: A table R in which each tuple has a list of tids lti.
Output: The BSPT P for R
Method:

Create an empty BSPT P;
For each tuple ldata in R with its list of tids lti do
P = Tuple2Ptree(P, ldata, lti) done;
Return P;

IV. COMPUTING THE LAST-HALF DATA CUBE

Let $S = \{1, 2, ..., n\}$ be the set of all dimensions of the fact table. To compute the last-half data cube:

– We begin by computing the first prime and next-prime cuboids based on the fact table, one over S and the other over $S - \{1\}$.

– Apart the first prime and next-prime cuboids (over S and $S - \{1\}$, respectively), for the current prime scheme X of size k (the number of all dimensions in X), the computation of the prime and next-prime cuboids over X and $X-\{1\}$, respectively, is based on a previously computed prime cuboid with the smallest scheme that contains X.

– To keep track of the computation, we keep the schemes of all computed prime cuboids in a list called the running scheme and denoted by RS. So, X is appended to RS (S is the first element added to RS). To build the RS, for the currently pointed scheme X in RS, for each dimension $j \in X, j \neq 1$ and $j \neq n$ (n is the last dimension of the fact table), we append $X - \{j\}$ to RS, if $X - \{j\}$ is not yet there.

More precisely, for computing the last-half data cube, we use algorithm $LastHalfCube$.

Algorithm LastHalfCube
Input: A fact table R over scheme S of n dimensions.
Output: The last-half data cube of R and the running scheme RS.
Method:
0) RS = emptyset; // RS: Running Scheme
1) Append S to the RS;
2) Using Table2Tree and R to generate two cuboids over S and S - {1}, respectively;
3) Set cS to the first scheme in RS; // cS: current scheme
4) While cS has more than 2 attributes do
5) For each dimension d in cS, d ≠ 1 and d ≠ n, do
6) Build a subscheme scS by deleting d from cS;
7) If scS is not yet in RS then append scS to RS, let cubo be the cuboid over cS (already computed); If cubo is not yet in memory then load it in memory;
8) Using Tuple2Ptree and cubo to generate two cuboids over scS and scS - {1}, respectively;
9) done;
10) Set cS to the next scheme in RS;
11) done;
12) Return RS;

A. Example of running scheme

Table II shows the simplified execution of LastHalfCube on a table R over $S = \{1, 2, 3, 4, 5\}$: only the prime and the next-prime (NPrime) schemes of the cuboids computed by the algorithm are reported. The prime schemes appended to RS (Running Scheme) during the execution of LastHalfCube are in the columns named Prime/RS of Table II. The first prime schemes are in the first column Prime/RS, the next ones are in the second column Prime/RS, and the final ones are in the third column Prime/RS. The final state of RS is $\{12345, 1345, 1245, 1235, 145, 135, 125, 15\}$. In Table II the schemes marked with x (e.g., 145x) are those already added to RS and are not re-appended to RS.

TABLE II. GENERATION OF THE RUNNING SCHEME OVER $S = \{1, 2, 3, 4, 5\}$.

Prime RS	NPrime	Prime RS	NPrime	Prime RS	NPrime
12345	2345				
1 345	345				
		1 45	45		
				15	5
		1 35	35		
				15x	
12 45	2 45				
		1 45x			
		1 25	25		
				15x	
123 5	23 5				
		1 35x			
		1 25x			

For a fact table R over a scheme S of n dimensions, $S = \{1, 2, ..., n\}$, algorithm LastHalfCube generates RS with 2^{n-2} subschemes. Indeed, we can see that all subschemes appended to RS have 1 as the first attribute and n as the last attribute. So, we can forget 1 and n from all those subschemes. By this, we can consider that the first subscheme added to RS is $2, ..., n-1$. Over $2, ..., n-1$, we have only one subscheme of size $n-2$ (C_{n-2}^{n-2}). In the loop For at point 5 of LastHalfCube, alternatively each attribute from 2 to $n-1$ is deleted to generate a subscheme of size $n-3$. By doing this, we can consider as, in each iteration, we build a subscheme over $n-3$ different attributes selected among $n-2$ attributes. So, we build C_{n-2}^{n-3} subschemes. So on, until the subscheme $\{1, n\}$ (corresponding to the empty scheme after forgetting 1 and n) is added to RS. As

$C_{n-2}^{n-2} + C_{n-2}^{n-3} + ... + C_{n-2}^{0} = 2^{n-2}$

By adding the corresponding next-prime schemes, LastHalfCube generates 2^{n-1} different subschemes. Thus, algorithm LastHalfCube computes 2^{n-1} prime and next-prime cuboids.

B. Data Cube representation

For a fact table R over a dimension scheme $S = \{1, 2, ..., n\}$ with measures $M_1, ..., M_k$, the data cube of R is represented by the three following elements:

1) The running scheme (RS): The list of the prime schemes over S. Each prime scheme has an identifier

number that allows to locate the files corresponding to the prime and next-prime cuboids in the last-half data cube.

2) The last-half data cube of which the cuboids are precomputed and stored on disks using the format to store the BSPT.

3) A relational table over $RowId, M_1, ..., M_k$ that represents the measures associated with each tuple of R.

Clearly, such a representation reduces about 50% space of the entire data cube, as it represents the last-half data cube in the BSPT format.

C. Computing the first-half data cube

Let X be a scheme in the first-half power set of $S = \{1, 2, ..., n\}$. For computing the cuboid over X, we base on the precomputed last-half data cube over S. The computation is processed as follows, where $lti(t)$ denotes the list of tids of a tuple t and $p(t)$ the prefix of t over X.

Let C be the stored cuboid over $X \cup \{n\}$;
Let $t1$ be the 1st tuple of C and $ltids = lti(t1)$;
For each next tuple $t2$ of C do
  If the $p(t2) = p(t1)$ then append $lti(t2)$ to $ltids$,
  Else {
    Write $p(t1) : ltids$ to the cuboid over X;
    $t1 = t2$; $ltids = lti(t1)$;
    }
Done;

V. Experimental results

The present approach to represent and to compute data cubes is implemented in C and experimented on a laptop with 8 GB memory, Intel Core i5-3320 CPU @ 2.60 GHz x 4, 188 Go Disk, running Ubuntu 12.04 LTS. To get some ideas about the efficiency of the present approach, we recall here the experimental results reported in [20] as references, because the work [20] has experimented many existing and well known methods for computing and representing data cube as Partitioned-Cube (PC), Partially-Redundant-Segment-PC (PRS-PC), Partially-Redundant-Tuple-PC (PRT-PC), BottomUpCube (BUC), Bottom-Up-Base-Single-Tuple (BU-BST), and Totally-Redundant-Segment BottomUpCube (TRS-BUC). The experiments in [20] were run on a Pentium 4 (2.8 GHz) PC with 512 MB memory under Windows XP. The results were reported on real and synthetic datasets. In the present work, we limit our attention to only the real datasets: CovType [3] and SEP85L5 [4]. However, by reporting the results of [20], we do not want to really compare the present approach to TRS-BUC or others, as we do not have sufficient conditions to implement and to run these methods on the same system and machine.

CovType is a dataset of forest cover-types. It has ten dimensions and 581,012 tuples. The dimensions and their cardinality are: Horizontal-Distance-To-Fire-Points (5,827), Horizontal-Distance-To-Roadways (5,785), Elevation (1,978), Vertical-Distance-To-Hydrology (700), Horizontal-Distance-To-Hydrology (551), Aspect (361), Hillshade-3pm (255), Hillshade-9am (207), Hillshade-Noon (185), and Slope (67).

TABLE III. EXPERIMENTAL RESULTS REPORTED IN [20]

CovType			
Algorithms	**Storage space**	**Construction time**	**avg QRT**
PC	#12.5 Gb	1900 sec	
PRT-PC	#7.2 Gb	1400 sec	
PRS-PC	#2.2 Gb	1200 sec	3.5 sec
BUC	#12.5 Gb	2900 sec	2 sec
BU-BST	#2.3 Gb	350 sec	
BU-BST+	#1.2 Gb	400 sec	1.3 sec
TRS-BUC	#0.4 Gb	300 sec	0.7 sec
SEP85L			
Algorithms	**Storage space**	**Construction time**	**avg QRT**
PC	#5.1 Gb	1300 sec	
PRT-PC	#3.3 Gb	1150 sec	
PRS-PC	#1.4 Gb	1100 sec	1.9 sec
BUC	#5.1 Gb	1600 sec	1.1 sec
BU-BST	#3.6 Gb	1200 sec	
BU-BST+	#2.1 Gb	1300 sec	0.98 sec
TRS-BUC	#1.2 Gb	1150 sec	0.5 sec

SEP85L is a weather dataset. It has nine dimensions and 1,015,367 tuples. The dimensions and their cardinality are: Station-Id (7,037), Longitude (352), Solar-Altitude (179), Latitude (152), Present-Weather (101), Day (30), Weather-Change-Code (10), Hour (8), and Brightness (2).

For greater efficiency, in the experiments of [20], the dimensions of the datasets are arranged in the decreasing order of the attribute domain cardinality. The same arrangement is done in the our experiments. Moreover, as most algorithms studied in [20] compute condensed cuboids, computing query in data cube needs additional cost. So, the results are reported in two parts: computing the condensed data cube and querying data cube. The former is reported with the construction time and storage space and the latter the average query response time.

Table III presents the experimental results approximately got from the graphs in [20], where "avg QRT" denotes the average query response time and "Construction time" denotes the time to construct the (condensed) data cube. However, [20] did not specify whether the construction time includes the time to read/write data to files.

Table IV reports the results of the present work, where the term "run time" means the time from the start of the program to the time the last-half (or respectively, the first-half) data cube is completely constructed, including the time to read/write input/output files.

As we do not compute the condensed cuboids, but only compute the last-half data cube and use it to represent the data cube, we can consider that the last-half data cube corresponds somehow to the (condensed) representations of data cube in the other approaches, and computing the first-half data cube corresponds to querying data cube. In this view,

 ISBN: 978-1-61208-444-2

TABLE IV. EXPERIMENTAL RESULTS OF THIS WORK

CovType			
	Storage space	**Run time**	**avg QRT**
Last-Half Cube	7 Gb	992 sec	
First-Half Cube	6,2 Gb	439 sec	
Data Cube	13,2 Gb	1431 sec	0.43 sec
SEP85L			
	Storage space	**Run time**	**avg QRT**
Last-Half Cube	3.6 Gb	691 sec	
First-Half Cube	3.3 Gb	243 sec	
Data Cube	6.9 Gb	934 sec	0.47 sec

the average query response time corresponds to the average run time for computing a cuboid based on the precomputed and stored cuboids. That is, the average query response time for SEP85L is 243s/512 = 0.47 second and for CovType 439s/1024 = 0.43 second, because the cuboids in the last-half data cube are precomputed and stored, only querying on the first-half data cube needs computing. Though the compactness of the data cube representation by the present approach is not comparable to the compactness offered by TRS-BUC, it is in the range of other existing methods. It is similar for the run time to build the last-half data cube of CovType. However, the run time to build the entire (not only the last-half) data cube of SEP85L seems to be better than all other existing methods. On the average query response time, it seems that the present approach offers a competitive solution, because querying data cube is a repetitive operation and improving the average query response time is one of the important goals of research on data cube.

VI. CONCLUSION, REMARKS AND FURTHER WORK

Essentially, this work represents a data cube by the last-half data cube: the set of cuboids over schemes that contain the last dimension of the fact table, called prime (or next-prime) cuboids. All other cuboids, those over schemes that do not contain the last dimension, are obtained by a simple projection of the corresponding cuboids in the last-half data cube. The binary search prefix tree (BSPT) structure is used to store cuboids in memory and on disk. Such a structure offers not only a compact representation of cuboids but also an efficient search of tuples. Building a cuboid in the last-half data cube is reduced to building a BSPT. Building a cuboid in the first-half data cube is reduced to copying the prefixes of the BSPT of the corresponding cuboid in the last-half data cube. The BSPT allows efficient group-by operation without previous sort operation on tuples in the fact table or in cuboids. With this advantage, we can think of the possibility of incremental construction of the last-half data cube and the possibility of updating the data cube when inserting new tuples in the fact table.

REFERENCES

[1] S. Agarwal et al., “On the computation of multidimensional aggregates”, Proc. of VLDB’96, pp. 506-521.

[2] V. Harinarayan, A. Rajaraman, and J. Ullman, “Implementing data cubes efficiently”, Proc. of SIGMOD’96, pp. 205-216.

[3] J. A. Blackard, “The forest covertype dataset”, ftp://ftp. ics.uci. edu/pub/machine-learning-databases/covtype, [retrieved: April, 2015].

[4] C. Hahn, S. Warren, and J. London, “Edited synoptic cloud reports from ships and land stations over the globe”, http://cdiac. esd.ornl.gov/cdiac/ndps/ndp026b.html, [retrieved: April, 2015].

[5] S. Chaudhuri and U. Dayal, “An Overview of Data Warehousing and OLAP Technology”, SIGMOD record 1997, 26 (1), pp. 65-74.

[6] J. Gray et al., “Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub-Totals”, Data Mining and Knowledge Discovery 1997 , 1 (1), pp. 29-53.

[7] K. A. Ross and D. Srivastava, “Fast computation of sparse data cubes”, Proc. of VLDB’97, pp. 116-125.

[8] Y. Zhao, P. Deshpande, and J. F. Naughton, “An array-based algorithm for simultaneous multidimensional aggregates”, Proc. of ACM SIGMOD’97, pp. 159-170.

[9] J. S. Vitter, M. Wang, and B. R. Iyer, “Data cube approximation and histograms via wavelets”, Proc. of Int. Conf. on Information and Knowledge Management (CIKM’98), pp. 96-104.

[10] J. Han, J. Pei, G. Dong, and K. Wang, “Efficient Computation of Iceberg Cubes with Complex Measures”, Proc. of ACM SIGMOD’01, pp. 441-448.

[11] L. Lakshmanan, J. Pei, and J. Han, “Quotient cube: How to summarize the semantics of a data cube,” Proc. of VLDB’02, pp. 778-789.

[12] Y. Sismanis, A. Deligiannakis, N. Roussopoulos, and Y. Kotidis, “Dwarf: shrinking the petacube”, Proc. of ACM SIGMOD’02, pp. 464-475.

[13] W. Wang, H. Lu, J. Feng, and J. X. Yu, “Condensed cube: an efficient approach to reducing data cube size”, Proc. of Int. Conf. on Data Engineering 2002, pp. 155-165.

[14] A. Casali, R. Cicchetti, and L. Lakhal, “Extracting semantics from data cubes using cube transversals and closures”, Proc. of Int. Conf. on Knowledge Discovery and Data Mining (KDD’03), pp. 69-78.

[15] L. Lakshmanan, J. Pei, and Y. Zhao, “QC-Trees: An Efficient Summary Structure for Semantic OLAP”, Proc. of ACM SIGMOD’03, pp. 64-75.

[16] D. Xin, J. Han, X. Li, and B. W. Wah, “Star-cubing: computing iceberg cubes by top-down and bottom-up integration”, Proc. of VLDB’03, pp. 476-487.

[17] Y. Feng, D. Agrawal, A. E. Abbadi, and A. Metwally, “Range cube: efficient cube computation by exploiting data correlation”, Proc. of Int. Conf. on Data Engineering 2004, pp. 658-670.

[18] Z. Shao, J. Han, and D. Xin, “Mm-cubing: computing iceberg cubes by factorizing the lattice space”, Proc. of Int. Conf. on Scientific and Statistical Database Management (SSDBM 2004), pp. 213-222.

[19] Y. Sismanis and N. Roussopoulos, “The complexity of fully materialized coalesced cubes”, Proc. of VLDB’04, pp. 540-551.

[20] K. Morfonios and Y. Ioannidis, “Supporting the Data Cube Lifecycle: The Power of ROLAP”, The VLDB Journal, 2008, 17(4), pp. 729-764.

[21] A. Casali, S. Nedjar, R. Cicchetti, L. Lakhal, and N. Novelli, “Lossless Reduction of Datacubes using Partitions”, In Int. Journal of Data Warehousing and Mining (IJDWM), 2009, Vol 5, Issue 1, pp. 18-35.

 ISBN: 978-1-61208-444-2

A System for Managing Transport-network Recovery according to Degree of Network Failure

Toshiaki Suzuki, Hiroyuki Kubo,
Hayato Hoshihara, and Kenichi Sakamoto
Research & Development Group
Hitachi, Ltd.
Kanagawa, Japan
E-mails: {toshiaki.suzuki.cs, hiroyuki.kubo.do,
hayato.hoshihara.dy, and
kenichi.sakamoto.xj}@hitachi.com

Hidenori Inouchi, Takanori Kato,
and Taro Ogawa
Information & Telecommunication Systems Company
Hitachi, Ltd.
Kanagawa, Japan
E-mails: {hidenori.inouchi.dw, takanori.kato.bq, and
taro.ogawa.tg}@hitachi.com

***Abstract*—A system for managing transport-network recovery according to the degree of network failures is proposed. Under this management system, an entire network is separated into multiple areas. A network-management server prepares a three-step recovery procedure to cover the degree of network failures. In the first step of the recovery, an inside-area protection scheme is used to recover current data-transmission paths in each area. In the second step, an end-to-end protection scheme is applied to the current data-transmission paths. In the third step, an operation plane is changed. Each assumed operation plane is composed of recovery configurations for restoring failure paths for assumed area-based network failures. If a small network failure occurs, it is recovered by the inside-area protection and end-to-end protection schemes. If a catastrophic network failure (caused by a disaster) that cannot be recovered by the protection schemes occurs, it is recovered by changing the operation plane in accordance with the damaged areas. A prototype system composed of a network-management server and 96 simulated packet-transport nodes was developed and evaluated. The system could recover a transport network according to the degree of network failures. In case of a small network failure, 1000 data-transmission paths were reconfigured by the inside-area protection scheme and end-to-end protection scheme in about 11 seconds. If a network failure was not recovered by these protection schemes, all tables for 1000 data-transmission paths were reconfigured by changing the operation plane in about 1.1 seconds. As a result, the proposed system could localize and recover a network failure according to the degree of failures.**

Keywords - network management; protection; disaster recovery; packet transport

I. INTRODUCTION

Lately, reflecting the rapid growth of the Internet and cloud systems [1], various services are being provided by way of networks. For example, on-line shopping, net banking, and social-networking services (SNSs) are being provided through networks. In addition, search engines are often used to find unknown information on the Internet. Under these circumstances, networks have become an indispensable service in daily life. If a network is out of service due to failures of network nodes, people's lives and businesses would be considerably damaged. Therefore, if a network fails, it should be recovered promptly [2]. As for failures of a network, small failures (such as a failure of a node or a link) and extensive failures (due to disasters) are envisioned. It is therefore a crucial issue to develop a scalable network-recovery scheme that can cover recovery from both a small network failure and a catastrophic network failure.

As recovery procedures for network failures, two major schemes [3], namely, "protection" and "restoration," are utilized. As for protection, it is possible to recover from a network failure promptly because a backup path to a current path is prepared in advance. However, to recover from a network disaster, plenty of backup paths must be prepared. Protection is therefore useful for small network failures. On the other hand, as for restoration, a recovery path is recalculated after a network failure is detected. It therefore takes much time to recover from network failures if plenty of current paths exist.

In light of the above-described issues, a robust network-management scheme is required. The overall aim of the present study is thus to develop a network-management scheme [4] for monitoring and controlling multi-layer network resources so as to quickly restore network services after a network disaster.

The procedure for recovering from a network failure consists of three steps: the first step is to quickly detect a network failure; the second is to immediately determine how to recover from the failure; the third is to promptly configure recovery paths. In the present study, the second step is focused on. In particular, a scalable network-recovery scheme covering a small failure to a network disaster is proposed. The target network is a transport network, such as the Multi-Protocol Label Switching - Transport Profile (MPLS-TP) network.

The rest of this paper is organized as follows. Section II describes related work. Section III overviews a previously proposed system and a requirement to apply it to small network failures. Section IV proposes a new network-disaster recovery system. Section V presents some results of

 ISBN: 978-1-61208-444-2

evaluations of the system's performance. Section VI concludes the paper.

II. Related Work

Several standardization activities related to reliable networks have been ongoing. The International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) [5] discussed specifications, such as Transport – Multi Protocol Label Switching (T-MPLS) in the first stage of standardization. In the next stage, the ITU-T jointly standardized MPLS-TP specifications with the Internet Engineering Task Force (IETF) [6]. Requests for comments (RFC) on requirements [7] and a framework [8] for MPLS-TP were issued. In addition, RFCs on a framework for MPLS-TP-related operation, administration, and maintenance (OAM) [9] and survivability [10] were issued. The OAM framework is useful for the previously proposed system to detect network failures promptly.

With regards to failure recovery, several schemes have been proposed. One major scheme, called "fast reroute" [11], prepares a back-up path. Another recovery scheme (for multiple failures) prepares multiple backup paths [12], and another one prepares a recovery procedure for multiple modes [13]. In the case of these protection schemes, to recover from catastrophic network failures, a huge volume of physical resources for preparing a large number of standby paths is needed. These schemes are useful for limited network failures, such as failures of a few links or nodes.

In the case of restoration schemes, in contrast to protection schemes, recovery paths are calculated after network failures are detected. Restoration schemes for handling multiple failures [14] and virtual networks [15] have been proposed. A scheme for reducing search ranges by using landmark nodes has also been proposed [16]. It is useful for recovering a seriously damaged network, since all reroutes are calculated from the first. However, if a large number of current paths exist, it might take much time to calculate all recovery paths.

III. Previous System and Requirements

The previously proposed network-recovery system is shown in Figure 1 [4]. As shown in the figure, the target network is composed of a packet transport nodes (PTNs), such as those in an MPLS-TP network. The system only focuses on recovery from multiple area-based network failures on PTN networks. A critical issue is the time consumed in recovering the numerous established paths (shown as solid blue arrows) in packet networks in the case of a network disaster. (Note that "path" means a label-switched path (LSP) [17] and a pseudo wire (PW) [18].) A user is connected to one of the PTNs through a network such as an IP network. A server located in a data center (DC) is also connected to one of the PTNs through an IP network.

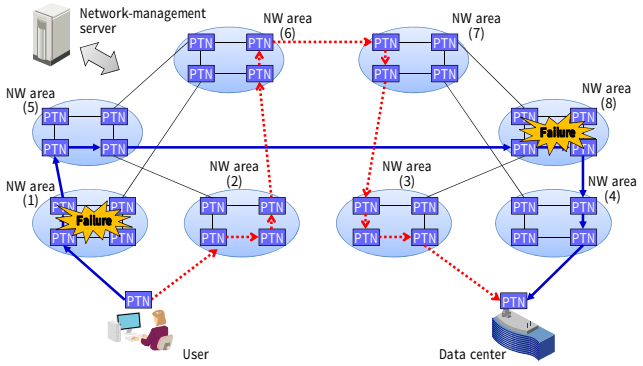


Figure 1. Previously proposed network-recovery system

The previously proposed system could promptly recover from a catastrophic failure of a network by using prepared back-up paths (shown as dotted red arrows). However, it significantly changes network configurations, even if a network failure is small, since network conditions are managed on the basis of divided network areas. It must therefore be enhanced so that it can recover from a catastrophic network failure, as well as a small network failure, by using fewer configurational changes based on the degree of damage due to network failures.

IV. Proposed Transport Network-Recovery Scheme

A. Overview of network management

The structure of the proposed transport network-recovery scheme is similar to the previously proposed scheme (shown in Figure 1). Namely, it is composed of a network-management sever and multiple PTNs. The network-management server centrally manages the whole network. However, recovery procedures are different from those of the previous system.

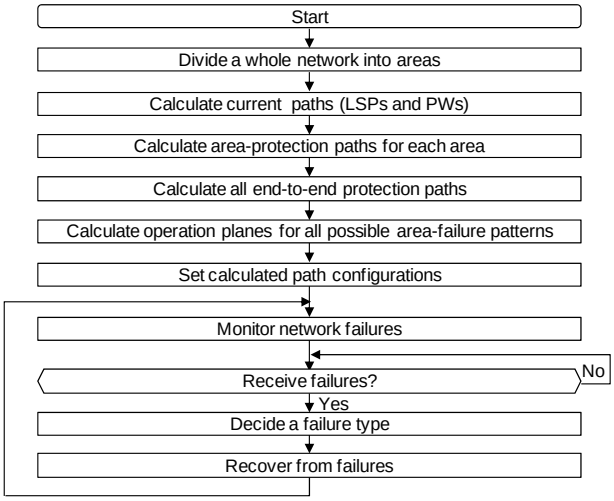


Figure 2. Overview of proposed recovery procedures

 ISBN: 978-1-61208-444-2

A flow chart of the new recovery procedures is shown in Figure 2. First, after starting a network-management function, the network-management server divides the whole network into multiple areas. It calculates current paths (composed of LSPs and PWs) for transmitting data from a sender node to a receiver node according to inputs by a network manager. The network-management server calculates "inside-area protection paths" for each area and "end-to-end protection paths". In addition, it calculates virtual operation planes for all possible area-failure patterns. The protection paths and virtual operation planes are described in detail in later sections. The network-management server sets the entire configuration of the calculated paths and starts to monitor the network for failures. When it detects a network failure, it determines the type of failure, such as an area-based failure. The network-management server then executes proper failure-recovery procedures according to the determined failure pattern.

B. *Path protection for small network failures in each area*

The proposed system should promptly recover a network from a small failure such as a link failure between PTNs or a PTN failure. A scheme called "inside-area protection"—for localizing and swiftly recovering from a small network failure—is overviewed in Figure 3. The network-management server divides an entire PTN network into multiple (e.g., eight) areas, by using a conventional scheme (such as cluster analysis), which it then manages. It configures a current path (shown as solid black arrows in the figure), composed of a LSP and a PW, for transmitting data from a sender to a receiver according to requests by end users. The network-management server configures a backup path for each current path, namely, an inside-area protection path (shown as dotted red arrows), between one edge PTN and another edge PTN in every area. In each area, both edge PTNs exchange OAM packets to check if a disconnection exists between the PTNs. If a disconnection is detected, they send an alert to the network-management server, which keeps the received alert and monitors the degree of failures, namely, numbers of link failures, PTNs, and damaged areas.

In the case shown in Figure 3, a link failure between PTN 14 and PTN 11 is assumed to occur in area (1). PTN 14 and PTN 11 detect the link failure, which is recovered by the inside-area protection. Specifically, a direct data transmission path from PTN 14 to PTN 11 is changed to a backup transmission path through PTN 13 and PTN 12. On the other hand, the path between PTN 14 and PTN 11 is a part of an end-to-end path between provider-edge 1 (PE1) and PE2. The link failure between PTN 14 and PTN 11 is therefore temporarily detected by PE1 and PE2, since both PEs exchange OAM packets. However, both PEs wait for 100 milliseconds to see whether the link failure is recovered by the inside-area protection. Therefore, when the link failure is recovered by the inside-area protection, both PEs do not execute further recovery action.

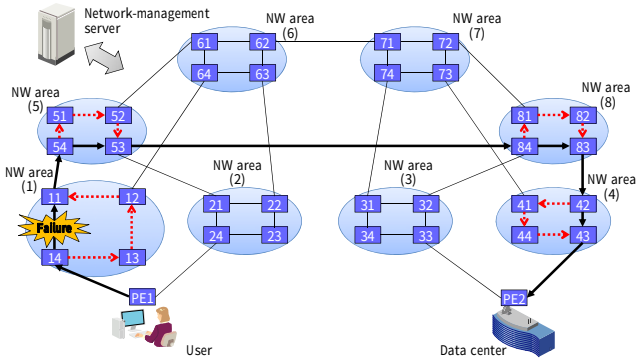


Figure 3. Configuration of path protection in each NW area

C. *End-to-end path protection for small network failures*

The proposed system should be able to immediately recover from a small failure that is not recovered by the above-described protection (such as a link failure between areas). A scheme called "end-to-end protection" to promptly recover from a failure that is not restored by the inside-area protection is overviewed as follows. The network-management server configures a backup path (called an "end-to-end protection path") for each current path between PE1 and PE2. PEs exchange OAM packets to check whether a disconnection exists between them.

Specifically, as shown in Figure 4, the network-management server configures a current path (shown as solid black arrows) between PE1 and PE2 [through areas (1), (5), (8), and (4)] for transmitting data packets between a user and a DC. In addition, the network-management server configures a backup path called an "end-to-end protection path (shown as dotted red arrows)" between PE1 and PE2. The end-to-end protection path is established so as not to travel through the same areas used by the current path as much as possible. In Figure 4, the backup path is configured to transmit data through areas (2), (6), (7), and (3).

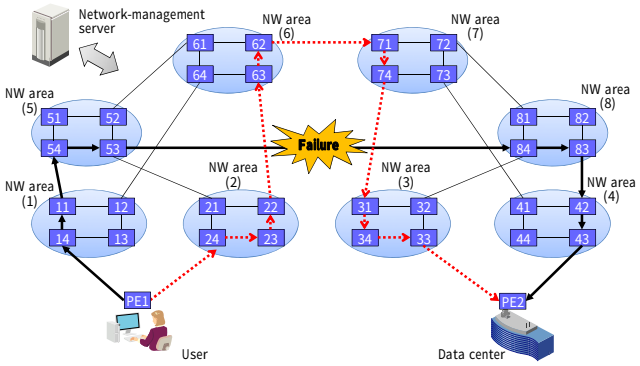


Figure 4. Configuration of path protection for end-to-end transmission

During network operation, the end-to-end protection is executed when the data transmission between PEs is disconnected for a while (for example, 100 milliseconds). In the case of Figure 4, a link failure between area (5) and area

 ISBN: 978-1-61208-444-2

(8) is assumed. This failure is not recovered by the inside-area protection; instead, it is recovered by the end-to-end protection because the failure occurs between areas. Specifically, a data-transmission path is changed from the current path (shown as solid black arrows) to a backup path (shown as dotted red arrows).

This end-to-end protection scheme is a similar to a conventional protection scheme. In the case of a conventional scheme, the protection is immediately executed after one of the PEs detects a disconnection. However, in the case of the proposed end-to-end protection scheme, it is not executed for 100 milliseconds so that whether a failure is recovered by the inside-area protection or not can be checked.

D. Changing operation plane for network-disaster recovery

The proposed system should be able to promptly recover not only failures inside a network area and between network areas but also catastrophic failures. A recovery scheme that changes the operation plane to recover from area-based network failures is overviewed in Figure 5. Before starting network operations, the network-management server prepares multiple backup operation planes for handling possible area-based network failures. Each backup operation plane is composed of recovery configurations for restoring failure paths due to assumed network failures. During network operation, if network failures are not recovered by both the inside-area protection and the end-to-end protection, the failures are recovered by changing an operation plane.

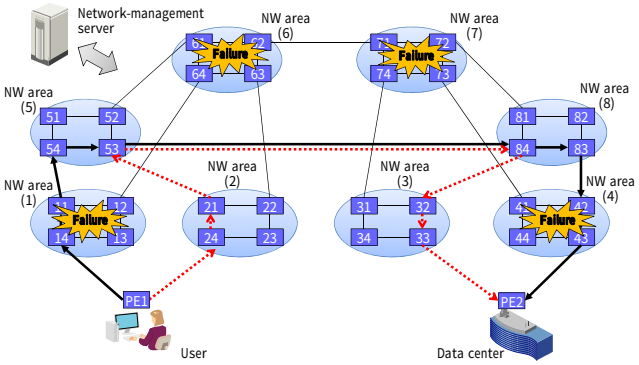


Figure 5. Configuration of operation-plane change for network-disaster recovery

In Figure 5, as an example, the network-management server configures multiple currents paths [through areas (1), (5), (8), and (4)] for transmitting data packets between a user and a DC. The network-management server calculates all recovery paths preliminarily by assuming all possible area-based network failures. The number of possible combinations of areas is 256 (i.e., 2^8), and it includes a pattern by which no area-based network failure occurs. The network-management server therefore prepares 255 backup operation planes. It then assigns a unique recovery identifier (ID) for each backup operation plane, and sends all recovery IDs and recovery configurations to each PTN. Each PTN stores all received recovery IDs and configurations.

An example area-based network-failure recovery procedure is shown in Figure 5. In the figure, area-based network failures are assumed to occur in areas (1), (4), (6), and (7). In this case, PE1 (namely, an edge node of the current path) detects a disconnection between PE1 and PE2. PE1 waits 100 milliseconds to check whether the failures are recovered by the inside-area protection. It also checks the availability of the end-to-end protection path (which is not shown in Figure 5) by using OAM packets. If the failures are not recovered in 100 milliseconds and the end-to-end protection path is not available, PE1 sends an alert to the network-management server to inform it that the end-to-end protection is not available. The network-management server then checks which areas are not available. In this example, by receiving many alerts sent by multiple PTNs, the network-management server determines that area-based network failures occur in areas (1), (4), (6), and (7). It then determines the most suitable backup operation plane to recover by using the determined network-failure information. To change an operation plane, the network-management server sends a recovery ID specifying the most-suitable backup operation plane to related PTNs. Those PTNs change data transmissions according to the received recovery ID. By means of the above-described procedures, the operation plane is changed, and catastrophic network failures are swiftly recovered.

V. PERFORMANCE EVALUATION AND RESULTS

The above-described recovery procedures were evaluated in the case of a small network failure and a catastrophic network failure by using a prototype system. In the evaluation, the times needed to calculate and to configure a table for current data-transmission paths (composed of PWs and LSPs) were evaluated. In addition, the times taken to configure recovery paths in the case of a failure of a PTN or an area-based failure were evaluated.

A. Evaluation system

The system used for evaluating the proposed recovery procedures is shown in Figure 6. As shown in the figure, an entire PTN network is divided into eight areas. Each network area is composed of 12 PTNs, as shown in NW area (7). In each area, the PTNs are connected in a reticular pattern. The network used for the evaluation is an example network composed of about 100 transport network nodes. In addition, a user terminal is connected to PTN-network areas (1) and (2) through PE1, and an application server in the DC is connected to PTN-network areas (3) and (4) through PE2.

Note that the PTN networks (composed of 96 PTNs) are simulated by a physical server. The user terminal and application server are also simulated by that physical server, whose specification is listed in Table I. Another physical

 ISBN: 978-1-61208-444-2

server, which executes the network-management function, has the same specifications as the former server.

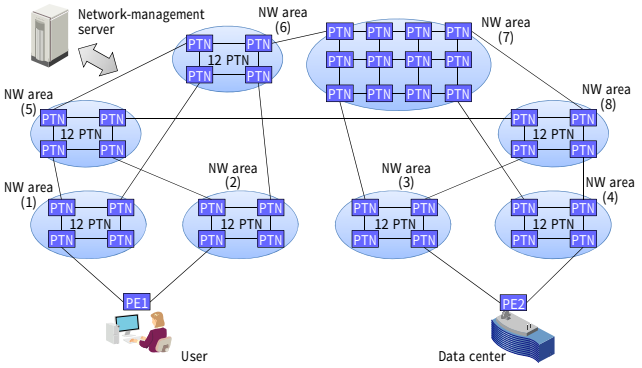


Figure 6. Evaluation system

TABLE I. SPECIFICATIONS OF SERVER

#	Item	Specifications
1	CPU	1.8 GHz, 4 cores
2	Memory	16 Gbytes
3	Storage	600 Gbytes

TABLE II. EVALUATED ITEMS

#	Item	Evaluation specification
1	Current-path calculation time	Time to calculate 100, 500, and 1000 PWs
2	Current-path distribution time	Time to distribute all calculated current paths in case of 100, 500, and 1000 PWs
3	Protection-path calculation time in each area	Time to calculate all protection paths in each area for 100, 500, and 1000 PWs
4	Protection-path calculation time for end-to-end	Time to calculate all protection paths for all end-to-end current paths for 100, 500, and 1000 PWs
5	Recovery-path calculation time for changing operation plane	Time to calculate recovery 100, 500, and 1000 PWs for all possible area-failure patterns
6	Recovery-configuration time	Time to configure all protection paths after detecting path failures
7	Recovery-ID distribution time	Time to distribute a recovery ID after detecting first area failure

B. Evaluation conditions

The times taken to calculate multiple PWs between PE1 and PE2 were evaluated. Each PW was included in a LSP. If a transmission path of a PW differed from the path of an already setup LSP, a new LSP was setup, and the PW was included in the new LSP. The evaluations were executed according to the patterns listed in Table II. Specifically, the times taken to calculate current paths, to distribute their configuration to all PTNs, and to calculate the inside-area protection paths and end-to-end protection paths were evaluated by changing the number of PWs (namely, 100, 500, and 1000). In addition, the times taken to calculate recovery paths for the operation-plane change, to configure protection paths, and to distribute the recovery ID were evaluated.

C. Evaluation results

1) Current-path calculation time

The times taken to calculate current PWs between PE1 and PE2 requested by a user are plotted in Figure 7. A scalability evaluation was executed by changing setup PWs. As shown in the figure, the times taken to calculate 100 current PWs, 500 current PWs, and 1000 current PWs were respectively about 142, 546, and 1094 milliseconds.

2) Distribution time for configuring current paths

The times taken to distribute all configurations of calculated current paths to all PTNs are plotted in Figure 8. As shown in the figure, the times taken to distribute all configurations of the 100 current PWs, 500 current PWs, and 1000 current PWs are respectively about 22, 373, and 767 milliseconds. The distribution times are a little shorter than the calculation ones.

3) Protection-path calculation time for all current paths in each area

The times taken to calculate protection paths corresponding to all current PWs in each area are plotted in Figure 9. As shown in the figure, the times required for calculating all the inside-area protection paths for 100 current PWs, 500 current PWs, and 1000 current PWs are respectively about 405, 778, and 1510 milliseconds.

4) Protection-path calculation time for all end-to-end current paths

The times taken to calculate end-to-end protection paths to all current PWs are plotted in Figure 10. As shown in the figure, the times taken to calculate all the end-to-end protection paths for 100 current PWs, 500 current PWs, and 1000 current PWs are respectively about 216, 936, and 1724 milliseconds.

5) Recovery-path calculation time for operation-plane change

The times taken to calculate all recovery PWs for 255 possible area-based network-failure patterns are plotted in Figure 11. As shown in the figure, the times taken to calculate all recovery PWs for 255 area-based network-failure patterns and 100 current PWs, 500 current PWs, and 1000 current PWs are respectively about 11.8, 42.2, and 79.9 seconds.

6) Recovery-configuration time required by both protection schemes for each area and end-to-end path

The times taken to set recovery configuration by the inside-area protection and end-to-end protection schemes after detecting a disconnection of a path are plotted in Figure 12. Specifically, recovery configuration time was evaluated by intentionally invoking a node failure in area (5). In the evaluation, if a disconnected path is not recovered for 100 milliseconds by the inside-area protection, it is automatically recovered by the end-to-end protection. Actually, disconnected paths were recovered by the end-to-end protection. As shown in the figure, the times to set recovery configurations for 100 current PWs, 500 current PWs, and 1000 current PWs by both protections are respectively about

 ISBN: 978-1-61208-444-2

1.0, 4.6, and 10.3 seconds. As a result, 1000 PWs were recovered in about 11 seconds in case of a node failure.

7) Recovery-ID distribution time for changing operation plane

The times taken to distribute the recovery ID to related PTNs and recover after the last area-based network failure is detected in the case of 100 current PWs, 500 current PWs, and 1000 current PWs are plotted in Figure 13. Three area-based network-failure patterns, namely, failures of network areas (1) and (6), failures of network areas (1), (6), and (4), and failures of network areas (1), (6), (4), and (7), were evaluated. As shown in the figure, in the case of 100 current PWs, the times taken to recover from the first failure for the three area-based network-failure patterns are respectively about 165, 160, and 132 milliseconds. In the case of 500 current PWs, the times taken to recover from the first failure for the three area-based network-failure patterns are respectively about 546, 540, and 526 milliseconds. In the case of 1000 current PWs, the times taken to recover from the first failure for the three area-based network-failure patterns are respectively about 1083, 1061, and 1063 milliseconds. As shown in the figure, the times taken to recover are almost independent of the number of area-based network failures, although they are dependent on the number of setup PWs. As a result, tables that are used for data transmission on 1000 PWs are reconfigured by changing an operation plane in about 1.1 seconds.

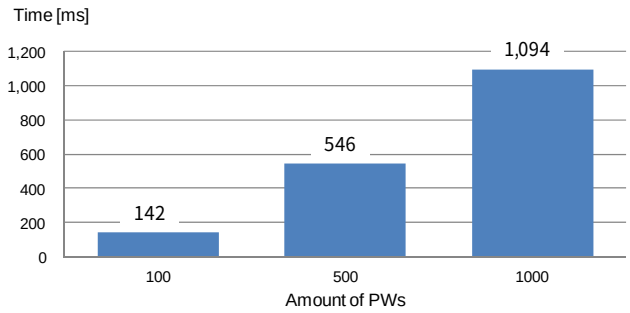


Figure 7. Calculation time for current paths

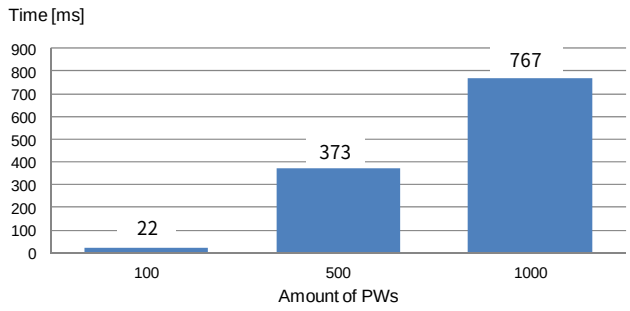


Figure 8. Distribution time for current-path configuration

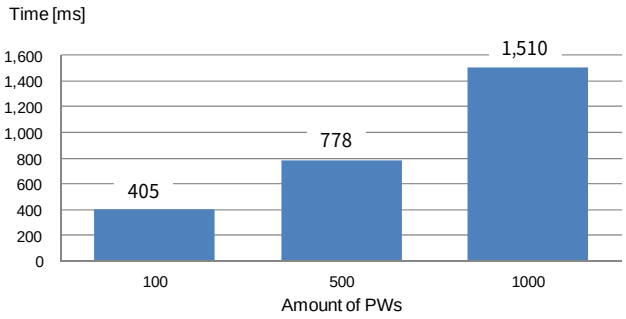


Figure 9. Calculation time for protection paths in each area

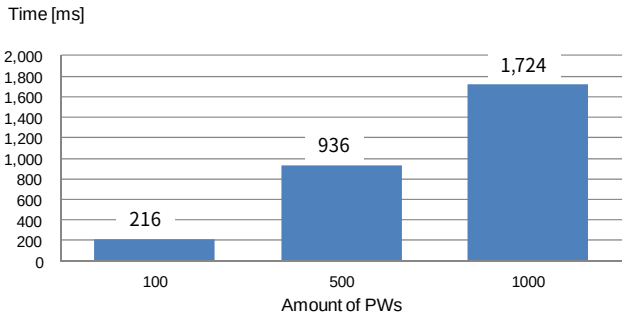


Figure 10. Calculation time for end-to-end protection paths

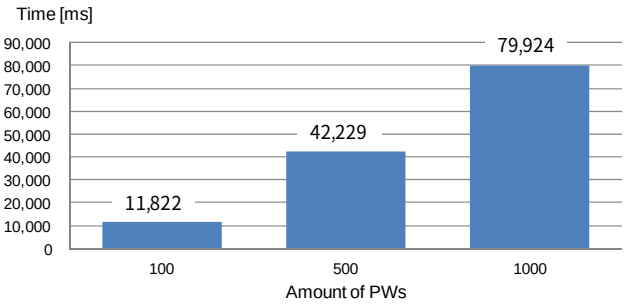


Figure 11. Calculation time for changing operation-plane

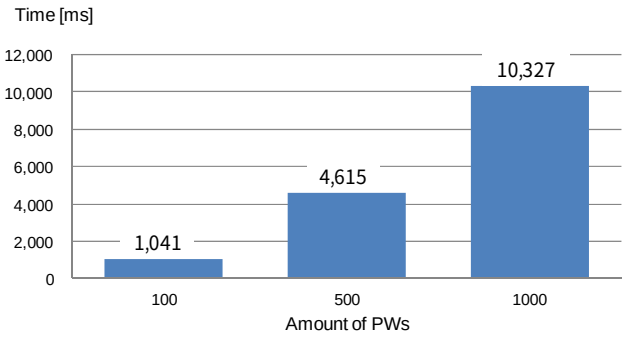


Figure 12. Recovery-configuration time in the cases of using protection paths in NW areas and end-to-end protection paths

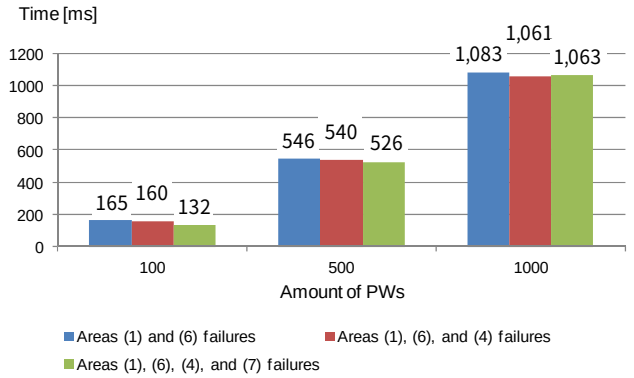


Figure 13. Recovery-configuration time in the case of changing operation-plane

D. Discussion

The times taken to recover from failures, such as disconnection of paths, are plotted in Figure 12. In this evaluation, a PTN failure was intentionally invoked in area (5). As a recovery procedure, inside-area protection is expected to be appropriate, since the failure was invoked in area (5). However, end-to-end protection was also used. As for the proposed system, updated PWs and LSPs are always stored after changing data-transmission paths by recovery procedures, such as inside-area protection. In addition, if a failure that is not recovered by the inside-area protection for 100 milliseconds occurs, it is recovered by the end-to-end protection. Over 100 milliseconds were taken to store the PWs and LSPs updated by the inside-area protection; therefore, the PTN failure in area (5) was recovered by both the inside-area protection and the end-to-end protection. The PTN failure was recovered in 11 seconds, which is a little longer, since recovery paths are configured one by one. In future work, the times taken to manage multiple updated PWs and LSPs should be shortened.

The times taken to distribute the recovery ID and store updated PWs and LSPs are shown in Figure 13. As shown in the figure, in the case of 96 PTNs, tables for data transmission on 1000 current PWs were reconfigured in about 1.1 seconds. The times taken to recover from the area-based network failure depend on the number of current PWs. The times for recovery are short because the times for setting up real PWs are not included; instead, the times for configuring tables to transmit data are included. In addition, all tables for data transmission are changed at once by switching the operation plane. According to the results of this evaluation, the proposed system can provide a faster recovery procedure than recalculating and transmitting recovery paths to PTNs (since it omits the recalculation process). In addition, this advantage is enhanced as the number of configured current PWs increases.

In this study, a transport-network-recovery management system, which can recover from both a small network failure and a major network disaster, was proposed and evaluated. Specifically, the three-step recovery procedure was proposed. As described above, updated data-transmission paths of PWs and LSPs are always stored in a database. Therefore, transmission paths composed of PWs and LSPs updated by changing the operation plane are also stored in the database. As a result, the times taken to recover from the network disaster by changing the operation plane depend on the number of PWs. However, as shown in Figures 12 and 13, the proposed system could recover from both a small network failure and a catastrophic network failure (which is not covered by conventional network-recovery schemes).

VI. CONCLUTION

A system for managing transport-network recovery based on the degree of network failures is proposed. Under this management scheme, an entire network is separated into multiple areas. A network-management server executes a three-step recovery procedure. In the first step, an inside-area protection scheme is applied to the current data-transmission path in each area. In the second step, an end-to-end protection scheme is applied to the current data-transmission path. In the third step, the operation plane is changed. Each assumed operation plane is composed of recovery configurations for restoring failure paths by assuming area-based network failures. If a small network failure occurs, it is recovered by the inside-area protection and end-to-end protection schemes. If a catastrophic network failure (due to a disaster) that is not recovered by the protection schemes occurs, it is recovered by changing the operation plane according to damaged areas.

A prototype system composed of a network-management server and 96 simulated packet-transport nodes was developed and evaluated. In the case of a small network failure, 1000 data-transmission paths were reconfigured by the inside-area protection and end-to-end protection schemes in about 11 seconds. If a network failure was not recovered by the protection schemes, all tables for data transmission were reconfigured to recover from the failure by changing the operation plane in about 1.1 seconds. As a result, the proposed system could localize and recover a network failure according to the degree of network failures.

Although the protection scheme could recover 1000 PWs from a small network failure, it took the network-management server about 11 seconds to configure and store changed-data transmission routes. If numerous current paths exist, it will take much time to assess changed paths. Accordingly, the protection scheme will be further developed so that it can promptly manage a large number of recovered paths.

ACKNOWLEDGMENTS

Part of this research was done within research project O3 (Open, Organic, Optima) and programs, "Research and Development on Virtualized Network Integration

Technology," "Research and Development on Management Platform Technologies for Highly Reliable Cloud Services," and "Research and Development on Signaling Technologies of Network Configuration for Sustainable Environment" supported by MIC (The Japanese Ministry of Internal Affairs and Communications).

REFERENCES

[1] Cisco Global Cloud Index: Forecast and Methodology, 2013–2018, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf [retrieved: Sept. 2015]

[2] A. Bianco, J. Finochietto, L. Giraudo, M. Modesti, and F. Neri, “Network Planning for Disaster Recovery,” 16th IEEE Workshop on Local and Metropolitan Area Networks, LAMAN 2008, Sept. 2008, pp. 43-48.

[3] E. Mannie and D. Papadimitriou, “Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS),” RFC 4427, Mar. 2006.

[4] T. Suzuki et al., “A Network-disaster Recovery System using Area-based Nework Management,” The Third International Conference on Communications, Computation, Networks and Technologies (INNOV 2014), Oct. 2014, pp. 8-15.

[5] International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) http://www.itu.int/en/ITU-T/Pages/default.aspx [retrieved: Sept. 2015].

[6] The Internet Engineering Task Force (IETF), http://www.ietf.org/ [retrieved: Sept. 2015].

[7] B. Niven-Jenkins, D. Brungard, M. Betts, N. Sprecher, and S. Ueno, “Requirements of an MPLS transport profile,” RFC 5654, Sept. 2009.

[8] M. Bocci, S. Bryant, D. Frost, L. Levrau, and L. Berger, “A Framework for MPLS in transport networks,” RFC 5921, July 2010.

[9] T. Busi and D. Allan, “Operations, administration, and maintenance framework for MPLS-based transport neworks,” RFC 6371, Sept. 2011.

[10] N. Sprecher and A. Farrel, “MPLS transport profile (MPLS-TP) survivability framework,” RFC 6372, Sept. 2011.

[11] P. Pan, G. Swallow, and A. Atlas, “ Fast reroute extensions to RSVP-TE for LSP tunnels,” RFC 4090, May 2005.

[12] J. Zhang, J. Zhou, J. Ren, and B. Wang, “A LDP fast protection switching scheme for concurrent multiple failures in MPLS network,” 2009 MINES '09. International Conference on Multimedia Information Networking and Security, Nov. 2009, pp. 259-262.

[13] Z. Jia and G. Yunfei, “Multiple mode protection switching failure recovery mechanism under MPLS network,” 2010 Second International Conference on Modeling, Simulation and Visualization Methods (WMSVM), May 2010, pp. 289-292.

[14] M. Lucci, A. Valenti, F. Matera, and D. Del Buono, “Investigation on fast MPLS restoration technique for a GbE wide area transport network: A disaster recovery case,” 12th International Conference on Transparent Optical Networks (ICTON), Tu.C3.4, June 2010, pp. 1-4.

[15] T. S. Pham. J. Lattmann, J. Lutton, L. Valeyre, J. Carlier, and D. Nace, “A restoration scheme for virtual networks using switches,” 2012 4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Oct. 2012, pp. 800-805.

[16] X. Wang, X. Jiang, C. Nguyen, X. Zhang, and S. Lu, “Fast connection recovery against region failures with landmark-based source routing,” 2013 9th International Conference on the Design of Reliable Communication Networks (DRCN), Mar. 2013, pp. 11-19.

[17] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol lable switching architecture,” RFC 3031, Jan. 2001.

[18] S. Bryant and P. Pate, “Pseudo wire emulation edge-to-edge (PWE3) architecture”, RFC 3985, Mar. 2005.

 ISBN: 978-1-61208-444-2

A Novel Time-Domain Frequency Offset Estimation Algorithm for LTE Uplink

Mirette Sadek, Khaled Ismail, Mahmoud Samy, and Sameh Sowelam

Axxcelera Egypt
Cairo, Egypt
(Mirette Sadek is also affiliated with Ain Shams University)
Email: {mirette.sadek, khaled.ismail, mahmoud.samy, sameh.sowelam} @axxceleraegypt.com

***Abstract*—Frequency offset (FO) estimation and compensation is critical to the performance of orthogonal frequency division multiple access (OFDMA) systems. In uplink single carrier SC-OFDMA systems, such as long term evolution (LTE), traditional correlation-based time-domain FO estimation techniques are not valid due to the presence of multi-users. Thus, frequency-domain techniques are mostly used. In case of frequency hopping (FH) in uplink LTE, correlation-based frequency-domain techniques cannot be used and other alternative techniques give modest results. In this paper, we propose a novel time-domain FO estimation technique and show that it results in superior performance in case of FH.**

***Index Terms*—LTE; Uplink; Frequency Offset; hopping.**

I. Introduction

In recent years, orthogonal frequency division multiple access (OFDMA) has been widely adapted as a multiple access technique in wideband wireless communication systems including long term evolution (LTE) [1]. OFDMA offers several advantages over both time division multiple access (TDMA) and frequency division multiple access (FDMA) techniques, namely simpler equalization and better spectral efficiency compared to the former and the latter, respectively.

OFDMA depends on the orthogonality between subcarriers. This means that OFDMA systems are particularly sensitive to frequency offset (FO) since it ruins the subcarriers orthogonality. Thus, FO estimation and compensation in OFDMA systems is critical for acceptable system performance. The main sources of FO are the lack of frequency synchronization between the transmitter and receiver on one hand and the Doppler shift introduced by the receiver mobility on the other.

Frequency offset results in a linear phase superimposed on the time-domain signal. In downlink systems, for instance in the physical downlink shared channel (PDSCH) of LTE, the user equipment (UE) estimates the FO through the phase difference between the cyclic prefix (CP) samples and the corresponding OFDM symbo l samples. This is not feasible in the uplink, for instance in the LTE physical uplink shared channel (PUSCH). In the uplink, multiple UEs transmit at the same time but are disjoint in frequency and the enhanced node B (eNB) receiver separates UE signals in the frequency domain. Thus, typical time-domain techniques cannot be used to estimate the FO. Instead, frequency-domain FO estimation techniques have been adopted for the LTE uplink case. Most algorithms for frequency offset estimation in OFDM systems are based on the correlation based method in [2] [3]. The algorithm proposed in [4] is an example of a frequency-domain correlation technique. This algorithm estimates the average phase difference between the slot 0 and slot 1 estimated channels through the use of reference signals (RS) of PUSCH. The aforementioned algorithm results in relatively accurate FO estimation results. However, it cannot be implemented in case of frequency hopping in LTE since in that case, the slot 0 and slot 1 channels are not aligned in frequency and thus the average phase difference between pairs of corresponding subcarriers cannot be attributed to the FO alone. A *frequency bins* algorithm has been introduced in [5] for the case of FH. However, simulation results show that the FO estimation using the frequency bins algorithm is sensitive to noise and does not provide acceptable estimation accuracy in low signal to noise ratio (SNR) cases as shown in the numerical results section.

In this paper, we propose a novel time-domain FO estimation algorithm that performs well for both the hopping and non-hopping cases. Estimation accuracy is significantly improved compared to the algorithm in [5].

This paper is organized as follows: Section II presents the system model. Section III introduces the proposed time-domain FO estimation algorithm. In Section IV, the numerical results are presented. Section V concludes the paper.

II. System Model

The LTE uplink uses single carrier orthogonal frequency multiple access (SC-OFDMA). This means that at the UE transmitter, the transmit data goes through

 ISBN: 978-1-61208-444-2

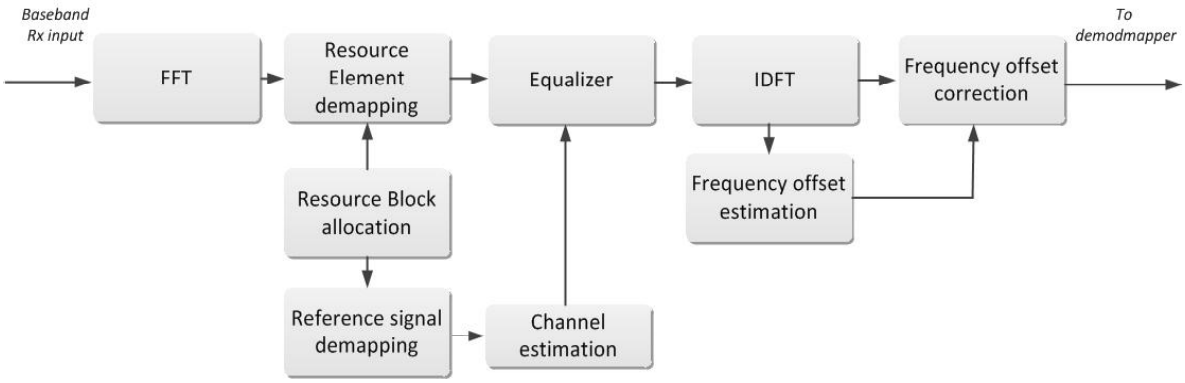


Figure 1. The block diagram of the LTE receiver.

a discrete Fourier transform (DFT) process before the OFDM symbol generation. A block diagram of an LTE receiver is shown in Figure 1. The received baseband signal is first transformed into the frequency domain where the RS is separated from the data. The RS is used to estimate the channel, which is fed to the equalizer. The equalized data is transformed into time-domain quadrature amplitude modulation (QAM) complex time samples. The proposed algorithm performs FO estimation using the inverse DFT (IDFT) output (of size N_{FFT}) time-domain QAM samples.

FO results in an excess linear phase across the time-domain QAM samples as shown in Figure 2. For OFDM symbol i, and time sample n, the excess phase due to an FO of Δf Hz can be modeled as an average phase $\bar{\theta}_i$ in addition to an additional zero-mean linear phase $\Delta\theta(n)$. The FO time sample $\tilde{s}_i(n)$ is given by

$$\tilde{s}_i(n) = s_i(n)e^{j\left(\bar{\theta}_i + \Delta\theta(n)\right)}, \quad (1)$$

where $s_i(n)$ is the time sample without FO, $\bar{\theta}_i = 2\pi\Delta f \Delta T_i$ is the mean excess phase of OFDM symbol i separated by ΔT_i seconds from a reference OFDM symbol, and $\Delta\theta(n) = 2\pi\Delta f n T_s$ is the additional zero-mean linear phase of sample n, where $-N_{FFT}/2 \leq n < N_{FFT}/2$, and T_s is the sample time. Figure 3 illustrates the time-domain constellation diagram of a 16-QAM signal offset by $\Delta f = 35$ Hz containing time samples from one subframe (12 OFDM symbols carrying data). The constellation is rotated by an average angle $\bar{\theta}_i$. Moreover, each group of samples carrying the same QAM symbol is further spread by different angles $\Delta\theta(n)$ depending on their respective sample numbers in their own OFDM symbols. In Section III, we show how the angular spread of the time-domain constellation samples is used to estimate and correct frequency offset.

III. Proposed FO Estimation Algorithm

As explained in Section II, the FO value results in a unique value of the angular rotation $\bar{\theta}_i$ and angular spread

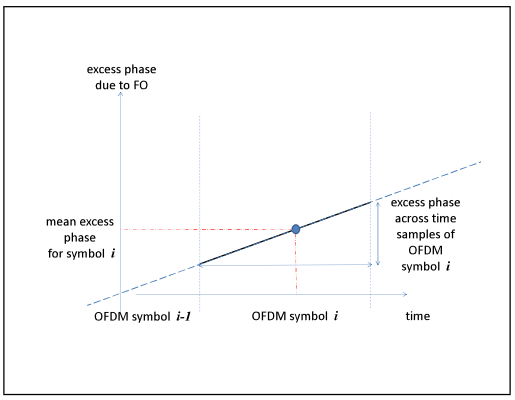


Figure 2. Excess linear phase caused by FO across OFDM symbols.

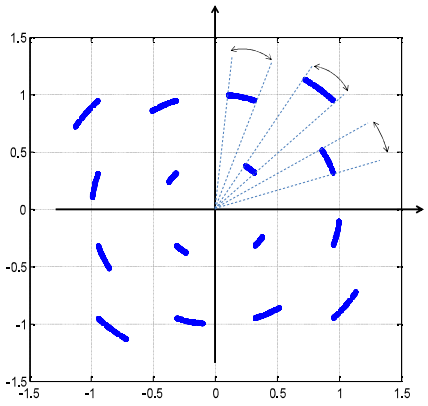


Figure 3. Rotation of constellation point with an angular spread corresponding to the linear phase spread.

$\Delta\theta(n)$ for each OFDM symbol. Conversely, a frequency offset OFDM symbol compensated using the correct value of the FO should have minimal angular rotation and spread. In case of perfect channel equalization and absence of additive noise, the residual angular spread after compensation should ideally be zero.

In this paper, we propose an algorithm for FO estima-

tion that processes the time-domain QAM constellation points in uplink LTE. The idea is to first specify a number of candidate values for FO spanning a reasonable range $-\Delta f_{max} \leq \Delta f \leq \Delta f_{max}$, where Δf_{max} is the maximum expected frequency offset determined by the deployment scenario. Maximum limits on the frequency error between the UE and eNB are mandated by the LTE standard [6]. Also, there are practical limits on the Doppler shift dictated by maximum expected speeds of users in the LTE system [7]. Both of these limits together determine the value of Δf_{max}. The number of candidate values spanning the specified range is a tradeoff between FO estimation accuracy and computational complexity of the algorithm. Simulation results show that a reasonable resolution (difference between two consecutive points in the range) is 100 Hz.

An exhaustive search is performed on all the candidate FO values to decide on the most accurate FO estimate. Each candidate value is used for FO compensation of the time-domain constellation. After FO compensation, the resulting signal is hard demapped to obtain different groups of constellation samples. Each group is centered around a rotated constellation point.

We propose to use the residual angular spread of each group around its center point as a metric to determine the best FO estimate, where the best estimate results in the minimum residual angular spread. A practical measure of the angular spread is the variance defined as $E[|x-\mu_x|^2]$ where x represents the constellation points in a group and the mean μ_x is the group center. Note that even if one of the candidate FO values falls exactly on the actual FO, the residual variance will not vanish since the source of the residual variance is the additive noise variance (independent of the compensation frequency value) in addition to the residual angular spread.

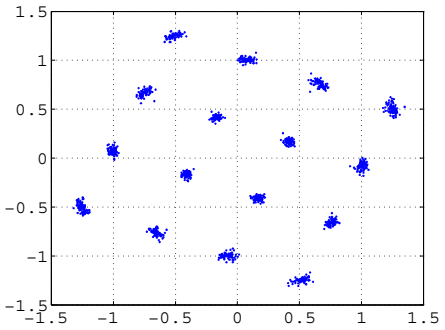


Figure 4. A noisy 16-QAM time-domain constellation diagram with an FO of 300 Hz.

Figure 4 shows the constellation diagram of a 16-QAM signal with frequency offset 300 Hz. Figure 5 shows the diagram after perfect compensation, i.e., 300 Hz is used to compensate the offset. Figure 6 is the diagram after compensation using only 100 Hz. It can be seen from both figures that when using the correct value for frequency offset compensation, the constellation rotation is eliminated and the scattering of constellation points is reduced. On the other hand, when using an incorrect value for compensation, there is a residual constellation rotation, as well as relatively large scattering of the constellation point.

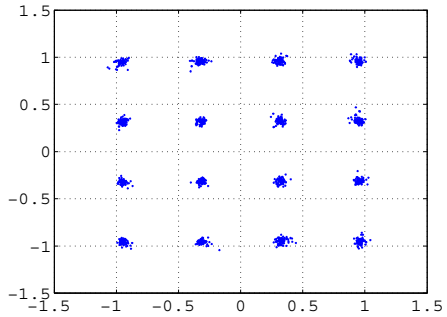


Figure 5. A noisy 16-QAM time-domain constellation diagram with an FO of 300 Hz after FO compensation of 300 Hz.

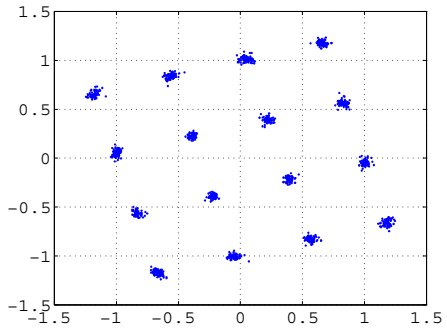


Figure 6. A noisy 16-QAM time-domain constellation diagram with an FO of 300 Hz after FO compensation of 100 Hz.

In our proposed algorithm, we measure the level of scattering through the variance of the constellaiton points around their respective centers and use that as a metric to decide on the correct FO value. The algorithm can be summarized in the following steps:

1) Set $FO = -\Delta f_{max}$ Hz.
2) Use FO to compensate the IDFT output.
3) Perform hard de-mapping of the compensated signal.
4) Group the soft values of the compensated signal based on their hard de-mapped values.
5) Calculate the variance of each group.
6) Increment FO by f_{step} Hz. If $FO > \Delta f_{max}$ Hz, go to step 8. Otherwise, proceed to step 7.
7) Go to step 2.

 ISBN: 978-1-61208-444-2

8) Compare all variances calculated in step 5.
9) The estimated FO corresponds to the minimum variance in step 8.

IV. Numerical Results

In this section, we present the numerical results for our proposed FO estimation algorithm.

Figure 7 is a plot of the mean estimated FO in Hz vs SNR for a 64-QAM modulation with code rate 5/6 and an FO of 300 Hz. Throughout our simulations, $\Delta f_{max} = 400$ Hz with a range resolution of 100 Hz. Note that system performance is not sensitive to small FO estimation errors within ±50 Hz.

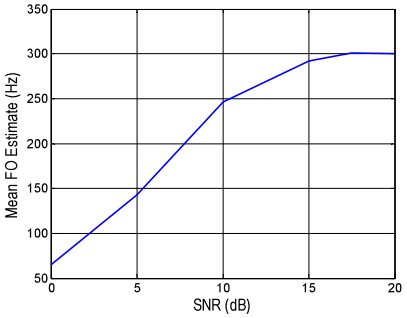


Figure 7. Mean estimated FO vs. SNR: 1 UE with 1 RBs, 64 QAM 5/6, one-tap channel with frequency offset of 300 Hz.

Figure 8 shows the block error rate (BLER) vs. SNR performance curves for both our proposed time-domain algorithm and the frequency bins algorithm proposed in [5]. These curves are for a QPSK modulation with code rate 1/3. The FO of 300 Hz in this simulation corresponds to a mobile user moving at a speed of 85 km/hr. The simulation results show that our proposed time-domain algorithm outperforms the frequency bins algorithm. The gap in performance is significant for the high SNR range.

Figure 9 shows the block error rate (BLER) vs. SNR performance curves for same aforementioned algorithms. These curves are for a 64-QAM modulation with code rate 5/6. Again, the FO of 300 Hz in this simulation corresponds to a mobile user moving at a speed of 85 km/hr. The simulation results show that our proposed time-domain algorithm outperforms the frequency bins algorithm with a significant gap for the high SNR range.

V. Conclusion

In this paper, we propose a novel time-domain frequency offset estimation and compensation algorithm that can be used in case of frequency hopping uplink LTE, as well as non-hopping case. The numerical results

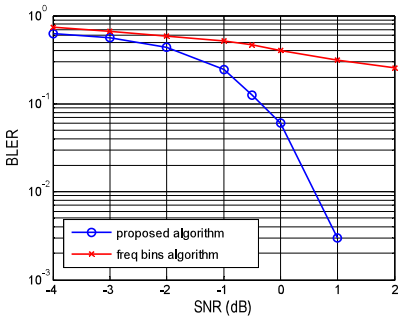


Figure 8. BLER performance of proposed and frequency-bins algorithms:1 UE with 1 RBs, QPSK 1/3, one-tap channel with frequency offset of 300 Hz.

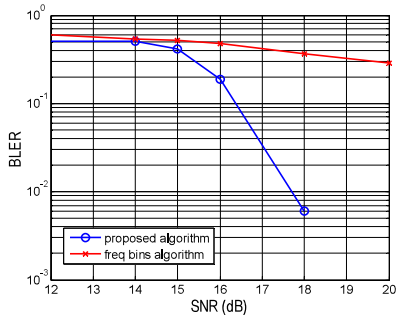


Figure 9. BLER performance of proposed and frequency-bins algorithms:1 UE with 1 RBs, 64 QAM 5/6, one-tap channel with frequency offset of 300 Hz.

show the superior performance of the proposed algorithm compared to the only published algorithm that can be used in the frequency hopping case to the best of the author's knowledge.

References

[1] "E-UTRA physical channels and modulation," 3GPP TS 36.211, 2012.
[2] C. J. K. M. Morelli and M.-O. Pun, "Synchronization techniques for orthogonal division multiple access (OFDMA): A tutorial review," in Proc. IEEE, vol. 95, no. 7, July 2007, pp. 1394–1425.
[3] P. H. Moose, "A technique for orthogonal division multiplexing frequency offset correction," Communications, IEEE Transactions on, vol. 42, no. 10, October 1994.
[4] Z. Zhaohua, "Study on frequency offset estimation and compensation in LTE uplink system," in Proceedings of the 2nd IEEE International Conference on Information Management and Engineering (ICIME),, April 2010, pp. 211 –215.
[5] P. Bertrand, "Frequency Offset Estimation in 3G LTE," in Proceedings of the IEEE 71st Vehicular Technology Conference (VTC),, May 2010, pp. 1 –5.
[6] "User equipment (UE) radio transmission and reception," 3GPP TS 36.101, 2011.
[7] I. T. Stefania Sesia and M. Baker, LTE The UMTS Long Term Evolution: From Theory to Practice. Wiley,New York, 2009.

 ISBN: 978-1-61208-444-2

Endorsement Deduction and Ranking in Social Networks

Hebert Pérez-Rosés
and Francesc Sebé

Department of Mathematics
University of Lleida,
Lleida, Spain
Email: [hebert.perez,fsebe]@matematica.udl.cat

Josep Ma. Ribó †

Department of Computer Science
University of Lleida,
Lleida, Spain

***Abstract*—Some social networks, such as LINKEDIN and RESEARCHGATE, allow user endorsements for specific skills. From the number and quality of the endorsements received, an authority score can be assigned to each profile, with respect to a specific skill. In this paper, we propose an authority score computation method that takes into account the relations existing among different skills. Our method is based on enriching the information contained in the digraph of endorsements corresponding to a specific skill, and then applying a ranking method admitting weighted digraphs, such as PAGERANK. We describe the method, and test it on a synthetic network of 1493 nodes, fitted with endorsements.**

***Keywords–Social Networks; Expertise retrieval;* LINKEDIN*;* RESEARCHGATE*;* PAGERANK**

I. INTRODUCTION

LINKEDIN and RESEARCHGATE are two prominent examples of professional social networks implementing the *endorsement* feature. A user can declare certain skills, and get endorsed for these skills by other users. From the endorsements shown in an applicant's profile, a potential employer can assess the applicant's skills with a higher level of confidence than say, by just looking at his/her CV.

The two endorsement systems are very similar: For each particular skill, the endorsements make up the arcs of a directed graph [4], whose vertices are the members' profiles. In principle, these endorsement digraphs could be used to compute an authority ranking of the members with respect to each particular skill. This authority ranking may provide a better assessment of a person's profile, and it could also be the core element of an eventual tool for finding people who are proficient in a certain skill, very much like a web search engine [6]. Expertise retrieval is the area of Computer Science that deals with those issues [1][5].

Now, people usually have more than one skill, with some of those skills being related. For example, the skill 'Java' is a particular case of the skill 'Programming', which in turn is strongly related with the skill 'Algorithms'. It may well happen that a person is not endorsed for the skill 'Programming', but he/she is endorsed for the skills 'Java' and 'Algorithms'. From those endorsements it can be deduced with a fair degree of confidence that the person also possesses the skill 'Programming'. In other words, a person's ranking with respect to the skills 'Java' and 'Algorithms' affects his/her ranking with respect to the skill 'Programming'.

If the members of a social network were consistent while endorsing their peers, this 'endorsement with deduction' would not add anything to simple (i.e., ordinary) endorsement. In this ideal world, if Alice endorses Bob for the skill 'Java', she would be careful to endorse him for the skill 'Programming' as well. In practice, however,

1) People are not consistent, for consistency would require a great effort. In an analysis of a small LINKEDIN community we have detected several inconsistencies. For example, several users have been endorsed for 'C++' but not for 'Programming'.
2) People are not systematic. That is, people do not usually go over all their contacts systematically to endorse, for each contact and alleged skill, all those contacts which, according to their opinion, deserve such endorsement.
3) Skills lack standardization. In most of these social networks, a set of standard, allowed skills has not been defined. As a result, many related skills (in many cases, almost synonyms) may come up in different profiles of the social network.

Endorsement with deduction may help address those problems, and thus provide a better assessment of a person's skills. More precisely, we propose an algorithm that enriches the digraph of endorsements associated to a particular skill with new weighted arcs, taking into account the correlations between that 'target' skill and the other ones.

A. Contributions of this paper

This paper focuses on professional social networks allowing user endorsements for particular skills, such as LINKEDIN and RESEARCHGATE. Our main contributions can be summarized as follows:

1) We introduce *endorsement deduction*: an algorithm to enrich/enhance the information contained in the digraph of endorsements corresponding to a specific skill ('target' skill or 'main' skill) in a social network. This algorithm adds new weighted arcs (corresponding to other skills) to the digraph of endorsements, according to the correlation of the other skills with the 'main' skill. We assume the

 ISBN: 978-1-61208-444-2

existence of an 'ontology' that specifies the relationships among different skills.
2) After this pre-processing we can apply a ranking algorithm to the enriched endorsement digraph, so as to compute an authority score for each network member with respect to the main skill. In particular, we have used the (weighted) PAGERANK algorithm for that purpose, but in principle, any ranking method could be used, provided that it admits weighted digraphs. This authority score could be useful for a conceivable tool for searching people having a certain skill. Thus, the results of a query might be displayed in decreasing order of authority.
3) We propose a methodology to validate our algorithm, which does not rely as heavily on the human factor as previous validation methods, or on the availability of private information of the members' profiles. Following this methodology, we test our solution on a synthetic network of 1493 nodes and 2489 edges, similar to LINKEDIN, and fitted with endorsements [13].

To the best of our knowledge, this is the first proposal that ranks users of a social network according to their proficiency in some skill, based on endorsements. Moreover, we are not aware of any other work that suggests to enhance the endorsement digraph corresponding to some particular skill, with information obtained from related skills.

The rest of the paper is organized as follows: Section II provides the essential concepts, terminology and notation that will be used throughout the rest of the paper. It also describes the PAGERANK algorithm, including the variant for weighted digraphs. After that, our proposal is explained in Section III together with a simple example. In Section IV we compare the results obtained by ranking with deduction with those obtained by simple ranking, according to three criteria proposed by ourselves.

II. PRELIMINARIES

A. Terminology and notation

A *directed graph*, or *digraph* $D = (V, A)$ is a finite nonempty set V of objects called *vertices* and a set A of ordered pairs of vertices called *arcs*. The *order* of D is the cardinality of its set of vertices V. If (u, v) is an arc, it is said that v is *adjacent from* u. The set of vertices that are adjacent from a given vertex u is denoted by $N^+(u)$ and its cardinality is the *out-degree* of u, $d^+(u)$.

Given a digraph $D = (V, A)$ of order n, the adjacency matrix of D is an $n \times n$ matrix $\mathbf{M} = (m_{ij})_{n\times n}$ with $m_{ij} = 1$ if $(v_i, v_j) \in A$, and $m_{ij} = 0$ otherwise. The sum of all elements in the i-th row of M will be denoted Σm_{i*}, and it corresponds to $d^+(v_i)$.

A *weighted digraph* is a digraph with (numeric) labels or *weights* attached to its arcs. Given $(u, v) \in A$, $\omega(u, v)$ denotes the weight attached to that arc. In this paper, we only consider directed graphs with non-negative weights. The reader is referred to Chartrand and Lesniak [4] for additional concepts on digraphs.

B. PAGERANK *vector of a digraph*

PAGERANK [2][12] is a link analysis algorithm that assigns a numerical weighting to the vertices of a directed graph. The weighting assigned to each vertex can be interpreted as a relevance score of that vertex inside the digraph.

The idea behind PAGERANK is that the relevance of a vertex increases when it is linked from relevant vertices. Given a directed graph $D = (V, A)$ of order n, assuming each vertex has at least one outlink, we define the $n \times n$ matrix $\mathbf{P} = (p_{ij})_{n\times n}$ as,

$$p_{ij} = \begin{cases} \frac{1}{d^+(v_i)} & \text{if } (v_i, v_j) \in A, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Those vertices without oulinks are considered as if they had an outlink pointing to each vertex in D (including a loop link pointing to themselves). That is, if $d^+(v_i) = 0$ then $p_{ij} = 1/n$ for each j. Note that $\mathbf{P}$ is a stochastic matrix whose coefficient p_{ij} can be viewed as the probability that a surfer located at vertex v_i jumps to vertex v_j, under the assumption that the next movement is taken uniformly at random among the arcs emanating from v_i. When the surfer falls into a vertex v_i such that $d^+(v_i) = 0$, then he/she is able to restart the navigation from any vertex of D uniformly chosen at random. So as to permit this random restart behaviour when the surfer is at any vertex (with a small probability $1 - \alpha$), a new matrix $\mathbf{P}_\alpha$ is created as,

$$\mathbf{P}_\alpha = \alpha\mathbf{P} + (1 - \alpha)\frac{1}{n}\mathbf{J}^{(n)}, \quad (2)$$

where $\mathbf{J}^{(n)}$ denotes the order-n all-ones square matrix.

By construction, $\mathbf{P}_\alpha$ is a positive matrix [11], hence, $\mathbf{P}_\alpha$ has a unique positive eigenvalue (whose value is 1) on the spectral circle. The PAGERANK *vector* is defined to be the (positive) left-hand eigenvector $\mathcal{P} = (p_1, \ldots, p_n)$ with $\sum_i p_i = 1$ (the left-hand Perron vector of $\mathbf{P}_\alpha$) associated to this eigenvalue. The probability α, known as the *damping factor*, is usually chosen to be $\alpha = 0.85$.

The relevance score assigned by PAGERANK to vertex v_i is p_i. This value represents the long-run fraction of time the surfer would spend at vertex v_i.

C. PAGERANK *vector of a weighted digraph*

When the input digraph is weighted, the PAGERANK algorithm is easily adapted so that the probability that the random surfer follows a certain link is proportional to its (positive) weight [15]. This is achieved by slightly modifying the definition, previously given in (1), of matrix $\mathbf{P}$ so that,

$$p_{ij} = \begin{cases} \frac{\omega(v_i,v_j)}{\sum_{v\in N^+(v_i)}\omega(v_i,v)} & \text{if } (v_i, v_j) \in A, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Nodes with no outlinks are treated in the same way as before.

III. ENDORSEMENT DEDUCTION AND RANKING

Let us consider a professional network in which users can indicate a set of topics they are skilled in, and be endorsed for those skills by other users. For each skill, we get an endorsement digraph. Our objective is to compute an authority ranking for a particular skill, which is not only based on the

endorsement digraph of that particular skill, but also takes into account the endorsement digraphs of other related skills. From now on, the skill for which we want to compute the ranking will be called the *main skill*.

Let $S = \{s_0, s_1, \dots s_\ell\}$ be the set of all possible skills, with s_0 being the main skill. Let $D_k = (V, A_k)$ denote the endorsement digraph corresponding to skill s_k, and let $\mathbf{M}_k$ be its adjacency matrix.

We now define the *skill deduction matrix* $\mathbf{\Pi} = (\pi_{kt})$ as follows: Given a pair of skills s_k and s_t, π_{kt} represents the probability that a person skilled in s_k also possesses the skill s_t. In other words, from s_k we can infer s_t with a degree of confidence π_{kt}. By definition, $\pi_{kk} = 1$ for all k. In this way, if some user endorses another user for skill s_k but no endorsement is provided for skill s_t, we can deduce that an endorsement (for s_t) should really be there with probability π_{kt}. In general, $\mathbf{\Pi}$ will be non-symmetric and sparse, thus it is better represented as a directed graph with weighted arcs.

Our proposal takes as input the skill deduction matrix $\mathbf{\Pi}$, together with those endorsement digraphs D_k, with $0 < k \leq \ell$, such that $\pi_{k0} > 0$. Without loss of generality, we will assume that the set of skills related to s_0 is $S_0 = \{s_k \mid k \neq 0,\ \pi_{k0} > 0\} = \{s_1, \dots, s_\ell\}$.

The proposed endorsement deduction method constructs a weighted endorsement digraph $D_0^{we} = (V, A_0^{we})$ on skill s_0, with weights ranging from 0 to 1, considering the endorsements deduced from related skills $\{s_1, \dots, s_\ell\}$.

1) First of all, if user v_i directly endorsed v_j for skill s_0, that is $(v_i, v_j) \in A_0$, then D_0^{we} has arc $(v_i, v_j) \in A_0^{we}$ with $\omega(v_i, v_j) = 1$ (that endorsement receives a maximum confidence level).
2) If $(v_i, v_j) \notin A_0$ but $(v_i, v_j) \in A_k$, for just one k, $1 \leq k \leq \ell$, then arc (v_i, v_j) is added to D_0^{we} with weight $\omega(v_i, v_j) = \pi_{k0}$, that is, the arc is assigned a weight that corresponds to the probability that v_i also considers v_j proficient in skill s_0, given an existing endorsement for skill s_k.
3) Finally, if $(v_i, v_j) \notin A_0$ but $(v_i, v_j) \in A_{k_1}, \dots, A_{k_\ell}$, then the arc (v_i, v_j) is assigned a weight corresponding to the probability that v_i would endorse v_j for s_0 given his/her endorsements for $s_{k_1}, \dots, s_{k_l}$. That is, let "$(s_{k_i} \to s_0)$" denote the event "endorse for skill s_0 given an endorsement for s_{k_i} (its probability is $p(s_{k_i} \to s_0) = \pi_{k_i,0}$) then (v_i, v_j) is assigned a weight that corresponds to the probability of the union event "$\cup_{k_i \in \{k_1,\dots,k_\ell\}}(s_{k_i} \to s_0)$", assuming those events are independent.

Next, we show how to construct the weighted adjacency matrix of D_0^{we} by iteratively adding deduced information from related skills. Computations are shown in (4). After the k-th iteration, matrix $\mathbf{Q}_k$ corresponds to the weighted digraph of skill s_0 after having added deduced information from skills $s_1, \dots, s_k$. The matrix computed after the last iteration $\mathbf{Q}_\ell$ corresponds to the weighted adjacency matrix of digraph D_0^{we}. Computations can be carried out as follows,

$$\mathbf{Q}_0 = \mathbf{M}_0 \tag{4a}$$

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} + \pi_{k0}((\mathbf{J}^{(n)} - \mathbf{Q}_{k-1}) \circ \mathbf{M}_k), \quad \text{for } k = 1, \dots, \ell, \tag{4b}$$

where the symbol '$\circ$' represents the Hadamard or elementwise product of matrices.

Note that (4b) acts on the entries of $\mathbf{Q}_{k-1}$ that are smaller than 1, and the entries equal to 1 are left untouched. If some entry $\mathbf{Q}_{k-1}(i,j)$ is zero, and the corresponding entry $\mathbf{M}_k(i,j)$ is non-zero, then $\mathbf{Q}_{k-1}(i,j)$ takes the value of $\mathbf{M}_k(i,j)$, modified by the weight π_{k0}. This corresponds to the second case above.

If $\mathbf{Q}_{k-1}(i,j)$ and $\mathbf{M}_k(i,j)$ are both non-zero, then we are in the third case above. To see how it works, let us suppose that some entry $\mathbf{M}_0(i,j)$ is zero, but the corresponding entries $\mathbf{M}_1(i,j), \mathbf{M}_2(i,j), \mathbf{M}_3(i,j), \dots$, are all equal to 1. In other words, person i does not endorse person j for the main skill (skill 0), but does endorse person j for skills $1, 2, 3, \dots$. In order to simplify the notation, we will drop the subscripts i, j, and we will refer to q_k as the (i,j)-entry of $\mathbf{Q}_k$. Applying (4), we get:

$$\begin{aligned} q_0 &= m_0 = 0 \\ q_1 &= q_0 + \pi_{1,0}(1 - q_0) = \pi_{1,0} \\ q_2 &= q_1 + \pi_{2,0}(1 - q_1) = \pi_{1,0} + \pi_{2,0}(1 - \pi_{1,0}) \\ &= \pi_{1,0} + \pi_{2,0} - \pi_{1,0}\pi_{2,0} \\ &\vdots \end{aligned}$$

which corresponds to the probabilities of the events $(s_1 \to s_0)$, $(s_1 \to s_0) \cup (s_2 \to s_0)$, and so on.

Once we have the matrix $\mathbf{Q}_\ell = (q_{ij})_{n \times n}$, we can apply any ranking method that admits weighted digraphs, such as the weighted PAGERANK algorithm [15]. For that purpose, we have to construct the normalized weighted link matrix $\mathbf{P}$, as in (3):

$$p_{ij} = \begin{cases} \frac{q_{ij}}{\Sigma q_{i*}} & \text{if } \Sigma q_{i*} > 0, \\ \\ \frac{1}{n} & \text{if } \Sigma q_{i*} = 0. \end{cases} \tag{5}$$

Then we compute $\mathbf{P}_\alpha$ from $\mathbf{P}$, as in (2), and we finally apply the weighted PAGERANK algorithm on $\mathbf{P}_\alpha$.

A. An example

As a simple illustration, let us consider a set of three skills: 'Programming', 'C++' and 'Java'. The probabilities relating them, depicted in Figure 1, have been chosen arbitrarily, but in practice, they could have been obtained as a result of some statistical analysis.

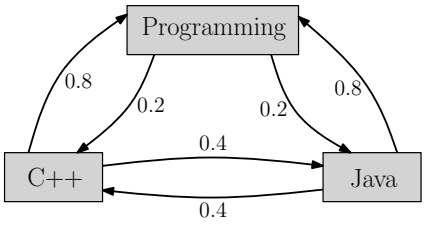


Figure 1. Directed graph representing a skill deduction matrix $\mathbf{\Pi}$.

Let us further assume that we have a community of six individuals, labeled from '1' to '6'. Figure 2 shows the

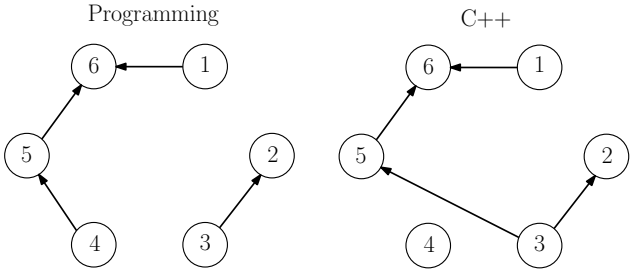


Figure 2. Endorsements for 'Programming'(left) and 'C++'(right).

endorsement digraphs among the community members for the skills 'Programming' and 'C++'.

Let us suppose that the skill 'Programming' is our main skill (skill 0). Thus, $\mathbf{Q}_0 = \mathbf{M}_0$ is the adjacency matrix of the digraph shown in Figure 2 (left). If we compute the PAGERANK for the skill 'Programming', without considering its relationships with other skills, we get the following scores ($\mathcal{P}(v)$ denotes the PAGERANK score assigned to vertex v): $\mathcal{P}(1) = \mathcal{P}(3) = \mathcal{P}(4) = 0.0988$, $\mathcal{P}(2) = \mathcal{P}(5) = 0.1828$, and $\mathcal{P}(6) = 0.3380$.

In other words, on the basis of the endorsements for 'Programming' alone, the individuals '2' and '5' are tied up, and hence equally ranked.

Now we will include the endorsements for 'C++' in this analysis (skill 1). We apply (4) to compute $\mathbf{Q}_1$, as follows:

$$\mathbf{Q}_1 = \mathbf{Q}_0 + \pi_{1,0}((\mathbf{J}^{(6)} - \mathbf{Q}_0) \circ \mathbf{M}_1),$$

where $\pi_{1,0} = 0.8$, and $\mathbf{M}_1$ is the adjacency matrix of the digraph shown in Figure 2 (right). This yields the endorsement digraph depicted in Figure 3.

The PAGERANK scores assigned to nodes in that digraph are: $\mathcal{P}(1) = \mathcal{P}(3) = \mathcal{P}(4) = 0.0958$, $\mathcal{P}(2) = 0.1410$, $\mathcal{P}(5) = 0.2133$, and $\mathcal{P}(6) = 0.3585$. The individuals '2' and '5' are now untied, and we have better grounds to trust Programmer '5' over Programer '2'.

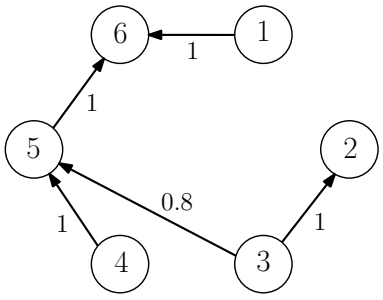


Figure 3. Endorsements for 'Programming', with information deduced from 'C++'.

Let us now suppose that the endorsement digraph for 'Java' is the one given in Figure 4 (left). We can include the endorsements for 'Java'in the same manner:

$$\mathbf{Q}_2 = \mathbf{Q}_1 + \pi_{2,0}((\mathbf{J}^{(6)} - \mathbf{Q}_1) \circ \mathbf{M}_2),$$

where again $\pi_{2,0} = 0.8$. The result is given in Figure 4 (right).

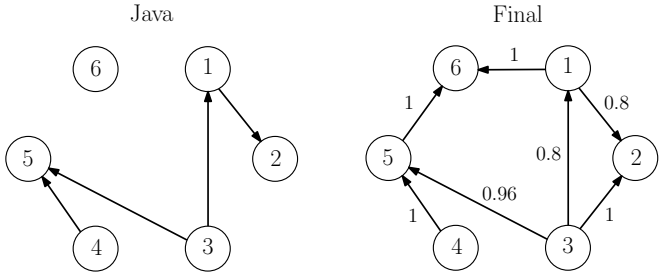


Figure 4. Endorsements for 'Java' (left), and endorsements for 'Programming', with information deduced from 'C++' and 'Java' (right).

If we apply PAGERANK to this final digraph we get: $\mathcal{P}(1) = 0.1178$, $\mathcal{P}(2) = 0.1681$, $\mathcal{P}(3) = \mathcal{P}(4) = 0.0945$, $\mathcal{P}(5) = 0.2027$, and $\mathcal{P}(6) = 0.3224$.

With the aid of the new endorsements, Programmer '1' differentiates itself from Programmers '3' and '4'.

IV. SIMPLE RANKING VS. RANKING WITH DEDUCTION

A. Evaluation criteria

Several criteria and measures have been developed for evaluating information retrieval and ranking systems, such as *precision*, *recall*, *F-measure*, *average precision*, *P@n*, etc. (see [3], Sec. 1.2). All these measures rely on a set of assumptions, which include, among others, the existence of:

1) a benchmark collection E of personal profiles (potential experts),
2) a benchmark collection S of skills,
3) a (total binary) judgement function $r : E \times S \to \{0, 1\}$, stating whether a person $e \in E$ is an expert with respect to a skill $s \in S$.

Unfortunately, none of these assumptions applies in our case. To the best of our knowledge, there does not exist any reliable open-access ground-truth dataset of experts and skills, connected by endorsement relations. To begin with, the endorsement feature is relatively new, and still confined to a few social networks, so that not enough data has accumulated so far. On the other hand, LINKEDIN does not disclose sensitive information of its members (including their contacts or their endorsements), due to privacy concerns.

The third assumption is also problematic: Even if we had a dataset with endorsements, we would still need a 'higher authority', or an 'oracle', to judge about the expertise of a person. Moreover, since our goal is to rank experts, a binary oracle would not suffice.

Traditionally, ranking methods have been validated by carrying out surveys among a group of users [6], which in our opinion, is very subjective and error-prone. We propose a more objective validation methodology, which is based on the following criteria:

1) Our ranking with deduction is close to the ranking provided by PAGERANK. This criterion is based on the assumption that GOOGLE's PAGERANK is widely accepted as a good method, as it has been validated by millions of users for more than fifteen years now. If we use endorsement deduction in connection with PAGERANK, results should not differ too much from PAGERANK.

 ISBN: 978-1-61208-444-2

2) Ranking with deduction results in less ties than PAGERANK. Ties are an expression of ambiguity, hence a smaller number of ties is desirable. In the example of Section III, we have seen that ranking with deduction resolves a tie produced by PAGERANK. However, this has to be confirmed by meaningful experiments.
3) Ranking with deduction is more robust than PAGERANK to *collusion spamming*. Collusion spamming is a form of *link spamming*, i.e., an attack to the reputation system, whereby a group of users collude to create artificial links among themselves, and thus manipulate the results of the ranking algorithm, with the purpose of getting higher reputation scores than they deserve [7][8].

B. Experimental setup and results

Our experimental benchmark consists of a randomly generated social network that replicates some of the features of LINKEDIN at a small scale [13]. LINKEDIN consists of an undirected *base network* (L), or *network of contacts*, and for each skill, the corresponding endorsements form a directed subgraph of (L). In [10], Leskovec formulates a model that describes the evolution of several social networks quite accurately, including LINKEDIN, although this model is limited to the network of contacts (L), and does not account for the endorsements, since that feature was introduced in LINKEDIN later. We have implemented Leskovec's model and used it to generate an undirected network of contacts with 1493 nodes and 2489 edges.

Additionally, we have considered five skills: 1. Programming, 2. C++, 3. Java, 4. Mathematical Modelling, 5. Statistics. We have chosen these skills for two main reasons:

1) These five skills abound in a small LINKEDIN community consisting of 278 members, taken from our LINKEDIN contacts, which we have used as a sample to collect some statistics.
2) These five skills can be clearly separated into two groups or clusters, namely programming-related skills, and mathematical skills, with a large intra-cluster correlation, and a smaller inter-cluster correlation. This is a small-scale representation of the real network, where skills can be grouped into clusters of related skills, which may give rise to different patterns of interaction among skills.

We have computed the co-occurrences of the five skills in our small community, resulting in the co-occurrence matrix $\mathbf{\Pi}_1$ of (6). The entry $\mathbf{\Pi}_1(i,j)$ is the ratio between the number of nodes that have been endorsed for both skills, i and j, and the number of nodes that have been endorsed for skill i alone.

$$\mathbf{\Pi}_1 = \begin{pmatrix} 1 & 0.42 & 0.42 & 0.5 & 0.33 \\ 0.62 & 1 & 0.62 & 0.25 & 0.12 \\ 0.62 & 0.62 & 1 & 0.12 & 0.12 \\ 0.75 & 0.25 & 0.12 & 1 & 0.5 \\ 0.5 & 0.12 & 0.12 & 0.5 & 1 \end{pmatrix} \quad (6)$$

Now, for each skill we have constructed a random endorsement digraph (a random sub-digraph of the base network), in such a way that the above co-occurrences are respected. We have also taken care to respect the relative endorsement frequency for each individual skill. The problem of constructing random endorsement digraphs according to a given co-occurrence matrix is not trivial, and may bear some interest in itself [13]. The base network and the endorsement digraphs can be found at [14].

Next, we have computed two rankings for each skill, one using the simple PAGERANK algorithm, and another one using PAGERANK with deduction. For PAGERANK with deduction we have used the skill deduction matrix $\mathbf{\Pi}_2$ given in (7). This matrix has been constructed by surveying a group of seven experts in the different areas involved. For real cases involving a large set of skills, $\mathbf{\Pi}_2$ must be generated automatically via data mining techniques.

$$\mathbf{\Pi}_2 = \begin{pmatrix} 1 & 0.7 & 0.7 & 0.4 & 0.3 \\ 1 & 1 & 0.6 & 0.4 & 0.3 \\ 1 & 0.7 & 1 & 0.4 & 0.3 \\ 0.3 & 0.2 & 0.2 & 1 & 0.8 \\ 0.3 & 0.2 & 0.2 & 1 & 1 \end{pmatrix} \quad (7)$$

For each skill, we have computed the correlation between both rankings, and the number of ties in each case, according to the first two criteria described above. Additionally, in order to test the robustness of the method to collusion spamming, we have added to each endorsement digraph, a small community of new members (the *cheaters*), who try to subvert the system by promoting one of them (their *leader*) as an expert in the corresponding skill. We have chosen the most effective configuration for such a spamming community, as described in [7], and depicted in Figure 5. Thereupon, we have compared the position of the leader of cheaters in simple PAGERANK with its position in PAGERANK with deduction.

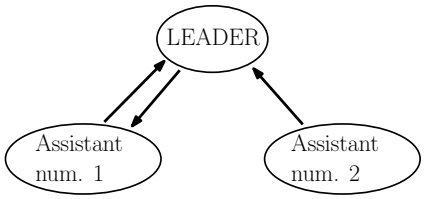


Figure 5. Link spam alliance: Three people collude to promote one of them.

Table I summarizes the results of the aforementioned experiments. We can see that there is a very high correlation between PAGERANK with deduction and PAGERANK without deduction for all skills, according to the values of Kendall's τ and Spearman's ρ correlation coefficients. With respect to the second criterion, the experiments also yield unquestionable results: For all skills, the number of ties is significantly reduced. As for the third criterion, in all cases there is a detectable drop in the position of the leader of cheaters, which may lead us to conclude that PAGERANK with deduction is more robust to collusion spam than simple PAGERANK. The last column of the table contains the difference between the position of the leader with deduction and without deduction, expressed as a percentage.

However, this may not lead us to the conclusion that PAGERANK with deduction is an effective mechanism against collusion spam. Actually, the spam alliance that we have introduced in our experiments is rather weak. If we strengthen the spam alliance, then PAGERANK with deduction may also be eventually deceived.

Several effective mechanisms have been proposed to fight collusion spam, an example being the so-called *asymmetric*

 ISBN: 978-1-61208-444-2

TABLE I. RESULTS OF THE EXPERIMENT WITH LOW-DENSITY ENDORSEMENT DIGRAPHS

Skill	Number of endorsements (arcs)	Correlation ρ	Correlation τ	Number of ties without deduction	Number of ties with deduction	Number of ties % reduction	Position of leader without deduction	Position of leader with deduction	Position of leader % fall
Programming	220	0.89	0.76	1460	1316	10%	1	48	3%
C++	140	0.85	0.63	1478	1304	12%	4	48	3%
Java	137	0.85	0.63	1486	1292	13%	1	48	3%
Math Modeling	134	0.85	0.63	1483	1318	11%	1	45	3%
Statistics	128	0.85	0.63	1486	1304	12%	1	45	3%
AVG						11.6%			3%

TABLE II. RESULTS OF THE EXPERIMENT WITH HIGHER-DENSITY ENDORSEMENT DIGRAPHS

Skill	Number of endorsements (arcs)	Correlation ρ	Correlation τ	Number of ties without deduction	Number of ties with deduction	Number of ties % reduction	Position of leader without deduction	Position of leader with deduction	Position of leader % fall
Programming	427	0.76	0.63	1428	625	56%	1	175	12%
C++	1793	0.97	0.93	1005	575	43%	66	178	7%
Java	1856	0.97	0.93	1005	566	44%	63	180	8%
Math Modeling	1406	0.95	0.89	1130	652	42%	56	168	7%
Statistics	1447	0.96	0.90	1113	580	48%	58	169	7%
AVG						47%			8%

reputation systems. A complete survey of such systems is given in [9]. Presumably, these mechanisms will give better results when combined with deduction.

On the other hand, our endorsement digraphs are rather sparse. It is reasonable to predict that if we should consider more skills, and if the total number of endorsements should increase, then the effects of PAGERANK with deduction will be stronger.

In order to verify this prediction, we have carried out a second experiment on the same base network and the same set of skills, increasing the number of endorsements. Thus, we have generated a second set of endorsement digraphs, with a larger number of arcs. This time we cannot enforce the co-occurrences observed in our small LINKEDIN community. Subsequently we have performed the same computations on this second set of endorsement digraphs, obtaining the results recorded in Table II. These results fully confirm our prediction: There is an increase in the correlation coefficients (except in one case), as well as a larger reduction in the number of ties, and a more significant fall in the position of the leader of cheaters.

ACKNOWLEDGEMENTS

The authors acknowledge partial support by the Spanish Government under projects TIN2010-18978 and MTM2013-46949-P, and by the Government of Catalonia under grant 2014SGR-1666. We also thank the anonymous referees for their timely and constructive comments, which have helped us improve the content and the format of the paper. Finally, the first two authors wish to dedicate this paper to the memory of our friend and co-author Josep Maria Ribó Balust.

REFERENCES

[1] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si, "Expertise Retrieval," Foundations and Trends in Information Retrieval, vol. 6, no. 1–2, 2012, pp. 127–256.

[2] P. Berkhin, "A Survey on PAGERANK Computing," Internet Mathematics, vol. 2, no. 1, 2005, pp. 73–120.

[3] S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali, and S. Quarteroni, Web Information Retrieval. Springer, 2013.

[4] G. Chartrand and L. Lesniak, Graphs and Digraphs. CRC Press, Boca Raton, fourth ed., 2004.

[5] H. Deng, I. King, and M. R. Lyu, "Enhanced Models for Expertise Retrieval Using Community-Aware Strategies," IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 42, 2012, pp. 93–106.

[6] K. Fujimura, H. Toda, T. Inoue, N. Hiroshima, R. Kataoka, and M. Sugizaki, "BLOGRANGER - A multi-faceted blog search engine," Procs. WWW 2006, 2006, pp. 22–26.

[7] Z. Gyöngyi and H. Garcia-Molina, "Link Spam Alliances," Procs. 31st VLDB, 2005.

[8] Z. Gyöngyi and H. Garcia-Molina, "Web Spam Taxonomy," Procs. AIRWeb, 2005.

[9] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A Survey of Attack and Defense Techniques for Reputation Systems," ACM Computing Surveys, vol. 42, 2009, pp. 1–31.

[10] J. Leskovec, "Dynamics of Large Networks," PhD Thesis, School of Computer Science, Carnegie-Mellon University, 2008.

[11] C. D. Meyer, Matrix Analysis and Applied Linear Algebra. SIAM, 2001.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PAGERANK Citation Ranking: Bringing Order to the Web," Technical Report, Stanford InfoLab, 1998.

[13] H. Pérez-Rosés and F. Sebé, "Synthetic Generation of Social Network Data with Endorsements," Journal of Simulation, 2014, DOI:10.1057/jos.2014.29.

[14] H. Pérez-Rosés and F. Sebé, Dataset of Endorsements, http://www.cig.udl.cat/sitemedia/files/MiniLinkedIn.zip, accessed in November, 2015.

[15] W. Xing and A. Ghorbani, "Weighted PAGERANK Algorithm," Procs. of the 2nd Annual IEEE Conference on Communication Networks and Services Research, 2004, pp. 305–314.

 ISBN: 978-1-61208-444-2

Wireless Sensor Technologies in Food Industry: Applications and Trends

Saeed Samadi, Hossein Mirzaee
Food Machinery dept.
Research Institute of Food Science and Technology (RIFST)
Mashhad, Iran
email: s.samadi@rifst.ac.ir, h.mirzaee@rifst.ac.ir

***Abstract*— Wireless Sensors Networks (WSNs) have emerged as an exciting new computing technology in food industry, particularly with the proliferation in Micro-Electro-Mechanical Systems (MEMS) technology providing advanced development of smart sensors. Such convergences of technologies which involve deploying self-contained sensor devices that sense and process data have opened up a revolution in the application domain of sensors in food industry. This includes the realization of more versatile systems in food industry regarding measurements, instrumentation and automation of functions. The vision of this technology is to develop a real time monitoring of in-door and outdoor environments. WSNs are deployed in various applications which enhance interaction between users and the physical environment. This has allowed physical environment in systems, such as in food industry, to be measured using a high resolution with the aim of increasing and monitoring the quantity and quality of real-time data and information collected for the applications. This paper reviews the application of wireless sensor technologies in food industry, looks into its application in real-time monitoring of food supply chain, food security and food intelligent packaging while discussing various challenges that should be addressed in order to push the technology further.**

Keywords- food industry; Wireless Sensor Networks; RFID; packaging.

I. INTRODUCTION

The food industry is known to be receptive to the use of information systems in all aspects. For many years, technology in food industry had focused on Enterprise Resource Planning (ERP) software. However, this has recently changed as many companies in the food industry have started adopting state-of-the-art technologies to provide solutions in areas such as traceability and logistics. With recent incidents in food safety issues, there has been an inevitable increase in interest in Production Life Cycle Management Software in combination with Radio Frequency Identification Technology (RFID) focusing on providing solutions for addressing food safety [1]. The potential applications of Smart Sensor Networks are diverse, especially in external systems in which Bluetooth has proven to be the communication standard between sensors. Wireless sensor networks consist of tiny devices referred to as sensor nodes that are battery-powered and equipped with sensing devices, a data processing unit and a communication unit [2]. Depending on the external interface in application, sensors for an attended wireless network, when randomly deployed in the field, are able to collect and aggregate data as well as communicate with electronic devices, such as PCs, PDAs and Laptops through which data can be transmitted to any network [3]. Regarding food security as a pressing issue in the food industry globally, wireless sensor networks can be used to detect contaminants in the food supply chain, provide real time monitoring of the cold chain industry [4], pest detection and control in farms [3] and food intelligent packing through the use of RFID and Surface Acoustic Wave (SAW) [5].

The rest of this paper is organized as follows: Section II summarizes a review of literature. Section III discusses some important applications of WSN technologies in the food industry. Section IV discusses in finer detail, the aim of the paper. Section V includes an acknowledgement and conclusions.

II. LITERATURE REVIEW

A. Overview of Wireless Sensor Networks

Wireless sensors consist of small devices called sensor nodes which are powered and equipped with self-contained sensing, data processing, data storage and communication unit. When deployed to any environment, the sensor nodes create a random wireless network in which data can be collected, aggregated and packaged for further fusion through other devices such as PCs and PDAs. Increasing capabilities of sensor nodes have led to the realization of wireless sensor networks. Advancements in wireless sensor node technology have been targeted by universities and research laboratories in experiments and development tests. However, there is increasing interest by many companies globally which intend to use the technology in building and industrial automation markets [1].

A sensor network is composed of thousands of small devices known as sensor nodes that are deployed inside or close to the phenomenon. Sensors are generally devised to sense information and transmit the same to a mote. The

 ISBN: 978-1-61208-444-2

information collected is used to measure changes in the physical environment such as pressure, humidity, sound and vibration. A mote consists of a processing unit, storage unit and power source, which can be a battery and a radio transmitter which it uses to form an adhoc network. Combining a mote and a sensor forms a sensor node. A typical wireless sensor node is composed of a sensing unit, a processing unit, a communication and power unit. Advancements have also been made to include location finding systems, power generators and mobilizers into sensor node. Recent technology advancements have enabled the development of self-contained motes [2]. High power density batteries are commonly preferred to achieve longer lifetime of the sensor nodes. Future wireless sensor nodes trends focus on smaller node size as well as very low energy consumption [6]. WSN differs from RFID in that it is able to integrate with other network devices in the field while an RFID tag can only be read with the RFID tag reader. WSN is comprised of Wi-Fi, Bluetooth, and ZigBee. The latter two operate within the Industrial Scientific and Medical (ISM) band of 2.4 GHz, which provides license-free operations, enormous spectrum allocation, and global compatibility [7]. Table I provides a comparison between these WSN technologies.

TABLE I. COMPARISON OF WIRELESS SENSOR NETWORKS TECHNOLOGIES AND STANDARDS

Feature	Technology		
	Wi-Fi	***Bluetooth***	***ZigBee***
Physica Layer Standard	IEEE 802.11	802.15.1	802.15.4
Data rate	11 Mbps	1Mbps	250kbps
Node per master	32	8	64000
Range	10 to 100 m	10 m	10 to 300 m
Data type	video, audio, graphics, pictures, files	video, audio, graphics, pictures, files	Small data packet
Topology	star	Star	Mesh,stsr,tree
Complexity	complex	very complex	simple
Battery life	hours	1Week	1year

There are thousands of sensor node platforms developed with varying components and operating systems depending on the categories of their research groups. Current WSNs are deployed on land, underground and underwater, which presents different challenges depending on the environment. The common types of wireless sensor networks in the market include; terrestrial WSN, Underground WSN, Underwater WSN, Multi-media WSN and Mobile WSN [6]. In terrestrial WSNs, a large number of wireless sensor nodes are deployed by placing them in the target area by a plane using 2-d, 3-d placement models. This enables multi-hop optimal routing, shorter transmission ranges and low data redundancy as well as minimal delays. Underground WSN is a group of nodes whose means of data transmission and reception is completely underground. It may be completely embedded in dense soil or rock.

B. Challenges to be addressed in order to push the WSN technology further

In underground WSNs, underground sensor nodes are supported by additional sinks nodes above ground to boost information relay from the sensor nodes to the base station. This presents a challenge in wireless communication due to frequent signal losses and high level of attenuation. Sensor nodes deployed underground require longer battery life due to lack of recharging media beneath surface. In underwater WSNs, the sensor nodes rely on acoustic wave transmissions which are challenged by limited bandwidth, long propagation, delays and signal fading. Multi-media WSNs consist of sensor nodes equipped with cameras and microphones, which interconnect over wireless networks to be able to retrieve the process and compress data. The main challenge in this model is a high bandwidth demand, high energy consumption and quality of service provisioning [6].

Power consumption is the main challenge that all wireless sensor networks consider during design. Since wireless sensors are powered by battery or energy harvesters, hardware that uses power intelligently is critical in determining the longevity of the devices. Development of ultra-low power networks is one of the developments being undertaken in wireless sensor technologies. With the current development of ultra-low power transceiver radio chips, it is now possible to develop low power wireless sensor applications with efficient energy harvesting, conversion and usage. Hardware costs also pose a challenge through increasing the overall expenditure in wireless sensor network developments. Most of the systems being used in wireless sensor applications and networking architecture are vertically integrated to be able to utilize performance. This can be improved through the development of more stable and mature systems and networking architecture. The use of low power radio frequency transceivers are affected by poor wireless reception in indoors environment such as buildings

Energy management in WSNs is another critical viewpoint because of the resource constraints. Remotely deployed energy stringent sensor nodes are typically powered by attached batteries. Also, resource constraints are addressed utilizing the diverse equipment units, protocols, and radio. Data transmission, in particular, is one of the most power consuming aspect. Hence, the power efficiency of communication between the base station and surrounding components is of paramount importance. Another issue may be that WSNs, due to their continuous monitoring nature, make tremendous amounts of data that are hard to manage, bringing about a colossal increment in the daily volume of information stored in a corporate data warehouse system.

B WSNs for real time monitoring of the food supply chain

With increasing strict regulations in the food supply chain management with the aim of increasing higher product quality and public safety, the need to track and trace the treatments of all ingredients and the use of wireless sensor network technology has become prevalent in most

 ISBN: 978-1-61208-444-2

researches. Traceability, global standards and adoption of radio frequency (RF) technologies have been widely investigated through scientific experiments in universities and by companies in food industry. It is now possible to track food products in the market in the entire food supply chain using RFID tags, which is a more superior technology to barcodes and labels. RFID tags have sensors equipped with unique ID for products or their batches. The sensors attached on tags are able to form a sensor network, which monitors food temperature and humidity. This provides continuous monitoring of data throughout the food supply chain, which provides assurance to retailer requirements such as maintaining the required temperature throughout the delivery and storage process of products.

WSN technology is used in agriculture industry for monitoring and surveillance of crops within a farm. However, weather variation is the sole challenge that affects performance of WSN in this industry. The technology utilizes radio frequencies that can be interfered byweather conditions. The technology is used in maintenance and monitoring of farmlands. This is achieved through installation of sensors and cameras on the field. These devices are linked to the control station on the farm via the mentioned wireless technology. Monitoring fields enables identification of severe conditions of the soil and weather; with such information, farmers may make comprehensive decisions concerning planting activities. Wireless technology also enables pest control and irrigation activities that are essential when pursuing maximum yield. Sensors deployed on the soil are able to determine moisture content of the soil. When soil moisture content is below the minimum, the information is transferred to the control that commands the irrigator to sprinkle the soil. Phytophotra is a disease that affects potatoes and is influenced by temperature and humidity conditions. Between 868MHz and 916MHz, motes can be used in determining moisture content on air and temperature [8]. Extreme temperatures can be reflected and relayed to the control station, which initiates spray of pesticides.

III. SELECTED APPLICATIONS OF WSNS IN FOOD INDUSTRY

WSNs have proved to be vital in certain applications that require data to be monitored in real time. Particularly, such networks have been widely used in precision agriculture. In this regard, agricultural sector has recently made use of WSNs in an attempt to enhance the monitoring operations associated with this sector. In light of this, precision agriculture refers to the science involving precise comprehending, approximating as well as evaluating the crop condition with the intent of determining the real irrigation needs, and correct utilizers during harvesting and sowing seasons.

Precision agriculture is actually an integrated product-based farming system and information specifically designed to boost productivity and minimize unintended effects of equipment failure, the environment and wildlife. Figure 1 depicts WSN deployment for agriculture.

Since wireless sensor network is an ad-hoc kind of network, it does not need infrastructure as it is the case for other technologies. It can contain several nodes (unassisted embedded components) which are used in processing and transmitting data gathered from various onboard physical sensors such as soil moisture, humidity, pressure wetness and temperature. It also entails base section which serves as the gateway between end users and nodes or between nodes.

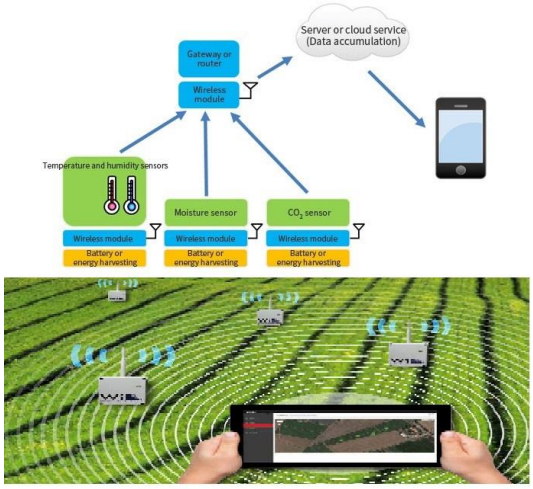


Figure 1. WSN deployment for the agricultural applications

Wireless sensor network technology has a variety of advantages as well as limitations. For instance, proper utilization of fertilizers and controlled irrigation can preserve resources and reduce wastage. Additionally, proper utilization of pesticides can minimize losses and improve and maintain the quality of product as well as enhance profitability. However, limitations associated with the deployment of WSNs for agricultural applications include security challenges concerning routing operations, physical security of software and hardware, and limited power [3].

Precision agriculture assists in reducing carbon dioxide, methane, nitrogen as well as other harmful liquid and gases emission. On the other hand, crop surveillance mainly focuses on understanding and monitoring the needs of crops in accordance to weather, as well as managing the available resources. Wireless sensor network also assists in preserving precious resources, effective utilization of such resources, and reducing wastage. This is referred to as proficient resource distribution [9].

Generally, wireless sensor networks have been employed in agriculture for pest control. In this regard, it is argued that "sensor network-based decision support system for agriculture" is used to conduct an experiment regarding pests and crop-weather [3]. Researchers are further trying to comprehend the hidden correlation between weather and pest and disease. In this case, a corrective prediction model has been developed that could be used to understand the correlation between such parameters in the future. The proposed model is shown in Figure 2.

 ISBN: 978-1-61208-444-2

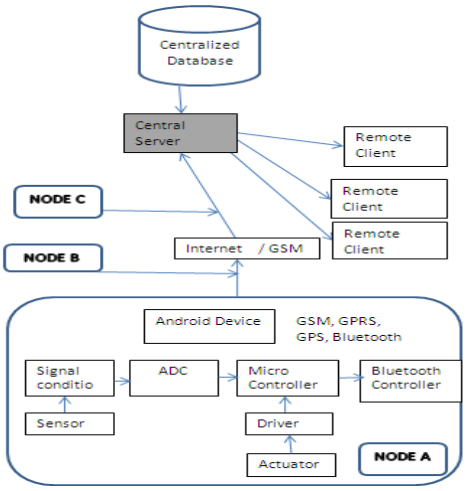


Figure 2. Proposed corrective prediction model for understanding the correlation between weather and pest and disease [3]

The proposed model is to offer ultra-low cost to the remote node because the user can only maintain a remote node rather than the primary node at the farm. The central server can then be accessed via web applications in order to obtain the relevant details regarding disease forecast and weather conditions [10].

On the other hand, the technology referred to as food intelligent packaging (IP) are currently under review with a special focus on the likelihood of using SAW and RFID technologies. According to [5], passive SAW and RFID technologies are more likely to attain a food intelligent packaging which can be used to wirelessly communicate to the food chains' different agents.

IV. DISCUSSION

Quality wireless sensor networks are required at different stages of food production and marketing. There are different types of wireless sensor networks including terrestrial WSN, underground WSN, underwater WSN, multimedia WSN and mobile WSN. All these WSNs are crucial in maintaining food security and quality throughout the supply chain. It is necessary for all companies in the food chain to adopt WSN to ensure that products can be traced, beginning from the producer to the consumer. This ensures that food products are clean, well labeled, and of good quality. Despite the fact that WSNs have had a positive impact on the agricultural sector, there are some visible constrains that inhibits their adoption in many countries. These challenges are both technically and culturally based.

Agronomists have been in opposition of WSN and distrust them with the belief that they are more informed and experienced in managing food production process. Secondly, there is still a big over-reliance on 1D optical barcodes, which inhibits successful application of other technologies.

The third challenge emerges from the costs of acquiring and setting up WSN [11]. For instant, many people around the world and more importantly in developing countries still practice small scale farming. As such, setting up WSN would offset other benefits such as low costs of farming. Still, the benefits associated with WSN regarding food security are immense and many countries are struggling to ensure a strong industrial adoption.

TABLE II. APPLICATIONS OF WSNS IN AGROFOOD INDUSTRY

Main Field of Application	Technology Used	Specific Function	Sensing Parameters
Environmental monitoring	Surface Acoustic Waves Resonator (SAW)	Weather monitoring Geo-referenced environment monitoring	Temperature, moisture, sunlight intensity
Advantage/ Functions	Easy to organize and process parameters		
Precision Agriculture	-Silage yield mapping system - Wireless infrared thermometer system -Automatic irrigation system	Spatial Data Collection Precision irrigation Variable rate technology Supplying data to farmers Automated fertilizer applicator for tree crops	Soil water availability, soil compaction, soil fertility, biomass yield, leaf area index, leaf temperature, leaf chlorophyll content, plant water status, insect disease weed infestation, grain yield.
Advantage/ Functions	-The system provides accurate field survey data enabling farm managers and engineers to monitor performance -The system is proved to have saved between 30-60% water usage		
Machine and process control	M2M technology Wireless personal safety device (WPSRD) Remote service system for agricultural machinery -Wireless Probe System (WPS)	Vehicle guidance Machinery management Robotic control Process control	State and condition of machinery, seamless remote control, cost efficiency
Advantage/ Functions	-The system prevents collision between human and motor vehicle properties in farms -The systems maximized production through ensuring minimum breakdown time -WPS improves accuracy and efficiency of food drying process and cost of data collection		
Facility automation	RF link (458 MHs), Solar Power data acquisition stations (SPSWAS)	Greenhouse control Animal feeding facilities	HVAC, Lighting, energy, access control, risk management,
Advantage/ Functions	- Improved productivity and reduce labor requirement		
Traceability system (RFID biosensor tags) Smart collars Acoustic wave systems	Hobo Pro data loggers (Onset computer corporation, Pacaset, MA)	Animal ID and health monitoring Food packaging Transportation Food inspection	Temperature, humidity, noise, light and ammonia content in the air -smart packing, automatic checkout and smart recycling, Vibration and animal behavior, bacterial concentration in food,
Advantage/ Functions	Improved security, traceability, productivity, inventory control, savings in capita and operational costs. High food monitoring quality		

 ISBN: 978-1-61208-444-2

Together with other forms of precision agriculture such as integration of scattered pieces of land, WSN had the ability of improving the quality of food products, enhancing sustainability, protecting the environment, and improving rural development [12]. Table II provides a comparison of main applications of wireless sensor networks in agrofood industry. Selected field of applications presents how this technology can be integrated to enhance safety and quality of food products and provide advantages such as mobility, transparency and autonomy. The WSN technology is mainly built on networked devices or utilizes networks for communication. However, much additional work still should be done for a large scale integrated communication and scalable coordination throughout the agro-food networks [13].

V. CONCLUSION

WSNs have gained increasing popularity over the last couple of years; they have helped revolutionize industrial operations starting with production to manufacturing to retailing. The agricultural sector has also seen an increased use of WSN to track the history of food products beginning from cultivation to post-harvest activities. Using WSNs, the history of food product starting from production to harvesting to marketing can be traced. The use of WSNs has seen improved food security around the globe. There are various benefits associated with WSNs despite the fact that there are still many challenges to overcome. For instance, there is a huge opposition from agronomists who believe that they have the knowledge and experience to manage fields. Further, the costs associated with acquiring and setting up WSNs is also a challenge to be addressed, particularly in developing economies where land is scattered and small scale farming is the major form of agriculture. Consolidation of such land is needed to ensure that the cost of using WSNs in agriculture is less compared with the gains. Furthermore, there is a need to address cultural issues such as the belief that food security cannot be fully achieved as well as the need for a vast campaign in a bid to educate stakeholders of the benefits of using WSN in the food production and marketing chain.

REFERENCES

[1] M. Connolly and F. O'Reilly, "Sensor networks and the food industry, Workshop on Real-World Wireless Sensor Networks,RealWSN, 2005, pp. 20-21.

[2] C. B. D. Kuncoro, "Miniature and Low-Power Wireless Sensor Node Platform: State of the Art and Current Trends," IPTEK Journal of Proceedings Series vol.1, no.1, 2015, pp 355-367.

[3] S. Azfar, A. Nadeem, and A. Basit, "Pest detection and control techniques using wireless sensor network: A review," Journal of Entomology and Zoology Studies, vol. 3, no.2, 2015, pp 92-99.

[4] C. Rhee, S. Berkovitch, Y. Kaufmann, and D. Wiseman, "USN applied to Smart Cold Chain based on the mesh wireless sensor network," In Industrial Engineering Proceedings, 2011, pp. 786-789.

[5] A. López-Gómez, et al. "Radiofrequency Identification and Surface Acoustic Wave Technologies for Developing the Food Intelligent Packaging Concept," Food Engineering Reviews vol.7, no.1, 2015, pp11-32.

[6] N. Srivastava, "Challenges of next-generation wireless sensor networks and its impact on society," Journal of Telecommunications, vol. 1, no.1, 2010, pp 128-133.

[7] A. Testa, A. Coronato, M. Cinque, and J. C. Augusto, "Static verification of wireless sensor networks with formal methods," In Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on , IEEE, November 2012, pp. 587-594.

[8] A. Baggio, "Wireless sensor networks in precision agriculture," ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden. 2005.

[9] M. Keshtgary and A. Deljoo, "An efficient wireless sensor network for precision agriculture," Canadian Journal on Multimedia and Wireless Networks, vol. 3, no.1, 2012, pp 1-5.

[10] S. Methley and C.Forster, "Wireless Sensor Networks Final Report," An Article of Plextek Limited, United Kingdom, 2008.

[11] L. Mainetti, L. Patrono, M. L. Stefanizzi, and R. Vergallo, "An innovative and low-cost gapless traceability system of fresh vegetable products using RF technologies and EPCglobal standard," Computers and electronics in agriculture, vol.98, 2013, pp 146-157.

[12] B. Talebpour, U. Türker, and U. Yegül, "The Role of Precision Agriculture in the Promotion of Food Security," International Journal of Agricultural and Food Research, vol. 4, no. 1, 2015, pp. 1-23.

[13] S. Samadi, "Applications and Opportunities for Internet-based Technologies in the Food Industry," In The Sixth International Conference on Advances in Future Internet, AFIN 2014, Lisbon, Portugal, Nov. 2014, pp 67-71.

 ISBN: 978-1-61208-444-2