

# A Self-Reconfigurable System of Contents among Smart Devices

Author: Hao Ren<sup>1,\*</sup>, Paul Kim<sup>2</sup>, Sangwook Kim<sup>3</sup>

## Abstract

In this system, mobile devices are not independent, they can communicate with each other, one device's change can affect the whole system or other devices. To achieve the above mentioned *A Self-Reconfigurable System of Contents*, through discover the device and connect process, to establish the connection between the mobile devices. After user assigns two dimension display type, the user can select content to input the system, contents are portioning and broadcast to devices. The system can self-reconfigure contents rapidly and exactly. This technique supports contents self-reconfiguration for devices remove, addition and position exchange. In this paper, when the user uses the hand contacts device, the device sends a signal to assist the system to detection device's position. The system does not need to get accurate devices moving direction, just according to all changed devices position to judge where the devices destination is. This research develops an application according to this technique, and the real machine tests the application using Android platform. Some communication protocols and mathematical modeling methods are proposed. These methods can also be used in other Internet of Things (IoT) fields, such as Drones Navigation, Smart Home, and Informational City management.

**Key Words:** contents, Internet of Things, mobile device, self-reconfiguration.

## I. INTRODUCTION

Expanding network connectivity to objects is an important part of a new generation of information technology, as known as the Internet of Things. This research motivated to create a method to support an object to perceive a change in another object's physical position and the object also changes its state correspondingly, thus solves the problem, which is manual operation' inconvenience and inaccuracy.

This research chooses the mobile devices as objects, develops an application to find the physical position of mobile devices. Moreover, it is proved through experiment the proposed algorithm is more feasible and accurate. The system establishes connections between mobile devices, and supports devices to perceive other device's existence. The system records device's initial physical position and then perceives change in device's position. The system sends message to notify device about the changing position of the other device. After receiving notification message, the device performs contents reconfiguration. Most of the contents equation techniques only support one device to display content [1], but this paper proposes a new perspective to support multi devices with cooperative

work for the user. And then according to user's intention makes some corresponding changes.

This research uses *Two Dimensional Relative Location* method to support mobile device to detect changing in other device position. *Contents Self-Reconfigurable System* can support device to do corresponding behavior. The proposed system has following major three contributions:

- Proposing a method of constructing nonlinear two dimension model by mobile device. This method supports the device to perceive other devices around it.
- *Contents Self-Reconfigurable System* supports multiple types of contents display on the multi-device. The system, then, does portioning the contents and shuffling contents to each device quickly and accurately.
- This system also supports the user action to remove, add, or exchanging the position of the devices in runtime. When any of the action like removing, adding, or exchanging device position is performed, the device self-reconfigures new contents based on the action.

About contents data transmission, this research proposes two methods as *Total Contents Splitting (TCS)*

Manuscript received June 27, 2015 ; Revised July 31, 2015 ; Accepted August 10, 2015. (ID No. JMIS-2015-18)

Corresponding Author(\*): School of Computer Science and Engineering, Kyungpook National University, Daehak-ro, Buk-gu, Daegu, Korea, +83-53-940-8881, rhao@media.knu.ac.kr.

<sup>1,2,3</sup>School of Computer Science and Engineering, Kyungpook National University, Daegu, Korea, rhao@media.knu.ac.kr, paulkim @media.knu.ac.kr, kimsw@knu.ac.kr.

and *Fragment Contents Splitting* (FCS). By experimentation, this research analyzed and discussed under what circumstances the designed system chooses which method is more appropriate.

Since this system only needs lightweight hardware support, like sensors and P2P communication module. The proposed system, *A Self-Reconfigurable System of Contents*, is not only suitable for smart devices but also can apply on a lot of things.

This paper is organized as follow: Section II reviews the related works. Section III describes the proposed *Contents Self-Reconfigurable System*. Section IV presents the development and implementation of the System. Section V discusses the experimental result for the system. Finally, Section VI describes the conclusion of the research work.

## II. RELATED WORKS

*Junkyard Jumbotron* is designed by the Center for Civic Media at MIT years ago[2]. In this scheme, the user is allowed to combine a group of mobile devices (including smart phones, tablet PCs and notebook computer) into a big screen. In this method, through input, an author provides a unique URL on every device (to connect server). Furthermore, the user takes a single photo for multiple screens and the image are analyzed by a special software that is mapping out the location of each screen in relation to the other device. Finally, the multiple images can be transferred to various mobile devices at the same time, by creating a fragmented effect. In this method, each device generates a unique QR code for identification of mobile devices. Further, the user uses other devices' camera to recognize their QR codes in order to get mobile devices' physical position, and then devices can map the contents on their screen

In *Junkyard Jumbotron* system, if the individual mobile device changes its physical position, it needs reposition and it needs manual operate contents presentation on the device again. In the research of *Dynamic Tiling Display System*[3], the author changes the position of device, and modifies the self-reconfigure contents on device. In this method, every device displays the same multimedia data and on using visual recognition by the camera device's individual view of the rendering can be determined.

Some researche not only presented contents on devices screen, they emphasize the user needs interact with Multi-devices[4]. In Jun Tanaka and Takashi Ohta's research[5], they devised an interface to relate applications, which are running on multiple smart phones. It is named "pinching",

a touch screens system for juxtaposed devices. Their system constructed a virtual display field by multiple screens. They design various contents of interaction that is a single screen unable to offer. In their research, the user can use a finger to touch two devices' screen at the same time to establish connection and device position detection. Device' screen appears an object, this object can move straight lines and cross device's bezel(visual effect) into the other devices.

A team from Aalborg University Centre for Society-Interactive Design shows results of their study Juxta Pinch [6]. In their design, they had two contributions, the first contribution discusses multi mobile devices interaction[7] application by a meaningful study. This study has opportunities and challenges related to use of several devices at the same time. Secondly, their contribution discusses the collocation image sharing research through demonstration of user interaction and exploration of friend's photos. Their research allows pinching three devices to connect devices. And they implemented JuxtaPinch as a web application. They developed a multi-platform to support more devices. Hence, their system is implemented by the socket.io library setup a Node.js websocket server. The sensor of accelerometer and gyroscope are used to detect when devices change its physical position[9].Jun Tanaka, Takashi Ohta's research and Aalborg University Centre for Society-Interactive Design's research and some other research [8] proposed their system gets device physical position information main using manual method.

In early 2014, researchers from Samsung issue their result of using *Galaxy Note 3* devices one by one like a tile as a commodious screen to display video[10]. For example, in figure 2.6 group video display across multiple devices is illustrated. It has its own drawback which is contents delivery. In this case, before device connects each other presents contents on screen has been already shared to all devices storage. When the devices start collaborating the display contents, and the devices they just need to display a part of content, not the whole contents.

Patwari, N in his paper *Relative location estimation in wireless sensor networks*[11] described: For some devices, no relationship of their have absolute coordinate information or not can affect estimate between devices and neighboring devices. The location estimation method is called "relative location" method, as the range estimates collected, remarkable between two devices which neither has absolute coordinate information.

In recent years, as mobile devices become more popular,

and smart mobile devices' relative location estimation method is richer. In Wang Tian's paper BlueCat: An Efficient Way for Relative Mobile Localization [12]. The work is identified if one mobile device user is located in the vicinities by other mobile users. The mobile device user can select Bluetooth as transfer data when general Bluetooth modules provided about 10 meters proximity range, which may enough for locate another mobile device users in the vicinity. And one mobile user also could "perceive" other users through multi-hop to communication.

Actually whatever Patwari N's research in 2003 and Wang Tian or other research [13] using similar theory in smart device, through complex calculation gets results not very accurate. In Patwari N's research, the RMS location errors range from 1.0m to 2.7m. This error range can be accepted in spacious place, but in narrow space like in the room, on the table or bed, the error range from 1.0m to 2.7m cannot be allowed. This paper adopted a new method of relative location has more applicable to some narrow space location for the *Contents Reconfiguration System*.

### III. CONTENTS SELF-RECONFIGURABLE SYSTEM

A complete *Contents Self-Reconfigurable System* contains a series processes. Firstly, the user can configure a system from of contents performance type: a two dimensional matrix, each device becomes one element in this matrix. The user inputs this matrix's column and row value in one device. For example, the user inputs two by four matrix type of system, the system can get the system type is two by four two dimensional matrix. After user according inputted column and row value to put the device in one area as Fig 1, the device can get its position and who surround it.



Fig. 1. Two by four matrix type *Contents Self-Reconfigurable*

System

In Fig.1 one device's screen written 'server' denotes this device is server device. The part of connection in this System is developed by Wi-Fi P2P. One solution of mobile devices connection establishment is Wi-Fi P2P devices establish P2P group[14] to communicates each other. One device implements Ad-Hoc functionality in P2P group which is referred as P2P Group Owner(P2P GO), and other devices is known as P2P clients.

#### 3.1. Device Matrix Initialization

The user inputs two values, i.e., rows and columns as user expects *Contents Self-Reconfigurable System* matrix model. After that, the user sets devices in the form of user's input.  $R$  represents the number of rows and  $C$  is the number of columns. On the other hand, traditional p2p network just saves node information as one-dimensional space, i.e., a list. But in order to better positioning device, the system according to user's input builds a two dimensional matrix model as  $S$ . Any element can be expressed by  $m$  rows and  $n$  columns, user expects device quantity ( $dq_e$ ) in equation (1).

$$dq_e = R \times C \quad (1)$$

The values of  $m$  and  $n$  are calculated as (2) and (3):

$$n = \sum_{i=0}^{dq_e} \frac{i}{C} \quad (2)$$

$$m = \sum_{i=0}^{dq_e} i - n \times C \quad (3)$$

Through equations (1), (2) and (3), according to user input, proposed system can identify using the location  $m$  and  $n$ , and the system can use the elements of matrix to represent one device in two dimensional space such as equation (4). The value of  $m$  and  $n$  adopt the rounding principle. For example, if  $m = 0.3$  in the system,  $m = 0$ . If  $n = 4.8$  in the system,  $n = 5$ .

$$S = \begin{bmatrix} M_{00} & \cdots & M_{0n} \\ \vdots & \ddots & \vdots \\ M_{m0} & \cdots & M_{mn} \end{bmatrix} \quad (4)$$

#### 3.2. Device Detection

Through *Device Matrix Initialization* and *Contents splitting*, proposed system constructs a matrix  $S$  with  $m$  rows and  $n$  columns, each element  $M_{ij}$  has device information. The  $M_{ij}$  has three attributes which are  $M_{ij}.ip$ ,  $M_{ij}.name$ , and  $M_{ij}.content$ :  $M_{ij}.ip$  is device IP of  $M_{ij}$ ,  $M_{ij}.name$  is device name of  $M_{ij}$ , and  $M_{ij}.content$  is the part of content  $C_{ij}$ . Also, content  $C$  is divided into  $|m * n|$  parts, each  $C_{ij}$  is the part of  $i$ 's row and  $j$ 's column of content  $C$ .

User catches up one arbitrarily device ( $Dev1$ ). Using a proximity sensor of the device, the system knows this device map with  $M_{xy}$  element in two dimensional matrix  $S$ . At this time, the status of matrix  $S$  is denoted by  $S_1$ . Moreover, the system generates a copy of  $M_{xy}(M'_{xy})$ . Then the system removes the information  $M_{xy}$  of  $Dev1$  and marks  $M_{xy}$ 's original position such as  $M_T$ . After user catches second device ( $Dev2$ ), the system also copies the information  $M_{vw}$  of  $Dev2$  to  $M'_{vw}$  such as equation(5) and (6). The status of matrix  $S_1$  becomes  $S_2$ .

$$S_1 = \begin{bmatrix} M_{00} & \dots & M_{0n} \\ \vdots & M_{xy} & \vdots \\ M_{m0} & \dots & M_{mn} \end{bmatrix} \quad (5)$$

$$S_2 = \begin{bmatrix} M_{00} & \dots & M_{0n} \\ \vdots & M_T & \vdots \\ M_{m0} & \dots & M_{mn} \end{bmatrix} \quad (6)$$

The system replaces the original position  $M_T$  and  $M_{vw}$  with  $M'_{vw}$  and  $M'_{xy}$  like equations (6) and (7), the status of matrix  $S$ ,  $S_2$  becomes  $S_3$ . As next step, the  $M'_{vw}$  and  $M'_{xy}$  in equation (7) are changed with  $M_{vw}$  and  $M_{xy}$  in equation (8). The status of matrix  $S_3$  becomes  $S_4$ . Finally, the system notifies the location of changed devices to related devices to change their contents.

$$S_3 = \begin{bmatrix} M_{00} & \dots & M_{0n} \\ \vdots & M'_{vw} & \vdots \\ M_{m0} & \dots & M_{mn} \end{bmatrix} \quad (7)$$

$$S_4 = \begin{bmatrix} M_{00} & \dots & M_{0n} \\ \vdots & M_{vw} & \vdots \\ M_{m0} & \dots & M_{mn} \end{bmatrix} \quad (8)$$

Related devices send related information to server device. The related information is two dimensional matrix  $S$ , as the input of Algorithm 1.

Through this algorithm the system updates the matrix, and detects position of changed devices and new position. According to system detection, the system sends notification message to  $Dev1$  and  $Dev2$ . The notification message is the position  $M_{xy}$  and  $M_{vw}$ .

---

ALGORITHM 1. Device Detection and Matrix Update Algorithm

---

Input	$S$ is two dimensional matrix with $m$ rows and $n$ columns, $M_{xy}$ and $M_{vw}$
Output	Updated matrix $S$
1	<b>if</b> $M_T = null$ <b>then</b>
2	$copy(M_{xy}, M'_{xy})$
3	$M_T \leftarrow M_{xy}$ .
4	<b>end if</b>
5	<b>for</b> $i = 0$ <b>to</b> $n$
6	<b>for</b> $j = 0$ <b>to</b> $m$
7	<b>if</b> $M_T \neq null$ <b>then</b>
8	$copy(M_{vw}, M'_{vw})$
9	$M_{vw} \leftarrow M_T$
10	$M'_{xy} \leftarrow M'_{vw}$
11	$notify(M_{vw}.IP, C_{xy})$
12	$notify(M_{xy}.IP, C_{vw})$
13	<b>end if</b>
14	<b>end for</b>
15	<b>end for</b>
16	<b>return</b> $S$

---

### 3.3. Content Splitting

After the system constructs new matrix  $S$ , the user can select specific content. This content will be divided into several parts according to the matrix  $S$  formed by *Contents Portioning* method. Using *Contents Shuffling* method in *Contents Splitting* module, the system sends segmentalized contents to all devices.

#### 3.3.1 Content Portioning

After *Device Matrix Initialization*, matrix  $S$  has been established. If the user selects content, the system will be obtained content's width and height such as  $O_w$  and  $O_h$ . The matrix  $S$  has  $m$  devices in rows and  $n$  devices in columns. In order to distribute original content, the system should divide the original contents into equal-sized contents. Given the width and height of segmentalized

content as  $d_w$  and  $d_h$ ,  $d_w = O_w/n$ ,  $d_h = O_h/m$ .

The system adopts several types of the content. The types include image, video, and etc. Each formatted contents need different portioning methods to divide the contents. However, in this paper, we propose a common solution to portion the contents. In proposed method, first step is to generate a vector with  $d_w$  and  $d_h$  such as  $vector(d_w, d_h)$ .

The system gives a start point ( $sp$ ) to  $vector$ . This a start position of original content. If the content needs dividing into several parts,  $sp$ 's value is defined as equation (9).

$$sp = \left( \sum_{j=0}^m j * d_w, \sum_{i=0}^n i * d_h \right) \quad (9)$$

The system makes some parts of the original content. The number of the parts is  $m \times n$ .

### 3.3.2 Content Shuffle

About contents data splitting process in the system, one method is server device splitting total contents to every client devices as TCS method(*Total Contents Splitting*) in this paper as Fig.2.

$NB$  is network bandwidth,  $Cfs$  is contents file capacity size, quote  $dq$  is means device quantity, assume  $T_{cs}$  is contents splitting time,  $T_{cs}$ 's equation is (10).

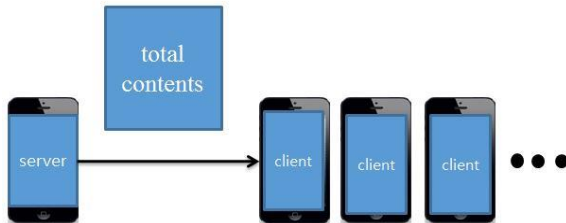


Fig. 2. TCS device contents splitting method

$$T_{cs} = \sum_{i=0}^{dq} \frac{Cfs}{NB} \quad (10)$$

In TCS method, the time of system using TCS method to transmit data is  $T_{TCS}$ . The notification message transfer between devices take almost negligible time, user main spend time on  $T_{cs}$ , so  $T_{TCS} \approx T_{cs}$ .

This research also has another method to split contents data. Before server device broadcasts contents data to other device, the system completes fragmented with total

contents. After this process, server device transfer specified content fragment to related device as Fig.3. This method is named FCS(*Fragmented Contents Splitting*) in this paper. After device detection, system also multicast contents change message to device as TCS but notification message's type is different. In FCS method, notification message is not vector but start point, the system multicasts new contents to instead of original contents.

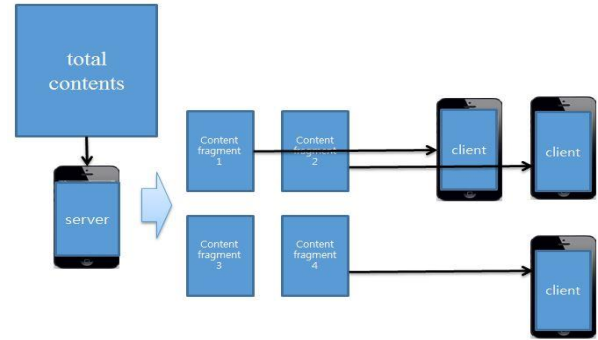


Fig. 3. FCS device contents splitting method

In method FCS,  $NB$  also network bandwidth,  $Cffs$  is fragment contents file capacity size.  $dq$  is device quantity and  $T_{fs \rightarrow ffs}$  is process spent time of total contents to be fragment contents in devices.  $T'_{cs}$  is contents splitting time in this method.  $T'_{cs}$ 's equation as (11).

$$T'_{cs} = T_{fs \rightarrow ffs} + \sum_{i=0}^{dq} \frac{Cffs}{NB} \quad (11)$$

The system multicast new contents to device spent time as  $T_{nc}$  and express by (12)

$$T_{nc} = 2 * \frac{Cffs}{NB} \quad (12)$$

Include contents data splitting time and the user dose  $N$  times device exchange action time, system using FCS method the total time is  $T_{FCS}$ ,  $T_{FCS}$ 's equation as (13).

$$T_{FCS} = T'_{cs} + N * T_{nc} \quad (13)$$

## IV. SYSTEM DEVELOPMENT

This system constitutes one by one devices and these

devices' position is not planted. The user can exchange some devices position and the system can get user's action and do some response to make display back to its original state. Based on the above request, we need develop a n application to support devices establish connection.

Before the data exchanging between devices, in P2P connection, the device needs to discover each other first. This system used WiFi P2P(a.k.a.Wi-Fi Direct)[15] to building network, so the user needs turn on device's WiFi module to discover other device which nearby user's device. The application of developed for this research made adds other device's information to the device list. And then user selects one device to connect. Then user's device become to server device, which has server function. The device with server function is called 'server device' and the invited device is called as 'client device'. The server device assigns IP addresses through DHCP.

Server device has all client devices' IPs and adds their IPs to other list and this list is different from above talked the discovered device list. And after the system gets column and row value of user inputted, this list constructed a matrix. This whole process is the part of *Device Matrix Initialization* in Fig 4. Before of this process only have original *List-mode* to store device information. After this process, two dimensional device position model has been established.

After the system constructed matrix, the user can select content, and this content will be divided according the matrix from by *Contents Portioning* method. Then the system using *Contents Shuffling* method in *Content Splitting module* to deliver divided the contents into devices as the part of *Content Splitting*.

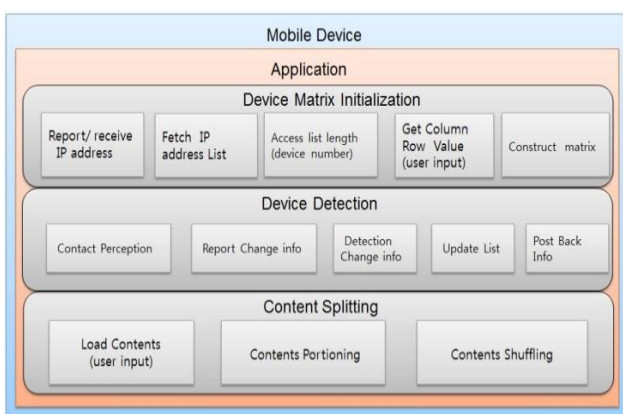


Fig. 4. System development architecture.

This system achieved by JAVA language and developed a mobile device application in Android platform, which is an open-source soft stack for mobile devices and

compares with other platforms, mobile device base on Android platform is more popularization.

After devices were established the connection between other devices, User can select contents from device' storage or device' camera. Then contents are splitting to devices as Fig 5.



Fig. 5. User takes photo and splitting to devices

In Fig 6, system constructs two by two contents configuration, because one device using for server device and in order to present the device information dynamic changing visualization, this device is closed contents, just presents devices information change.

User catches arbitrarily two devices as marked device A and device B laid low its original position on the table. User changes their devices' physical position.

The system detects device's physical position changed and through *Self-Reconfigurable system* completes contents shuffling as part 4 in Fig.6. Fig.7 showed contrast of devices position change before and after. After device A, B and C all changed their position, the system reconfigured their contents.



Fig. 6. Two by two contents configuration





Fig. 7. Contrast of devices position change before and after

## V. SYSTEM EVALUATION AND DISCUSSIONS

This section through some experiment to evaluate this research a *Self-Reconfigurable System of Contents among Smart Device*'s performance, and then the system's results of the evaluation are analyzed.

### 5.1 Performance Criterion

To evaluate a *Self-Reconfigurable System of Contents*' performance criterion from two aspects as the maximum device quantity support and contents data operation spent time.

#### 5.1.1 Maximum Devices Support

In theory, this system has not set any limits when the user assigns two dimensional matrix types as row value and column value to establish arbitrary *magnitudes Self-Reconfigurable System of contents*. But this system is using WiFi P2P as data transmission technique. WiFi P2P development section is not clear in one WiFi P2P group how many devices can connect at the same time because if one device become a group owner and it just like a router assigns addresses to other device. Wi-Fi Alliance® gives an answer that is the number of devices in a Wi-Fi Direct-certified group network, which is expected to be smaller than the number supported by traditional standalone access points intended for consumer use the most common One IEEE 802.11 AP[16] can typically communicate with 30 client systems, this system also can support as many as approach 30 devices, some results are inferred and some research[17] and also make some experiment and predictions to prove.

#### 5.1.2 Contents Data Transmit

The System can support as many as near 30 devices at same time but in view of test device quantity limitation, in this research only have experiment as from 2 to 8 devices connection in system. Through calculates system

operation time from 2 to 8 devices in the system, and the results can be estimated.

In Wi-Fi Alliance's opinion, latest Wi-Fi P2P technology supports typical Wi-Fi, which can be as high as 250 Mbps[16], but in our test SAMSUNG S3(SHV-E210S) device sends transfer data to SAMSUNG S4(SHV-E470S) device because is not newest hardware device and various factors, the result is obtained as 1.45 Mbps.

### 5.2 Analysis and Discussions

Firstly, a group contents files size as 25.8 MB image contents and 34.3 MB mp4 video contents are selected and using TCS method to contents portioning. The test respectively records 2 devices to 8 devices' seven values figure out by seven values from 2 devices to 8 devices as equation (10). Compare two kinds of time values as Fig.7.

In Fig.8 blue bar expresses the total time of system broadcast data to all devices by the actual record. And orange bar is an equation of using TCS method estimated value. According to compare result of two values, though have some errors with two values, their errors is acceptable and using TCS method estimates the situation as more than eight devices is feasible.

The test also using FCS method to broadcast contents data to device' s actual record time compare with FCS estimated value of according the equation (11). In this experiment also using a 25.8 MB and 34.3 MB contents file. Firstly, using manual to test  $T_{fs \rightarrow ffs}$  value in various situations of device quantity and  $T_{fs \rightarrow ffs}$  value plus total contents data broadcast time can estimate  $T'_{cs}$  and its contents splitting time in FCS method. In this experiment SAMSUNG S4(SHV-E470S) device has CPU Clock Speed 1.5GHz and RAM 2GB, Trough tested SAMSUNG S4(SHV-E470S) handling time be record in Table 1.  $T_{fs \rightarrow ffs}$  (image) records image content fragmentation time and  $T_{fs \rightarrow ffs}$  (video) records video content fragmentation time.

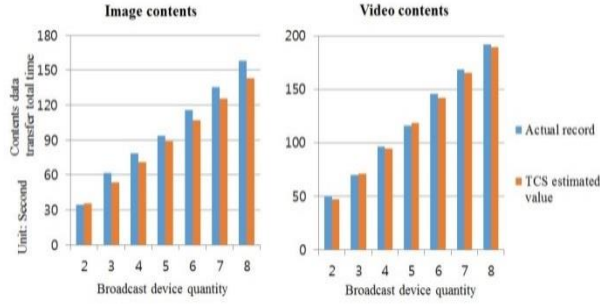


Fig. 8. Actual total time of system broadcasts contents data compare TCS method estimated value

Integrates the result of Table.1 can calculate  $T'_{cs}$ 's value by equation (11) and  $T'_{cs}$ 's result as Fig.8. The image content fragmentation  $Cfs$  nearly equals  $Cffs * dq$  in equation (11). But in video content fragmentation, because this research using FFmpeg algorithm[18] portioning video contents to fragmentation, each  $Cffs$  value has difference between each other, and  $Cffs_1 + Cffs_2 + \dots + Cffs_{dq}$ 's value greater than  $Cfs$  value about 43.1% increase. So 34.3 MB size video contents after contents fragmentation become to 49.08 Mb size fragments total. In Fig.9 blue bar expresses the total time of system broadcasts data to all devices by actual record. Orange bar is equation of using FCS method estimated value  $T'_{cs}$ . For some reason the result of actual record also greater than  $T'_{cs}$  estimated value but still can predict the tendency of next step.

Table.1. SAMSUNG S4(SHV-E470S) device FCS *Contents Portioning* processing time

Device quantity	$T_{fs \rightarrow ffs}$ (image)	$T_{fs \rightarrow ffs}$ (video)
2	2.549s	5.248s
3	3.556s	8.651s
4	3.671s	12.364s
5	6.745s	15.826s
6	8.160s	19.882s
7	10.534s	24.571s
8	12.203s	30.295s

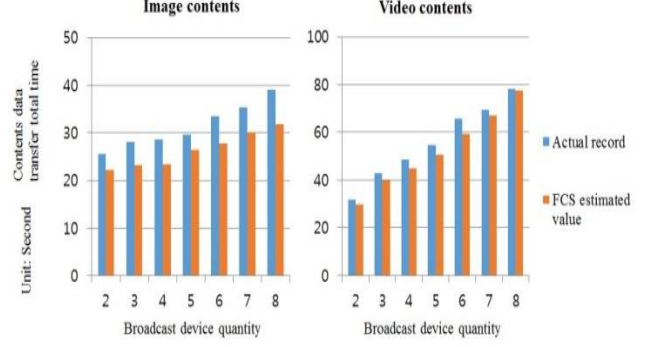


Fig. 9. Actual total time of system broadcasts data compare FCS method estimated value

Compare TCS and FCS method according the result as Fig.8 and Fig.9. FCS method spent less time than TCS in process of *Contents Portioning*. But integrates the process of *Contents Shuffle*, if the System needs to process *Contents Shuffle* frequently, the test result is FCS method specifically suites the environment of has a lot of devices and don't change devices position frequently.

When user exchanges two devices position continuously, the system also needs self-reconfigure continuously. Under certain conditions, as signal transmission failure or wireless interrupted. The Fig.10 shows the test result of data transmission success rate. This test using thread to simulate after system detected two devices position continuously transmits a message between devices. In every three seconds, this thread lets the device do one time contents self-reconfiguration like they changed their position. Until the finish, 50 times or 100 times test has been accurately complete. *Contents Self-Reconfiguration* tests range from 2 devices to 8 devices. Blue and orange represent test 50 times and 100 times.

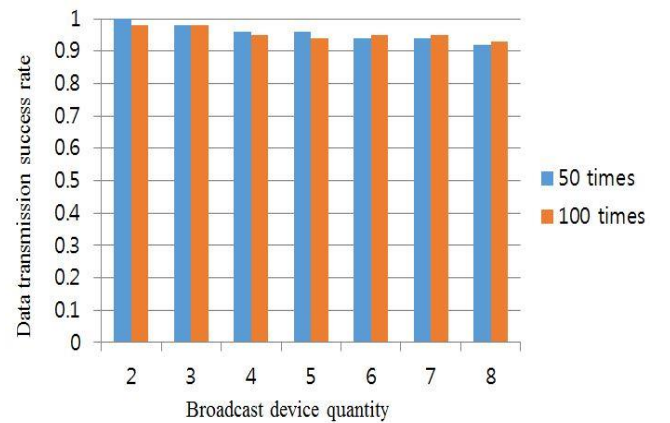


Fig. 10. Data transmission success rate



## VI.CONCLUSIONS

This research developed an application to achieved proposed *Self-Reconfigurable System of Contents*. This system can accurate detect device's physical position's changes.

Some techniques and methods are proposed in this *A Self-Reconfigurable System of Contents among Smart Devices*, but in here some techniques and methods not only apply to *Self-Reconfigurable System of contents*. Some communication protocols and mathematical modeling methods are propose. These methods can also be used in other Internet of Things fields, such as Drones Navigation, Smart Home, Informational City management, etc.

### Acknowledgement

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2012M3C4A7032185))

### REFERENCES

- [1] L. Chittaro, "Visualizing information on mobile devices," in *Proceedings of the Computer Published by the IEEE Computer Society*, pp. 40-45, Mar, 2006.
- [2] R. Borovoy, B. Knep "Junkyard jumbotron," MIT Center for Civic Media. Apr, 2011.
- [3] M. Li and L. Kobbelt "Dynamic Tiling Display: Building an Interactive Display Surface using Multiple Mobile Devices," in *Proceedings of IEEE International Conference on Multimedia and Expo(ICME)*, pp. 980-985, 2011.
- [4] D. Baur, S. Boring, S. Feiner, "Virtual projection: exploring optical projection as a metaphor for multi-device interaction" in *Proceedings of the SIGCHI Conference on Human*, pp. 1693-1702, 2012.
- [5] J. Tanaka and T. Ohta. "Multi-Smartphone-Display Applications that React Dynamically to Display's Layout," *IPSJ Interaction 2012* , pp. 101, 2012.
- [6] H. S. Nielsen, M. P. Olsen, M. B. Skov, J. Kjeldskov "JuxtaPinch: exploring multi-device interaction in collocated photo sharing," in *Proceedings of Mobile HCI'14 Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services* pp.183-192, Sep, 2014.
- [7] S. Boring, M. Jurmu and Butz, "A Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays, " in *Proceedings of the 21st Annual Conference of the Australian Computer- Human Interaction Special Interest Group*, NewYork, pp. 161-168, 2009.
- [8] A. Lucero, J. Holopainen, and T. Jokela, "collaborative use of mobile phones for photo sharing, " in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.1787-1796, 2011.
- [9] N. Marquardt, K. Hinckley, S. Greenberg, "Cross-device interaction via micro-mobility and f-formations," in *Proceedings of the 25th annual ACM symposium on UIST '12*, pp.13-22, 2012.
- [10] The Samsung Galaxy Note 3 Multi, <http://blog.gsmarena.com/the-samsung-galaxy-note-3-multi-vision>
- [11] N. Patwari, A.O. Hero, M. Perkins, N.S Correal, R.J. O'Dea. "Relative location estimation in wireless sensor networks," in *Proceedings of IEEE Transactions on*, pp. 2137-2148, 2003.
- [12] W. Tian, J. Weijia, S. Ping, "BlueCat: An Efficient Way for Relative Mobile Localization, " *2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 209-215, 2012.
- [13] S. Li, Y. Lou, B. Liu, "Bluetooth aided mobile phone localization: a nonlinear neural circuit approach, " *ACM Transactions on Embedded Computing Systems (TECS)* , No. 78, 2014.
- [14] D. Mur, A. Saavedra and P. Serrano "Device to device communications with WiFi Direct: overview and experimentation, " in *Proceedings of Wireless Communications, IEEE* , pp. 96-104, 2013.
- [15] Wi-Fi Direct, <http://www.wi-fi.org/discover-wi-fi/wi-fi-direct>
- [16] How many devices can connect? <http://www.wi-fi.org/>
- [17] M. Conti, F. Delmastro, G. Minutiello, Paris. R, "Experimenting opportunistic networks with WiFi Direct, " in *Proceedings of Wireless Days (WD), 2013 IFIP*, pp. 1-6 , 2013

[18] FFmpeg, <https://www.ffmpeg.org/>

mobile environment. He is currently interested in Community Detection on evolutionary network and Social Network Analysis.



**Sangwook Kim** received B.S degree from Kyungpook National University, Korea, in 1979. He received M.S. degree from Seoul National University, Korea, in 1981. He received Ph.D. in Computer Science and Engineering from Seoul National University, Korea, in 1989. He is currently Professor of School of Computer Science and Engineering at Kyungpook National University. He is interested in mobile media systems, media contents, human computer interaction, social network analysis and Internet-of-Things.

#### AUTHORS



**Ren Hao** received the B.S. degree in Information and Communication from Catholic University of Daegu, Korea, in 2013. He received M.S. degree Kyungpook National University, Korea, in 2015. He is interested in mobile media systems, media contents, human computer interaction, social network analysis and Internet-of-Things.



**Paul Kim** received a M.S degree from Kyungpook National University, Daegu, Korea in 2011. His research area is Context-Aware Platform and Social Network Computing. The goal of these studies are to research technology of human-oriented and human-centric in