Mobile Cloud Computing for Vehicle Servers with Delay Tolerant Network Protocol

Masaki Otomo[#1], Noriki Uchida[*2], Yoshitaka Shibata[#3]

[#1,3]*Iwate Prefectural University,*
Iwate, Japan
[1]g231n201@s.iwate-pu.ac.jp
[3]shibata@iwate-pu.ac.jp
[#2]*Fukuoka institute of Technology,*
Fukuoka, Japan
[2]n-uchida@fit.ac.jp

***Abstract*— Serious disasters have been annually occurred in Japan. Particularly, in the Great East Japan Earthquake that occurred on March 11, 2011, the communication infrastructure was seriously damaged over the wide area. Due to such situation, more resilient network infrastructure that does not stop under challenged communication environment is required. In this paper, we propose a disaster information system that can flexibly deal with critical network connectivity. In order to achieve this purpose, we develop a dynamic allocation of server resources in accordance with the load change on the system so that it is possible to take full advantage of the server and network resources in the disaster areas. Also, by introducing mobile cloud computing and DTN protocols, our system can share widely disaster information even if the communication breakdown occurred in the area.**

***Keywords*— Disaster Information System, Mobile Cloud Computing, System Virtualization, Delay Tolerant Networking, Load Balancing**

1. INTRODUCTION

There are many disasters in Japan, such as earthquake, tsunami, typhoon due to its inherit geological condition. In particular, the Great East Japan Earthquake caused huge damages on March 11, 2011 [9]. Many Information networks infrastructures were destroyed and their traffics were congested seriously. In order to deal with the anticipated large scale disasters such as Nankai Trough earthquake and Tokai earthquake, Web applications which confirm safety of victims and urge decision making on disaster situations have been developed.

On the other hand, cloud computing technology which has expandability of computer resources is getting popular for various businesses because it is easy and efficient to introduce computing resources and their functions. Using cloud computing services, operations and maintenance of hardware and software can be carried out at a data center in a lump. Since the users do not need to introduce new servers for businesses physically, the operations and maintenance cost has been reduced greatly. Furthermore, by introducing virtual computing technology, users can run own private cloud computing. Thus, there are many advantages to use the cloud computing as Web services.

So far we have made progressed the researches in disaster information sharing systems which consider mobile environment. In these systems, the network states are monitored at background. If the network accesses to Internet are difficult, then the disaster information is locally stored in mobile relay stations. After moving to the location where Internet environment can be established, the stored disaster information can be transmitted to the objective disaster information server. However, this system does not consider rapid load changes of the servers and networks. In addition, the servers and networks may be failed. Therefore, when large disasters occur, we need to cope with concentration of traffic loads in networks, server systems and their failures caused by disasters [10].

In this research, we introduce a mobile cloud computing disaster information system for large scale disasters which is able to keep continuous operations even if the network environment is unstable or disconnected. The computing resources can also be dynamically provided to different user groups such as local governments as required to maximize the physical resource utilization. Moreover, this system does not only provide data transmission by introducing DTN function but the disaster information system also collects data more quickly and shares the service by introducing mobile cloud computing in some areas where communication networks are unstable or disconnected.

2. RELATED WORKS

There are several researches concerned with disaster information systems. Some of the previous systems have been researched to cope with network environment with large delay and link disconnection which happens frequently in network environment [1]. In this system, data can be transmitted if the disaster server can connect to the network by monitoring the network state. The disaster information can be also shared with multiple servers of different organization such as some local governments. However, this system does not consider rapid traffic changes of networks and if the servers of the local government are down by external factors such as tsunami, those systems cannot be served.

In the other previous system, disaster information can be visually shown on the display combined by GIS systems [2]. The users obtain what kind of disaster information is registered on the system by using various icons and figures. By operating 'seek bar' on the window, the registered disaster information can be displayed in temporal orders as replay operation on the video window.

3. SYSTEM CONFIGURATION

3.1. Proposed System Concept

Figure 1 shows a network system of our proposed disaster information system. There are two types of cloud computing which are called Global Disaster Cloud (GDC) and Local Disaster Cloud (LDC).

GDC is a central cloud server located on Internet and integrates disaster information of LDC in local areas. LDC is a mobile cloud computing which is carried on a vehicle and circulated around local government offices, evacuation places, community centers and public places where Temporary Servers (TSs) exit. TSs store local disaster information of those places by them immediately after a disaster occurs.

After the occurrence of a disaster, when communication through Internet is available, those TS can share its disaster information with GDC directly. When TS cannot use Internet to communicate, the vehicles with LDC goes around the disaster areas and collects the local disaster information from TSs until the communication network is recovered.

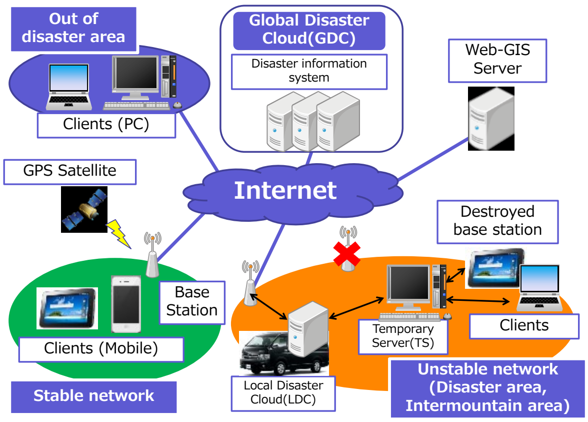

Fig. 1 System Configuration

3.2. Delay Tolerant Network

Delay/Disruption Tolerant Network (DTN) [6][7] is defined as a protocol of communication which achieves reliable data transmission in critical network environment such as disconnected or intermittent connectivity of Internet by generating bundle layer which is located between Application and Presentation Layers.

Transmissions of packets on unstable networks tend to take a long delay time and the network might be disconnected on the way where a node transmits packets to the destination. Therefore, a mechanism in which the packets are transmitted if network capability is enough to send and if network capability is not enough the packets are stored is required. In order to implement this mechanism, we employ DTN to achieve Store and Forward functions in such environment.

Here, two cases of network conditions including normal and challenged network condition are as follows.

1) In case of normal network condition:

Figure 2 shows system behaviour in normal network condition where the network has capability to transmit data. In this case, GDC collects data of disaster information from Temporary Servers and stores it to the database server. Therefore, many people from the stricken disaster areas will access to the server to obtain disaster information. Moreover, the server resources of GDC are controlled depending on access rates from clients and traffic rates caused by increasing the number of virtual CPUs and memories dynamically.

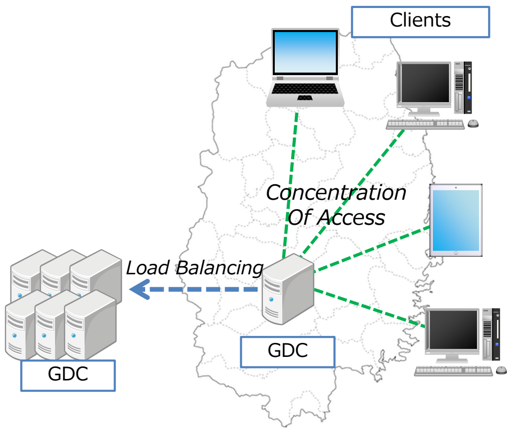

Fig. 2 Operation in lack of Computer Resources

2) In case of challenged network condition:

Figure 3 shows system behavior in challenged network condition where the network is unstable to connect to Internet. Different wireless network devices such as 3G/LTE, Wi-MAX, Wi-Fi and satellite networks are adopted to organize cognitive wireless networks. After disasters occur immediately, mobile LDC vehicles with one of the cognitive wireless networks perambulate around disaster areas and approach to each Temporary Server which preserves disaster information with each area such as disaster status information, resident safety information and so on. Then LDC receives those stored data from Temporary Servers by DTN protocols in the areas. After providing their information, all of the data in GDC and TS, LDC are synchronized to guarantee consistency of information.

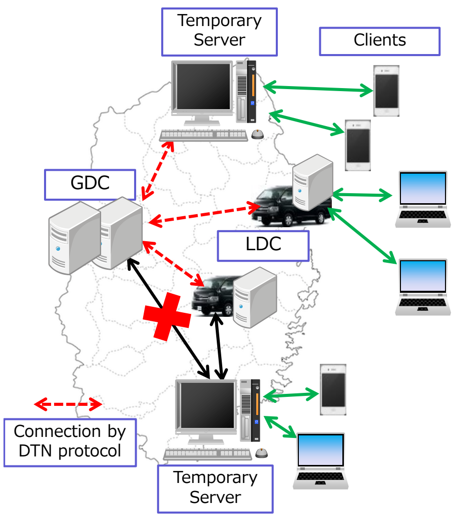

Fig. 3 Challenged Networks Environment

4. SYSTEM ARCHITECTURE

Figure 4 shows architecture of our proposed system. By assigning the required number of Virtual Machines (VMs) to each cloud server, Disaster Information system will be provided even if it is under critical network connectivity. Each server is consisted of Monitoring Module (MM), Resource Management Module (RMM), VM Control Module (VMCM) and DTN Transport Module (DTM). MM monitors resource utilization rates of CPU and memories of VM and sends them to RMM. RMM controls VM resource assignments based on the rates and sends operation commands to VMCM such as Start, Stop, Addition and Reduction of VMs. DTM manages sending, receiving and storing data by DTN protocols.

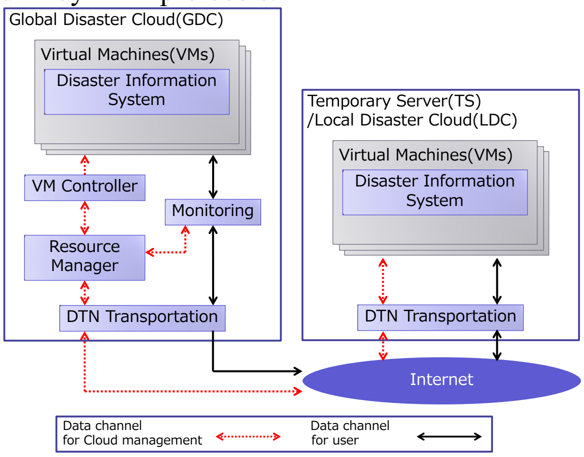

Fig. 4 Flow of Increasing Server Resource Process

5. ALGORITHM OF RESOURCE CONTROL

Figure 5 shows a flow of server processes for resources addition [3][4][5]. Firstly, the resource utilization rates of VMs are monitored by MM. Secondary, the numbers of clocks or CPUs resources and their clock frequency are controlled by referring to the utilization rates of CPU and memories. When the CPU utilization rate is more than X [%], then the CPU clock frequency is increased until U [GHz] or the number of CPU resources is increased. Also when CPU utilization rate is less than X [%] and memory utilization rate is more than Y [%], the volume of memory is increased until V [Mbytes]. Finally the process of load balancing for each resource is executed depending on the change of VM specification.

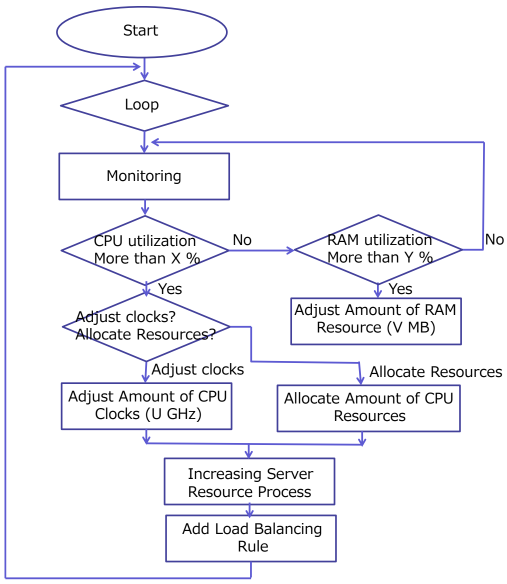

Fig. 5 Flow of Increasing Server Resource Process

On the other hand, Figure 6 shows a flow of server processes for resource subtraction [3][4][5]. When both CPUs and memories are less than each Z [%] and W [%], then one of VMs is stopped. After that, the process to decrease the server resources is executed. Finally the function of load balancing is also added depending on changes of VM specification.

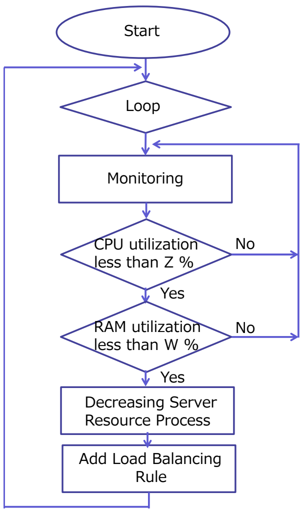

Fig. 6 Flow of Decreasing Server Resource Process

6. PROTOTYPE SYSTEM

6.1. Prototype System Configuration

In order to verify usefulness of our proposed system, a prototype system is constructed to evaluate its functionality and performance. Figure 7 shows a system configuration of the proposed prototype. In the prototype system, Management Server manages the whole cloud system and Host provides VM resources. As LDC function, mobile computers are suitable to carry them in vehicles. The hardware and software specification of GDC, TS and LDC is shown in the TABLE 1.

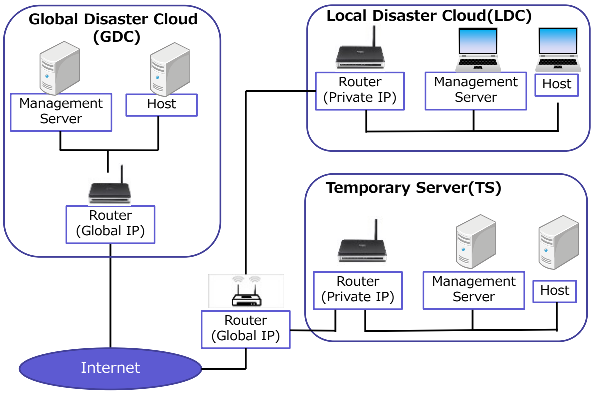

Fig. 7 Prototype System

TABLE 1
Hardware Specification of Servers

	GDC, TS	LDC
CPU	Intel(R) Core(TM) i3-3240 Processor (3M Cache, 3.40 GHz)	Intel(R) Core(TM) i7-4712MQ Processor (6M Cache, 2.30 GHz)
RAM	4GB	16GB
HDD	500GB	1TB

6.2. CloudStack

As cloud computing system environment, we applied CloudStack [8] which is one of open sources and provides Infrastructure as a Service (IaaS) to construct in publics or in private cloud computing systems such as Amazon EC2. CloudStack is used by many organizations because of its excellent GUI and easy operations. Since load balancers and firewall functions are installed as internal architecture, more functional expansions could be possible. In our system, the dynamic resource control function of VM which depends on the resource utilization rate is implemented using CloudStack API on Linux OS environment. On the other hand, Figure 6 shows a flow of server processes for resource subtraction [3][4][5]. When both CPUs and memories are less than each Z [%] and W [%], then one of VMs is stopped. After that, the process to decrease the server resources is executed. Finally the function of load balancing is also added depending on changes of VM specification.

7. PERFORMANCE EVALUATION

We have already evaluated the algorithm of VMs for resource allocation systems at a previous paper [11]. Therefore, in this paper, we ascertain reliability of data transmission in concerned with DTN protocols. As a DTN protocol, DTN2 is used [7]. In order to generate infrastructure in disasters, it is assumed that LDC, TS and GDC exist in disaster areas and communicate with each other to send data of disaster information. 7.1 shows as a series of the server flows which would be the case of network failure or the enable condition that we can use networks in disaster situations. In addition, we set environment for a preliminary experiment as 7.2. Finally, the quantity of data stored in each server is showed as 7.3.

7.1. A series of flows in each server

After the occurrence of disaster, the LDC server traverses the disaster area and transmits disaster information by relying from TS which is in a shelter to GDC. It is assumed that the network to Internet in the shelter is disconnected and the vehicle with LDC carries the disaster information from TS using DTN. As network conditions on traversing in the disaster areas, connection and disconnection periods are set. Figure 8 shows the traverse of LDC to TS and GDC.

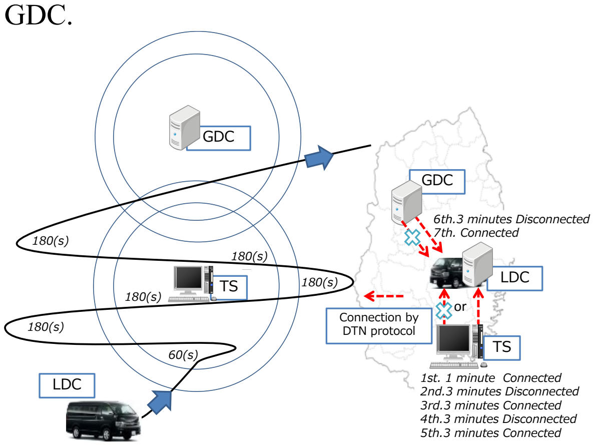

Fig. 8 Perambulation of LDC between each server

Firstly, the vehicle with LDC traverses in the disaster area and receives the disaster information on TS in the shelter using DTN for 60 sec. ("Connected"). Then, the vehicle releases from the shelter for 180 sec. ("Disconnected").

After that the vehicle comes back to the shelter and again receives the disaster information for 180 sec. ("Connected"). In this procedure, "Connected" or "Disconnected" between the LDC and TS servers are repeated until the 6th step in Figure 8 in this experiment.

From the 7th step, the LDC vehicle traverses to Headquarter where the Internet connection to the GDC can be available and transmits all of the disaster information from the LDC server to the GDC server. Thus, in the procedure form the 1st step through to the 7th step, the network communication can be guaranteed even though the servers are in challenged communication environment.

7.2. Environment for Preliminary Experiment

In order to ascertain reliability of data transmission in our system, we use a prototype system as Fig. 9 follows.

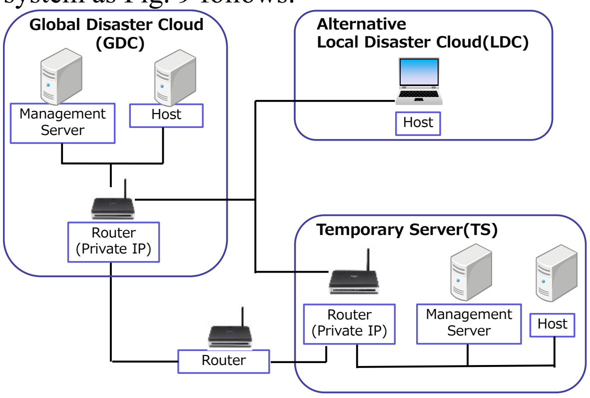

Fig. 9 Environment for Preliminary Experiment

Each server locates on local network environment. The vehicle of LDC traverses TS or GDC and collects the data files of the disaster information during 1st to 7th steps with wireless environment.

Reliability of data transmission in DTN protocols differs depending on the distance between each server. Figure 10 shows communication areas where the data can be transmitted between LDC and TS.

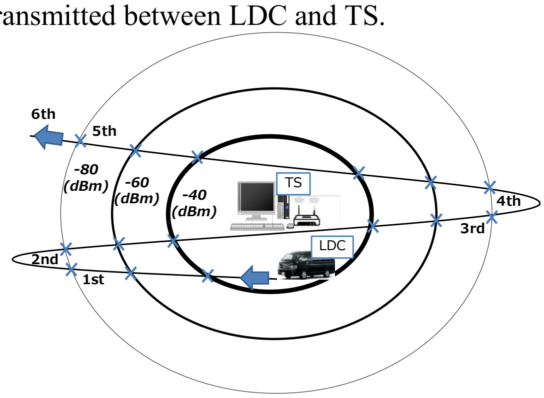

Fig. 10 Transmission Area in TS

At the area where signal strength is more than

-40dBm, it is assumed that the LDC vehicle can collect data from TS sufficiently and the network condition is enough to send data between each the server. Along with this, the LDC vehicle can get high throughput during it receives data.

At the area where signal strength is between -40dBm to -60dBm, however the LDC vehicle collects data depending on the network conditions, some data is not transmitted because of the conditions.

Between -60dBm to -80dBm, the network connectivity is getting worse by approaching the outside. If the LDC vehicle goes over -80dBm, it cannot communicate with TS at all. In these situations, such environment is made prospectively.

Finally, the LDC vehicle transmits the collected data to GDC in the 7th step. Figure 11 shows the communication areas where data is transmitted between LDC and GDC.

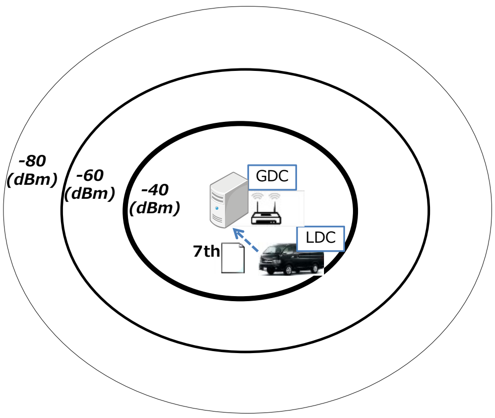

Fig. 11 Transmission Area in GDC

The LDC vehicle approaches to the GDC until can transmit the collected data to the GDC server without any interference on networks at the 7th step. Through these steps, all of the data can be transmitted from the TS server to the GDC server. The data is stored in the GDC server so that be provided to clients who obtain the disaster information.

7.3. Results in proposed preliminary experiment

The results in the preliminary experiment are shown as Figure 12 to Figure 15. The data size of the disaster information to be transmitted between TS to the LDC vehicle and the GDC server is 1Mbyte on every 1 second. The buffer size of the DTN bundle layers on each server is large and enough to store the transmitted data. This means that the minimum network transmission speed requires 8 Mbps.

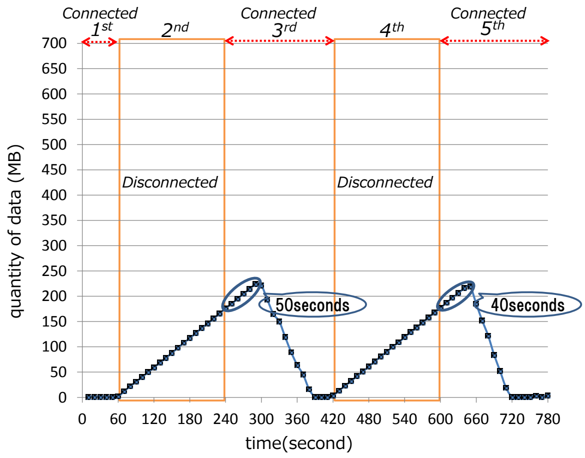

Fig. 12 Data Quantity during 1st to 5th Step in TS

Figure 12 shows the result of the variation in quantity of the transmitted data which are stored in the DTN buffer of TS. In the scenario of the 1st step, because the network condition between TS and LDC was "Connected", the data on the DTN bundle layer in TS were immediately transmitted to the DTN buffer in LDC, the quantity of the data was 0 for this period. In the network area, the throughput value to transmit data from TS to LDC was 34.91 Mbps.

During 60 to 240 sec. on the 2nd step, because the network condition was "Disconnected", the data in the TS were accumulated in the DTN buffer.

During 240 to 420 sec. on the 3rd step, although the network condition was "Connected", the data were not transmitted for 50 seconds. This is due to the time delay of the restarting process transmission on DTN2 in which the source host took time to find the destination host. In addition, another reason except the redundancy of system configuration is the electric field strength. When the vehicle of LDC connected to the network on TS at the 3rd step, the throughput value was 7 to 12 Mbps at -80dBm. This means the network condition was intermittent because the condition needs more than 8 Mbps to send data. After the 50 seconds, those stored data in the DTN buffer were sent to the LDC with the maximum throughput.

During 420sec to 600sec. on the 4th step, the same result was repeated as the 2nd step.

Finally, during 600sec to 780sec. on the 5th step, the same result was repeated as the 3rd step.

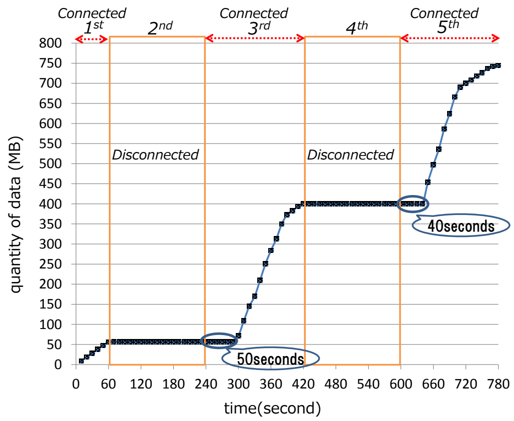

Fig. 13 Data Quantity of 1st to 5th Step in LDC

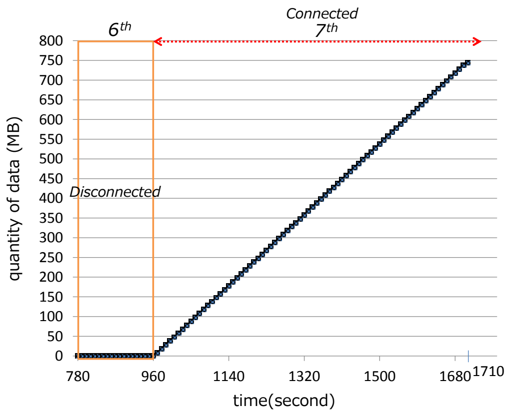

Fig. 14 Data Quantity of 6th to 7th Step in GDC

Figure 13 also shows the variation in quantity of the received data from TS which are stored in the DTN buffer of LDC.

During 0 sec to 60 sec. on the 1st step, the data from TS were received and be accumulated increasingly in the DTN buffer of LDC.

During 60 to 240 sec. on the 2nd step, because the network condition was "Disconnected", the data transmissions from TS were stopped and the accumulated data of 60 Mbytes were maintained.

During 240 to 420 sec. on the 3rd step, although the network condition was "Connected", the data from the TS were not transmitted for 50 seconds. This is due to the same reason of TS as shown Figure 12. After 50 seconds, those stored data in DTN buffer were received from TS with the maximum throughput after 290 seconds.

420sec to 600 sec. on the 4th step, the same result was repeated as the 2nd step and 600 sec to 780 sec. on the 5th step, the same result was repeated as the 3rd step.

After 780 seconds, when the iteration in the 5th step finished, the vehicle went out from the TS transmission area and went into the GDC transmission area at 960 seconds. In this transmission area, GDC could receive all of the accumulated data, 744 Mbytes from LDC with the throughput of more 1 Mbyte per second (8 Mbps) as shown in Figure 14.

Figure 15 shows the change of each data quantity of LDC at the 6th step and the 7th step. Along with Figure 14, each data quantity in the DTN buffer of LDC was decreasing 1MB per second. Therefore, these processes were completed at 1710 seconds.

Thus, all of the data generated in TS could be uploaded to the GDC server via LDC even though the network environment is disconnected in the disaster.

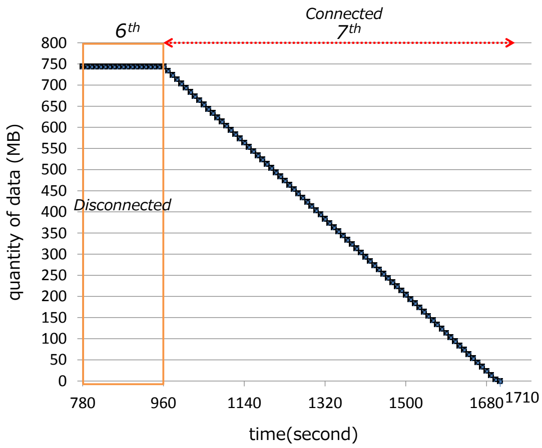

Fig. 15 Data Quantity of 6th to 7th Step in LDC

Discussion

In the preliminary experiment result, there is one issue. It is non-arrival of data from TS to LDC during the 1st to 5th step. Even though the network is changed to the enabled condition from the disabled condition, TS didn't transmit data to LDC soon and stored it to own DTN buffer for 40 to 50 seconds.

Conclusion and Future Work

In this paper, we proposed a mobile cloud disaster information system that can flexibly deal with critical network connectivity based on DTN. By providing this, disaster information collected will be provided for victims in the stricken areas after a disaster occurs immediately.

In addition, to use DTN protocols means that data can be stored locally in communication networks and the data is automatically transmitted to servers in the area where recovered networks get available, eventually the data transmission can be attained on the any network conditions.

As some future researches, we are going to implement and evaluate server resource control functions, data synchronization functions between cloud computing and reliable transmission functions in DTN with the transmission speed improved from TS to LDC.

Acknowledgment

The research was supported by SCOPE (Strategic Information and Communications R & D Promotion Program) Grant Number 142303010 by Ministry of Internal Affairs and Communication in Japan.

REFERENCES

[1] Y. Sasaki, and Y. Shibata, Construction of Distributed Disaster Information System in consideration of the mobile communication environment, The Institute of Electronics, Information and Communication Engineers (IEICE), "S-65"-"S-66" (2011)

[2] Y. Sasaki, and Y. Shibata, Construction of Disaster information system that enables the display time series uniform, Information Processing Society of Japan (IPSJ), "3-427"-"3-428" (2010)

[3] Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve and Alla Segal, “Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment”, 2009 IEEE International Conference on e-Business Engineering

[4] Gihun Jung, Kwang Mong Sim, “Agent-based Adaptive Resource Allocation on the Cloud Computing Environment”, 2011 International Conference on Parallel Processing Workshops

[5] Zhang Zhang, Jizhong Han, Bo Li, Wei Zhou, Dan Meng, “Lynn: A Multi-Dimensional Dynamic Resource Management System for Distributed Applications in Clouds”, 2013 International Conference on Cloud and Service Computing

[6] Delay Tolerant Networking Research Group,<http://www.dtnrg.org/wiki/Home>

[7] Delay Tolerant Networking Research Group “DTN2 Documentation”, <http://www.dtnrg.org/wiki/Dtn2Documentation>

[8] CloudStack User Group, <http://cloudstack.jp/>

[9] Ministry of Internal Affairs and Communications “Situation of information and communication in the Great East Japan Earthquake”, <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h23/pdf/n0010000.pdf> 2011

[10] Ministry of Internal Affairs and Communications “Enhance of disaster tolerance in communication”, <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/html/nc134210.html> 2012

[11] Yosuke Kikuchi, Yoshitaka Shibata, “Mobile Cloud Computing for Distributed Disaster Information System using in Challenged Communication Environment”, The 7th International Workshop on Disaster and Emergency Information Network System, (IWDENS2015), pp.512-517, Mar. 2015