ADAPTIVE COMMUNICATION MECHANISMS OF RESPONSIBLE SERVERS FOR DISTRIBUTED CARE ROBOT SYSTEMS

Satoru Nakayama[1], Miyuki Nakano[2], Midori Sugaya[1]

Shibaura Institute of Technology [1]

Advanced Institute of Industrial Technology [2]

MA15058@shibaura-it.ac.jp

ABSTRACT Robotic distributed systems have a great potential in many novel applications. These robots are connected to the wireless and network easily and offer variable solutions to many application areas such as nursing assistance, medical care, life support etc. We are developing "Distributed Care Robot System" that multiple robots working with patients in hospitals, or elderly people who live alone. Each robot should collect and send information of subjected objects and peoples that needs to be taken care. In order to achieve the appropriate response time for urgent treatment such as fall down of patients or elderly people watched by robots. We propose a novel mechanism to achieve responsive server that keeps the bandwidth for the urgent data transfer from robots. This mechanism can keep an appropriate communication bandwidth between robots even under the high overload on the server by controlling the computing resource form the operating system level. In this paper, we describe requirements for the platform of distributed care robot systems, proposal a new communication mechanisms and report the result of preliminary evaluation.

1. INTRODUCTION

Recently, the spread of inexpensive robots and wireless connection, distributed robotic system that works with multiple robots, is recognized as having great potential in the new service applications. In Japan, declining birthrate and aging society rapidly, it is expected that robots perform human behavior support. For example, there are introduced robots for the purpose of communication with human as floor guides such as Pepper [1]. They can be expected relaxing queue in a large shopping center where human resources are limited. Thus, distributed robotic system is expected to support our society and to decrease society human costs in a variety of situations. For these services, a dense communication that gathers and manages information is significantly important between the service provider robot and service customers, since robot will be near for the people to collect biological, behavioral, and environmental information to provide the appropriate services for each individual.

We have developed "Distributed Care Robot System" in which multiple robots work with patients in hospitals, and elderly people who live alone. It will be one of novel examples of a distributed robotic system. In our distributed care system, multiple robots provide support to the subject depending on the roles. That means that each robot transmits data to the server asynchronously. According to their role, some robots may frequently issue communication request and send their data to the server. Others may not request communication so often. Each robot sends its own collected data to the server asynchronously. Since required data, response and robot operation are different based on their role, the transmission frequency and the communication requirement are also different.

Table 1 Collected Data and Requirement on Distributed Care Robot System

Subject	Data	Transmission frequency	Communication requirement
Robot itself	Operation	Constant cycle	
	Power consumption	Constant cycle	
	Position	Constant cycle	
Human	Biological information	Stream (continuous)	
	Position	At the time of event arouse	High speed Certainty
	State	Stream (continuous)	
Other robots	Position	Constant cycle	

We classify the role and communication requirements in Table 1. From Table 1, the significant requirement is found in the communication requirement that influences the life of the subjected human watched. If the watched human has serious disease or injured, emergency signals concerning his/her condition change should be accepted urgently even the server would be busy.

The discussion of a mechanism to collect data from multiple robots has just begun. In general, conventional robots, unlike the computer, have unique hardware or software, moreover there is a restriction such as the low reliability of resources and network. In addition, about data processing in the server, the scheduling approach improving response and reducing computational cost by performed optimal placement of the core has been proposed [2]. However, when the load of the network itself is high, the data may not reach on the server in the first place. Therefore, controlling I/O subsystem is considered to be effective [3, 4]. However, there have been not a great deal of examples applied to a distributed robotic system, so must be verified.

In this paper, we propose a resource control mechanism that accepts anytime data requiring immediate processing and responds in the shortest on the keeping bandwidth, and verify validity within the middleware. In order to implement, the mechanism comprise resource control at the network level and allocation of data group.

As a method for resource control in the network level, and supported monitoring mechanism to ensure a bandwidth for receiving and acceptor for accepting a data transfer. At this time, we need to select communication protocol suitable for communication request data, the high data urgency to see the data flow on the network to. The monitor and assign different resource group for each socket in advance by resource container that offers by the operating system [3], and at the same time performs the control, the acceptor by a determined group in the server, is achieved by managing the different resource groups each thread. This time, as a first step to create a program for a server client model, with each communication protocol used when sending from the client to the server, followed by comparative experiments in which the time required for data transfer.

The remainder of this paper will be as follows. It describes the outline of the Distributed Care Robot System in Section 2, we introduce the existing research in Section 3. It proposes a proposal server design in Section 4, reported on preliminary experiment in Section 5 summarizes the paper in Section 6.

2. DISTRIBUTED CARE ROBOT SYSTEM

Distributed Care Robot System has been developed for use in support such as nursing support and elderly people living alone in the hospital. The system needs to be configured by the robots to perform different supports by arranged location, time, situation of the watching target has been changed. Specifically, the following operation is assumed. When the patient of interest does not move, one robot watch a plurality of patients. When the patient of interest is moving, the robot watch the individual. If the subject patient falls down, the watch robot is required to respond urgently in order to help the subject as soon as possible. In this situation, using the data collected from the sensor mounted on the robot and the data stored in the server operates each robot. Furthermore, the collected data transmits to the server asynchronously.

Rehabilitation Robot is a system in which the robot fitted with a cane to the top moves according to the movement of the subject. The robot is operated by always getting the data from the ultrasonic sensor mounted on it and calculating the distance to the subject.

Wandering elderly person tracking robot would be a system in which the robot moves to the falling position of the elderly person when he/she had fall in wandering without bring assistants, and the robot judged that he has been in dangerous condition. The robot makes determination of the fall and the position by indoor positioning on basis of the data obtained from the acceleration sensor and a gyro sensor mounted on the smartphone elderly has. The location information is transferred in case of an emergency when the elderly fall, because it is human life information, needs to be fast and securely transmitted to the server.

It is also important to get the robot own data. By acquiring operation status, the power consumption and position information of each robot, we can grasp the operational status or determine failure of them. Further, by knowing the position of the robot within a range, it is possible to move the robot to the required extent depending on the excess or shortage of the support robots. Data on the robot itself is acquired at a constant period and intends to be used to identify the cause of malfunction of the robot.

On the Distributed Care Robot System, since the frequency or communication request to send the data to the server are different, respective data are transferred and be processed in a suitable way.

3. EXISTING RESEARCH

Currently, ROS (Robot Operating System) middleware is widely used in distributed robotic system [5]. It includes libraries, hardware abstraction, device drivers, etc. for creating robotic applications. These tools are very useful, however, it is difficult that direct change of service such obtaining data via the network, holding them and switching the service quickly.

Furthermore, in order to improve the responsiveness of the data processing, there are scheduling methods, such as Inter-Core Time Aggregation Scheduler (IAS) [6] and SmartShare [2]. IAS is a scheduling technique for the reduction of execution time of process and improvement of throughput by aggregating the data to be used for the caching on the sibling threads and cores. SmartShare performs dynamic allocation of instances in the cloud servers and enables resource sharing between a multiple of interactive sessions from different users. It separates

performance and can reduce costs.

However, these scheduling mechanisms prerequisite that the data to be processed are present in the machine. Systems for collecting data over a network, when the network load is high, there are problems that it can not send the data.

4. PROPOSAL

4.1 OVERVIEW

In this study, regardless of the load on the network, it is intended to respond quickly relative to urgent data. Therefore, we propose a mechanism that performs dynamic adaptation as a middleware has resource control at the network level and allocation of data group (Fig. 1).

The middleware will provide a mechanism for capable of performing communication control with the priority granted in response to the advance service. For example, as shown on the right side of Figure 1, the *Robot 1* detected that the target has fallen. Then, the communication priority of the *Robot 1* is set to high, so that the information provided by *Robot 1* is processed with the highest priority. In this case, when the total data size of the transmitting low priority data size and the data size of when all clients connected to send high priority data to the server simultaneously exceeds bandwidth, by applying a resource limit, it will be at all times capable of the state receiving urgent data. the received data and distributes the queues in accordance with the priority that it is processed to the scheduler, and allow rapid response to urgent data.

The acceptor that allocates the data received from the client to each priority queue and the monitor that applies resource limits based on the flow of data transmitted between the server and client configure this middleware. We will describe these mechanisms in detail in the following subsection.

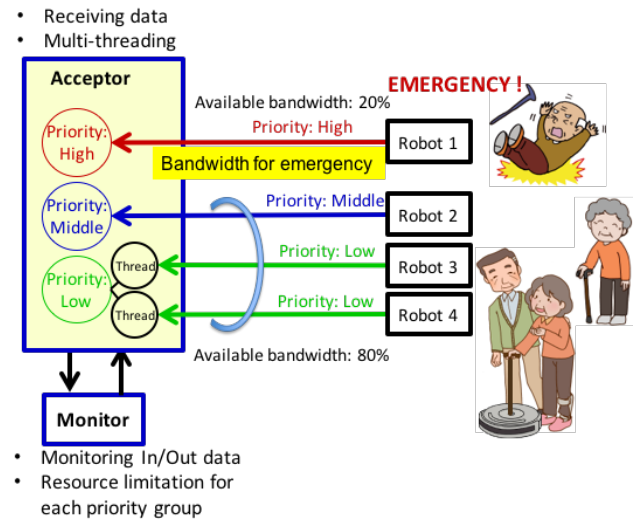


Fig. 1 System Architecture and Configuration

4.2 ACCEPTOR

The role of the acceptor is to accept the data from the clients that provides services at robot. Generally, it is provided with the computers. The acceptor, when it received data from the client, will create processes for each receiving data of client as the different priority groups. We provide a mechanism that spawn thread for successive data that would be contained in the group that have been already spawn as a more light weight process, Clients can be connected to a thread that matches the priority of the type of data to be transmitted, and transfers the data in the proper communications protocol for the communication request data.

4.3 MONITOR

Based on the information of urgency of the services, the Monitor manages the available resources amount of each process within the Acceptor as groups of different priorities. As a management method, it is assumed a certain process with the group, to use *cgroup* that limits the amount of resources to be used in the process. Monitor calculates the percentage of the amount of data the server is sending and receiving, using *sar* command, in the bandwidth. Then it ensures the bandwidth required to transfer urgent data by dynamically managing a resource amount to each group can be used.

5. PRELIMINARY EXPERIMENT

There are communication protocols of the transport layer in the OSI reference model. In this Distributed Care Robot System, in addition to regular status information of the robot, the robots send the video stream of the subject to the server. Therefore, we assumed to be mixed two data transfers of large-capacity video data and a small capacity. For UDP and TCP, which is mainly used to data transfer, we measured the data transfer time. Experimental environment is as shown in Table 2.

Table 2 The Experiment Machine's Environment

Environment	OS	Processor	L3 Cache	Memory
Source (client)	ubuntu 14.04 LTS	Intel Celeron 2.00 GHz	2MB	1.8GB
Desti-Nation (server)	OS X	Intel Core i5 2.6 GHz	3MB	8GB

In UDP and TCP each protocol, we have created the basic socket program of the Source and the Destination. T_{S1} the time the Source begins to transmit the data, T_{S2} the time of receiving a reply from the Destination, T_{D1} the time when the Destination receives the data, T_{D2} the time of starting to send the reply. We measured among the time from the Source begins to send the data to receiving the reply as data transfer time $((T_{S2}-T_{S1})-(T_{D2}-T_{D1}))$ (Fig. 2). Using the local network that is not used except this experiment, the load on the network is assumed to be constant. Data size to the sender to send is 256 bytes.

Moreover, in the case of the machine, the data to be transmitted is first copied from the program to a system buffer (Fig. 3). In this case, in order to use what is stored in the cache when a memory address of the data is not changed, the processing time is shortened. However, on final implementation, since the collecting data from the robots, the cache does not affect it. Therefore, in this experiment, in each of sending and receiving data, we eliminated the influence of the cache by shifting the start address of the data 3 cache size above.

Measuring the data transfer time by 12 times in the case of UDP and TCP, respectively, it showed the worst value, the fastest value, average, standard deviation in Table 3.

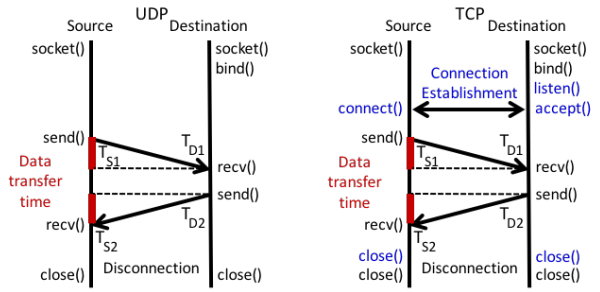


Fig. 2 Data transfer time

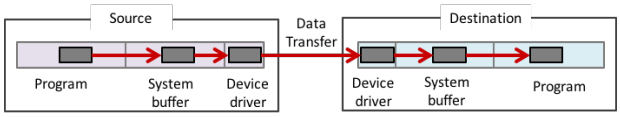


Fig. 3 Flow of data transfer

Table 3 Results for Each Protocol (μ sec)

Protocol	Worst	Fastest	Average	Standard deviation
UDP	3510	630	2013	1002
TCP	4440	1430	2679	774

Data transfer time becomes shorter overall than the UDP in TCP. There are speed differences about 20% on the worst values and the averages, and about 50% on the fastest values. However, the standard deviation towards the TCP reaches approximately 20% smaller than the UDP. is a protocol that emphasizes a reduction in latency. It is only a simple data transmission and does not guarantee the reliability of data transfer. Meanwhile, TCP is the emphasis on reliability protocol. It performs processing such as retransmission of the lost data or rearrangement of the order of the arrival of data. Therefore, It considered the data transfer time on TCP has become generally longer than the UDP.

6. CONCLUSION

Distributed inexpensive robots are becoming popular now and expected to be widely used for human modern society. However, current middleware for distributed robotics systems has not yet provided an adaptive control mechanisms of the communication priority in order to keep touch in service servers at urgent situation. In this report, we report the necessity of a middleware for the distributed care robot system to serve receiving a data transmission with multiple priorities, We propose a mechanism for improving responsiveness concerning to urgent data. While the communication request for each data to be collected may vary, we have implemented a basic data transfer program and evaluated results. It was subjected to comparison with UDP and TCP protocols. From results, it observed that data transfer speed difference is up to 50% .

In future, we will implement the middleware portion of the proposed system and evaluate it from the view point of responsiveness at the time of collecting various types of data having different priorities.

REFERENCES

[1] Robot | Softbank, Softbank, http://www.softbank.jp/robot/

[2] Tao Zou, “Optimizing Response Time For Distributed Applications In Public Clouds”, *A Dissertation Presented to the Faculty of the Graduate School of Cornell University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy,* January, 2015

[3] Gaurav Banga, Peter Druschel and Jeffrey C.Mogul, “Resource Containers: A New Facility for Resource Management in Server Systems”, *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation New Orleans*, Louisiana, February, 1999

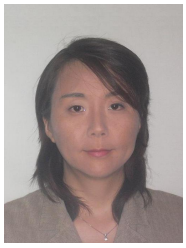
[4] Eric W.Anderson and Joseph Pasquale, “The Performance of the Container Shipping I/O System”, *ACM SIGOPS*, Vol.29, Issue.5, p.229, December, 1995

[5] ROS Wiki, Open Source Robotics Foundation, http://wiki.ros.org/ja

[6] Satoshi Yamada and Shigeru Kusakabe, “Implementation and Evaluation of Inter-core Time Aggregation Scheduler with an Aggregation Control Mechanism”, *ACS*, Vol.3, No.3, pp.235-247, 2010

Satoru Nakayama received the B.E. degrees in Information Science and Engineering from the Shibaura Institute of Technology, Tokyo Japan, in 2015. He is now a master course student of Graduate School of Engineering and Science, Shibaura Institute of Technology.

Miyuki Nakano received BS, Ph.D from the University of Tokyo, Associate professor at the Institute of Industrial Science, University of Tokyo in 2009, At the Shibaura Institute of Technology, professor. Currently, she is professor at Advanced Institute of Industrial Technology.



Midori Sugaya received MS, Ph.D. degree in Computer Science from Waseda University in 2010. Currently, a associate professor at Shibaura Institute of Technology from 2013.