Child watching support system with dynamic routing of motion pictures

Airi Kaneko and Shigeo KANEDA
Information System Laboratory
Graduate School of Doshisha University
Kyotanabe, Kyoto, JAPAN
email: airi@ishss10.doshisha.ac.jp

ABSTRACT
We investigate a ubiquitous home network service that most resembles our everyday life. When looking at childcare support systems as important ubiquitous applications, proposals do research and development on nursery school. However, proposals rarely do considering inside homes. Therefore, we propose a child-care support system inside homes called DROP-CWS with peer-to-peer networks that don’t use servers. We use a communication platform called “Jupiter” in which peer-to-peer protocols are incorporated. DROP-CWS observes a child playing inside the home by surveillance camera, allowing care providers to monitor from another room. Furthermore, care providers can select a camera for observation because we use multicast communication within a group that consists of one camera and two televisions. We did operation experiments in an environment with two rooms and used commercial cameras and televisions. Even though results showed successful delivery of motion pictures, the picture was projected on a television screen by a frame-by-frame advance. As future work, we will continue research and development of this system.

KEY WORDS
Motion picture, Jupiter, platform, peer-to-peer, ubiquitous, and child-care support

1. Introduction
Recently attention has focused on the research of home network digital appliances that have been spreading over the years. While research for ubiquitous societies is being conducted, ubiquitous services that feel the most familiar to our everyday life are services in home networks. In ubiquitous environments, since various sensor and/or database information are needed from computers connected to networks, load concentrations and real-time decreases are caused when the information is managed by servers. Furthermore server cost is also a problem for home network environments. Therefore, we use peer-to-peer (P2P) networks to solve these problems. Unlike client/server networks, P2P networks do not need brokers because they directly communicate information among users. Child-care support is another important ubiquitous application. Research has been done into services that allow guardians to watch and communicate with children from a distance, for example. However, they are mainly designed for nursery schools. Research and development aimed for home service is scarce. Based on the above, we consider a child-care support system for home networks with P2P networks. However, we need to think about protocol and operation management of the P2P node when constructing a P2P network system. If such a base platform is prepared, the system can be easily constructed. As a platform candidate, we selected “Jupiter,” a communication platform developed by Peer-to-peer Universal Computing Consortium (PUCC), and based an implement system on it.

In this paper, we describe specification of DROP-CWS and description of Jupiter platform in section 2. Development of DROP-CWS is described in section 3. In section 4, experiment and discussion of this system is described. Section 5 contains the conclusion.

2. Overall DROP-CWS
2.1 Application Example
Here, we describe an example of a child-care support system in a home network shown in the preceding section. We consider a child-watching support system with commercial cameras and televisions inside the home. The system shows a child playing in a room captured by video camera; his care provider can confirm the child’s appearance with a television in another room. That is the simple motion picture system at a glance. However, when a child or care provider moves between rooms, the system can switch delivery routes to a video camera or television located at the new destination and take/project motion pictures. In other words, cameras and televisions
communicate with each other to make delivery route switches. Therefore, the purpose of the system is to create a P2P network with home electric appliances and realize a delivery system with dynamic routing of video. We call our system Dynamic Routing Of motion Picture Child Watching Support System (DROP-CWS).

Web cameras are generally used for motion picture delivery systems. Because the images captured by Web cameras are of relatively low quality, they aren’t good enough to be shown on television for child watching. The instruments used for lectures, meetings, and medical treatment scenes are not Web cameras but high-resolution ones that adjust to each scene.

2.2 Function Specification

Figure 1 shows an image of DROP-CWS. It assumes four rooms in which surveillance cameras or televisions have been installed. Figure 1 also shows an image taken by camera while a child is playing in room 1. A parent working in room 2 can monitor the child by the television installed in room 2. In other words, a parent who cannot be in room 1 can monitor the child while working or doing housework in another room. When the parent moves from room 2 to room 4 to answer the telephone, the system switches the video image delivery route from room 2 to room 4, allowing the parent to continue watching the child on the television in room 4. When the child moves from room 1 to room 3, the system will stop taking the image in room 1 and switch the routing of the camera to room 3. Therefore, the parent can continue watching the child.

DROP-CWS provides parents/care providers with an opportunity to keep watching children on a television installed in their room because delivery routes are switched when a target object moves to another room inside the home.

2.3 Brief Description of Jupiter Platform

Jupiter [1] is a P2P networking infrastructure for ubiquitous communication environments that can realize P2P networking among various devices. Therefore, Jupiter is suitable for the development of P2P applications.

P2P communication is performed between P2P nodes. Each node belongs to one or more communities defined as follows. The term “community” means a logical collection of P2P nodes that have a common interest and obey a common set of policies. A community is identified by a community ID. A P2P node is an independent, bidirectional communication entity in the P2P network. Such devices as mobile devices, PDAs, personal computers, servers, or workstations can be used on the P2P node.

P2P Multicast Communication

In P2P networks, multicast communication is available for a variety of applications, such as communication among work group members in group ware applications and so on. A multicast network overlays a P2P network. Multicast forwarding is carried out by multi-hop or multi-destination unicast in a P2P network. Therefore, not all nodes in P2P networks have to support multicast.

Figure 2 shows the P2P multicast path construction procedure. When a P2P node tries to join a certain multicast group, it is operated as follows:

1. A P2P node sends a join message to the nearest member node.
2. The nearest member node returns a join response message to it.
3. The two nodes establish a relationship regarding the multicast group.
4. Next, a multicast message propagates along the multicast paths.

In Jupiter’s architecture, a node looks for the nearest member node using a flooding message when it tries to join a multicast group.
Protocol Design on Jupiter

Figure 3 describes the protocol stack on Jupiter. Protocols on Jupiter are designed over TCP and IEEE 1394 in two layers. The P2P Core Protocol is defined to process P2P messages based on the P2P communication model. Seven protocols realizing P2P multicast, communication with a control node, control of a P2P session and so on are defined over the P2P Core Protocol. Based on a layered protocol design, it is easy to design a new P2P application protocol based on the requirements of P2P applications. Furthermore, the protocols are defined using XML. Since XML can represent tree structure data, it is possible to design complicated protocol messages required by P2P applications and layered protocols independently using XML Namespace. Therefore, XML is well-suited for application protocols.

Overview of P2P Core Protocol

The P2P Core Protocol plays such fundamental roles as sending, receiving, or forwarding P2P messages as a P2P communication model. There are three types of messages: request, response, and advertising. Request and response messages are defined for the reactive communication mode, and advertising messages are defined for a proactive communication mode. In addition, three communication types are defined in Jupiter: unicast, broadcast, and multicast. A unicast message is directly sent to the specific destination node using multi-hop or multi-destination unicast. When a node receives a multicast message from a multicast member node, it forwards the received message to the remaining adjacent member nodes using multi-hop unicast. Then a node sends a broadcast message to all adjacent nodes. The forwarding of a broadcast message is controlled by its hop count. Since the naming and message routing mechanisms of the P2P Core Protocol are defined to be independent of transport protocols, the P2P Core Protocol can be defined over various transport protocols.

3. Prototype Development

3.1 Processing in DROP-CWS

We will now describe the required processing method to realize the DROP-CWS shown in the preceding section. Figure 4 shows the structure of the prototype DROP-CWS. As shown above, we used commercial video cameras and televisions. To use the Jupiter platform, all equipment must include Jupiter software. But currently there are no video cameras or television sets on which Jupiter software can be installed. Therefore, putting a proxy node on each device has been adopted. Proxy nodes and such equipment as video cameras and televisions are connected by IEEE 1394 interfaces. Using the proxy node concept, any equipment can participate directly in a P2P network for virtual communication.

As shown in Figure 4, we implemented a motion picture delivery system with multicast communication within a group that consists of one camera and two televisions. In unicast communication, one node (called the originating node) transmits and specifies one destination node. Therefore, unless the originating node on the camera side changes destination node when the parents move, that is, the television being watched by the parents is switched away from, the motion picture is not delivered to the switched television. Since it cannot be operated from the camera side when a child is in the room, the destination node cannot be changed. Furthermore, since the camera side node does not know when the parents move to another room, the system’s purpose is not satisfied. On the other hand, since nodes can communicate with each other within a group in multicast communication, the system is satisfied. For example, suppose that Camera 1 shown in Figure 4 participates in group “A.” Camera 1 specifies group ID “A” as a destination node and starts transmission. TV 1 starts receiving motion pictures from Camera 1 by participating in group “A.” When a parent moves to another room, that is, when delivery route is switched from TV 1 to TV 2, Camera 1 must change the node of TV 2 as destination node in unicast communication. However, TV 2 can receive motion pictures from Camera 1 by participating in group “A” in...
multicast communication. Furthermore, the node on Jupiter can leave the participating group. If receiving motion pictures from Camera 2, each television can watch them by leaving group “A” and participating in group “B,” similar to the groups of Camera 2. In other words, parents who are on the television side can choose a camera to watch motion pictures.

For example, suppose that Camera 1 shown in Figure 4 participates in group “A.” Camera 1 specifies group ID “A” and starts transmission. A television participating in group “A” starts receiving motion pictures from Camera 1. Moreover, the television receives motion pictures from Camera 2 if leaving group “A” because it can secede from one group to participate in group “B” in which Camera 2 participates.

In addition, because the data are in binary form when being transmitted and received, Jupiter outputs error when analyzing them as XML if the data is transmitted as expressed in the preceding paragraph. Therefore, using a data encoding method called BASE64 makes it possible to transmit and receive binary data.

3.2 Function Implementation on Jupiter
Based on the structure of the prototype system shown in Figure 4, we implement DROP-CWS as an application on Jupiter. It is implemented with Java when a Jupiter Application Programming Interface (API) is offered. However, since Java does not have a library to read and write an IEEE 1394 bus, that is, an interface for cameras and televisions, we implement it with C for reading and writing IEEE 1394 and communicating with Jupiter. For an OS, we use Linux whose library source is opened. Furthermore, we implemented an application using a GUI (Java.awt package) to display the log because we want to know the state of the self node, the adjacent node, and the communication condition when the application is being operated. Reactive communication of request and response is used for the communication mode. Multicast transmission is used for requests, and unicast is used for responses. All the operations done with GUI are shown as follows. We prepared entry field functions of group ID in which the self node participates and leaves and a destination node ID for transmitting on GUI. Table 1 shows the methods referring to node states that are mainly used for this system. The methods are defined over P2P Application Protocols (Figure 3). Moreover, we added join and leave methods to output messages whether self node participates in specified multicast groups or leaves. They are defined over P2P Multicast Communication Protocols.

In addition, button functions are prepared to start and stop transmission and select which camera or television is connected to self node.

Next, we describe the processing procedure of each node connected to the cameras and televisions. First, at the camera side node, a user inputs a group ID in which self nodes participate and pushes a button to select equipment connected to the self node (camera button). Furthermore, the caregiver inputs an ID in the field of the destination node ID, similar to a group ID that he inputted in the participating entry field. On the other hand, at the television side node, he pushes the television button and inputs an ID similar to the group ID in which the camera to watch participates in the participating field. For example, at the Camera 1 node, which is connected to Camera 1 in Figure 4, he inputs “A” in the participating field. Furthermore, he inputs “A” in the field of the destination node ID and inputs “A” in the participating field at the TV 1 node, which is connected to TV 1. When the transmission start button is pushed at the Camera 1 node, motion picture data are transmitted to the TV 1 node. At TV 2 node, which is connected to TV 2, “A” is similarly inputted in the participating field, allowing the parent to watch the motion pictures of Camera 1.

Furthermore, the motion picture can be watched from other camera when the node leaves the group in which it has participated to join another group. At Camera 2 node, a user inputs “B” in the participating field. At the nodes of TVs 1 and 2 at that time, “A” is inputted in the leaving field, and “B” is inputted in the participating field. When the transmission start button is pushed at Camera 2 node, he can deliver a motion picture from Camera 2. In this way, the group ID in which it participates is decided in advance at the camera side. Therefore, users can operate freely so that the node participates in and leaves a group similar to the group in which the camera for watching participates at the television side.

Camera and Television Control
It is necessary to use an IEEE 1394 driver offered by Linux Kernel as an interface for video cameras and televisions. Dv1394 [2] is a device driver from the Linux Kernel 2.4.18 standard for controlling DV cameras that is connected through IEEE 1394 and offers device files to communicate data with DV devices through IEEE 1394. Reading or writing this makes it possible to transmit and receive data. Using this, we implement a function to communicate data with cameras and televisions connected through IEEE 1394 as a class that offers each function to communicate with Jupiter. We summarize each class as follows.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adjacentNodeChanged</td>
<td>Notification that the adjacent relation of the node has changed.</td>
</tr>
<tr>
<td>nodeStarted</td>
<td>Notification that the node has started/stopped.</td>
</tr>
<tr>
<td>nodeStopped</td>
<td></td>
</tr>
<tr>
<td>messageTransferActivated</td>
<td>Notification that message can/cannot be transmitted or received.</td>
</tr>
<tr>
<td>messageTransferDeactivated</td>
<td></td>
</tr>
<tr>
<td>multicastMessageTransferActivated</td>
<td></td>
</tr>
<tr>
<td>multicastMessageTransferDeactivated</td>
<td></td>
</tr>
</tbody>
</table>
Implementation of DvCamera class
This class takes pictures per frame. This is an open device by function `dv_init` and takes data for one frame by function `get_frame`. In NTCS, which is the normal format in Japan, they are 120,000 bytes/frame and 30 fps.

Implementation of DvTv class
This class transmits motion picture data per frame to the converter connected to televisions. It is an open device by function `dv_init` and transmits data for one frame by function `throw_frame`. If the argument of function `throw_frame` indicates the frame buffer, the data are transmitted. If the argument is NULL, the data that take last are transmitted again. As a result, delay problems of communication and processing are solved.

The above classes and nodes on Jupiter transmit data one frame at a time. Reactive communication is used to learn the time at which a node sends the next frame. In another method, a node continues transmitting data for one frame on its own using proactive communication. However, the node usually takes the form of a response when communicating with each other. In this case, a node also exactly transmits the next data after checking that the latest data have been accumulated by the receiving side buffer when transmission takes a long time.

Furthermore, when data are in binary form, they must be encoded because Jupiter transmits data read in XML documents when nodes on Jupiter communicate with each other. Otherwise, error will be output. Therefore, using an encoding data method called BASE64 allows binary data to be transmitted.

4. Experimental Usage and Discussion

4.1 Experimental Environment and Used Devices
We prepared two rooms 3 m × 3 m (Figures 5 and 6) as an experiment environment of DROP-CWS. A list of the devices used is shown in Table 2, and the connection framework is shown in Figure 7.

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>television</td>
<td>Sony VEGA KDL-L26HDX</td>
</tr>
<tr>
<td></td>
<td>Panasonic Viera TH-37PX500</td>
</tr>
<tr>
<td>converter</td>
<td>Canopus HDVC-110</td>
</tr>
<tr>
<td>camera</td>
<td>Sony Handycam DCR-HC40 NTSC</td>
</tr>
<tr>
<td></td>
<td>Canon DM-FV20</td>
</tr>
</tbody>
</table>

Table 2 Devices used for experiment

4.2 Results and Discussion of Experiment
We performed experiments in the above environment. As a result, data communication of the motion pictures succeeded on the camera and television sides, confirming that the pictures taken by video camera were projected onto the television screen. However, they were output not in a smooth state but by frame-by-frame advance. A 2-fps picture cannot be called a motion picture if measured by the eye.

One possible cause is dropping frames, that is, whether part of the data is dropped during communication. As an experiment, when a person is photographed sitting still for several seconds, and then makes such a movement as shaking a hand, we understand that the picture of him moving does not seem unnatural by the eye on the television screen, even though the few frame rate is low. We understand that the frames of the picture have not been dropped. When consecutively watching one frame of motion picture data using a video-editing program on the television side PC, the frames do not seem dropped.

Next, we examine whether communication time was late. First, for communication time with the camera side PC
transmitting data and the television side PC receiving data, no problem exists because synchronization is almost required when transmitting and receiving. Second, for communication time with a camera and a camera side PC, we understood that, even though the data size sent from camera to camera side PC is 120,000 bytes/frame, as shown in Section 3.2, it was actually sent in 2 or 3 steps. In the reading data file test in the experimentation phase, we confirmed that 120,000 bytes can be read at a time. However, when real-time communication is actually performed, the data are accumulated in buffers twice 65,536 bytes, 120,000 bytes, or three times 32,768 bytes, 98,304 bytes, 120,000 bytes at times, and transmission is started. The cause is unknown. In addition, for communication time with the television side PC and the television as well as with the PCs, there is also no problem because synchronization is almost achieved. Therefore, as future research we must execute the following things from three points of view to solve the uncertain causes. The first step is whether Jupiter architecture itself controls the quantity of data for some causes. Other areas include a review of data processing treated by the DvCamera Class and programs implemented on Jupiter.

5. Conclusion and Future Work
In this paper, we implemented a DROP-CWS switching delivery route of motion pictures with human movement on a “Jupiter” communication platform that comprises a P2P network. We experimentally demonstrated the system; however, good results were not obtained. We discussed the reasons and intend to continue research and development of it in the future.

The development of Jupiter has advanced, and it aims to be installed and operate with such mobile devices as Personal Digital Assistances (PDAs) in the future. In the current Jupiter, mobile terminals can also participate in P2P networks. If operated from mobile terminals, a user can easily confirm a child’s situation anytime, which is the system’s purpose. Therefore, attention to the child is strengthened. A node can also communicate outside the home, for example, to mobile terminals by gateway nodes included in the network inside the home. This addresses the security problem. Since the current gateway node only has a function to relay networks, all communication to be accessed from outside the home goes through the gateway node. Therefore, security will be enhanced if a function is added that filters those communications.

Furthermore, in this system a motion picture is not delivered unless the user watching television activates the “participate in group” operation. Therefore, we must consider an operation done whenever a user moves to a different room. It is possible to install a sensor that detects user movements between rooms and so on. The network path automatically switches based on such information so that a motion picture is delivered to television. This is limited to inside homes. However, assume that we offer non-portable televisions that lack the convenience of automatic switches. For example, since the care provider dislikes troublesome operation when occupied with cleaning and so on, he does not confirm the child’s appearance by television. At such times, a user can exclude that the tendency that attention to a child is inferior.

While various research and development has been done toward ubiquitous societies, communication platforms like Jupiter have been also developed. Furthermore, service development and experiments have been reached based on those platforms; however, they do not actually exist in our daily life yet. The background of the environment where we live differs from the experiment environment of ubiquitous service. However, a sensor network environment can be seen in companies or commercial buildings. On the other hand, such environments are scarce inside homes. Currently private associations are establishing interconnections of digital appliances by spreading them. Although the environment changes slowly in this way, some appliances whose interconnection is not established in the environment are used for research. Ubiquitous services will exist someday. Therefore, we have proposed and constructed DROP-CWS to realize a system that can be used immediately with current appliances. If this system is implemented, we hope that it can eventually be operated from mobile terminals such as cellular phones, changing the system that can become part of our daily life.

Acknowledgements
This work was done using the Jupiter platform developed by Peer-to-peer Universal Computing Consortium (PUCC). The authors sincerely appreciate its technical support and assistance from the members of the PUCC Home Appliance Working Group (HA-WG).

References