Seamless Service Platform for a Ubiquitous Network Environment

Koichi Takasugi†, *Motonori Nakamura, and Minoru Kubota*

Abstract

In the future, many kinds of terminals will collectively form a ubiquitous network environment that is available anytime and anywhere. In this type of network, nodes will often change communication media and network addresses and will be frequently disconnected from the network. Moreover, users will often move and use different terminals and want to enjoy services without interruption even when such changes occur. Our seamless service platform conceals changes in a dynamic network environment by applying an overlay network on top of the transport layer.

1. Introduction

In a ubiquitous network environment, nodes can be arranged anywhere and they construct networks dynamically. A dynamic network provides high flexibility, but if existing technology is applied, communications between nodes become unstable and intermittent. From a user's viewpoint, it is not enough to keep a terminal (node) connected to networks, as is now common with always-on network connections. Users want to continue to enjoy the same service without interruption even if they change terminal or use different network resources.

Therefore, we are studying node migration which keeps a node's connection to a network and service migration which continues a service even if nodes are changed [1]. This is achieved by using a virtual network constructed with seamless proxies that relay data between applications and keep connections among themselves.

Most research to date considers the movement of a node running a client application. However, in a ubiquitous network environment, a node acting as a server may move frequently. For example, a sensor that delivers content or a mobile terminal acting as a content source may move. We focused on nodes that act as servers and on the duplication of data needed to provide the same service when the service is dispersed on the network, and we developed a method that continues server service seamlessly even if the node running the server service is changed while service is being offered. In addition we developed a method of automatically making a route, which is a sequence of seamless proxies. This method decides the route considering the situations of the user and each service.

2. Seamless service

Consider a user watching a movie being delivered over the network. If the movie delivery service application is disconnected from the network while it is delivering the movie, then another movie delivery service application that has the same movie can pick up the delivery from the point of interruption (server service migration). Another example of seamless service occurs when a user who is shopping online goes outdoors: the user switches from an application running on a desktop PC to one on a mobile PDA (client service migration). Our seamless service supports various dynamic network changes and user requirements at the application level that occur in a ubiquitous network environment.

We can represent seamless service using the seamless proxy network model, shown in Fig. 1. A seamless proxy is placed at each node. These seamless proxies construct a virtual network by an overlay network on top of the transport layer. Data is exchanged

† NTT Network Innovation Laboratories
Musashino-shi, 180-8585 Japan
E-mail: takasugi@ma.onlab.ntt.co.jp

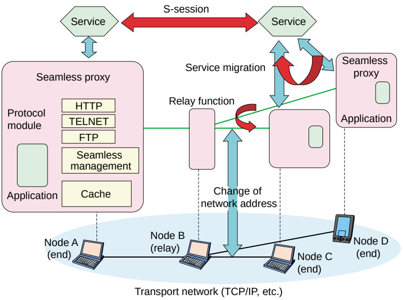

Fig. 1. Proxy-network model.

between applications via two or more seamless proxies, which act as relays for the data. A service is defined as the distribution of content to a user (server service), the display or playback of content for the user, or operations by the user (client service). A service is provided by an application, and the user is generally able to choose from among several different applications capable of handling the same service. The communication in a seamless service is represented by the data exchanged between services. An S-session (seamless session) is a series of this meaningful data. Service continuity means keeping this S-session.

It is possible to select the communication medium to use and reconnect and resume the communication if communication between seamless proxies is cut off. This is achieved by a seamless proxy accumulating relayed data and sending this data flow to an arbitrary communication device. For example, it is possible to change the communication medium between seamless proxies from a wired LAN to a wireless LAN. In addition, by changing the relay route among seamless proxies, it is possible to shorten the route if the route acquires redundancy as a result of network topology changes.

Furthermore, the seamless proxy understands the state of an application and protocol by analyzing relayed data and asking the application. By transmitting this state to another seamless proxy and changing the end point of the relay route of the seamless proxy to this seamless proxy, it is possible to continue the service with a new application on a new node.

For example, in Fig. 1, the communication between applications in nodes A and D is switched to communication between applications in nodes A and C by transmitting the state of the seamless proxy in node D to the seamless proxy in node C. As far as the user is concerned, the service (either client or server service) seems to continue without interruption.

2.1 Related work

Examples of existing technology that can adapt to dynamic changes in a network include the routing protocols for an *ad hoc* network [2] and peer-to-peer (P2P) service. An *ad hoc* network is formed by direct links between wireless mobile nodes and does not require the full infrastructure of a wired network. It is a good method of constructing a dynamic network. However, it is not possible to continue service when a node moves from a wireless network to a wired network or when the user changes nodes freely.

The P2P concept makes it easy to discover services in a dynamic network, but if the communication environment is changed frequently, it lacks the ability to continue service seamlessly because communication is assumed to be stable during service. Internet Indirection Infrastructure (i3) [3] is a P2P overlay network that provides a rendezvous-based communication abstraction. This is for a best-effort service and cannot support service continuity directly. However, it is compatible with our proposed approach and the service discovery method of i3 is useful.

Our seamless service platform can make up for these deficits by associating with the routing protocol of *ad hoc* and P2P networks.

Service continuity schemes, which are implemented in the IP or TCP layer, include Mobile IP [4] and Migrate [5]. They are independent applications that allow node migration. However, service continuity schemes have drawbacks: they need certain fixed nodes to run a home agent and are OS-dependent with regard to the implementation. They also have difficulty working with a firewall. Furthermore, to support service migration, it is necessary to manage the state of applications and resources of nodes and the network, and assign these resources. Therefore, it is difficult to support service migration using only the IP layer approach.

In the mobile agent system [6], a program's code and its state move from node to node and the same program is continuously executed. However, at that time, the mobile agent maintains only the program state and does not maintain the service state to the communication partner node, so the mobile agent system cannot continue a service through the network. In addition, our service migration does not need to move program code. Therefore, mobile agent migration is basically different from our service migration.

Our application-level approach is independent of the OS, transport network, and wireless technology and it does not assume the existence of any fixed nodes and achieves service migration. By cooperating with applications closely, it enables them to fulfill their intentions. Furthermore, node migration is achieved just by our platform; if necessary, the *ad hoc* network protocol and mobile IP can be used under our platform.

2.2 Seamless-service scenario

A seamless-service scenario is shown in Fig. 2. A communication between two personal digital assistants (PDAs) is initially established within an *ad hoc* network (1). The topology of the *ad hoc* network is changed and a new PDA is inserted between the first two (2). After that, the PDA at one end migrates from the *ad hoc* network to a hotspot for a wireless LAN, which is in a restaurant or some other public place (3). The user of that PDA then goes home and switches on his desktop PC to take advantage of its higher-quality display (4). Thus, the client node for this user is changed. Our platform can maintain seamless service throughout this sequence; that is, it continues to deliver the same service while maintaining the service state.

3. Implementation

We implemented a prototype using Java. It is

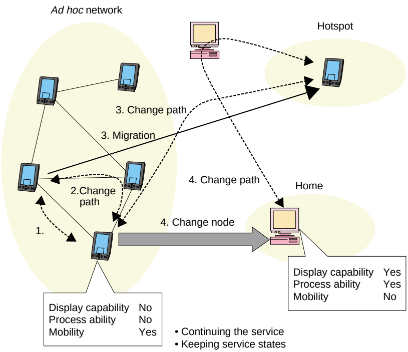

Fig. 2. Seamless-service scenario.

important to run on various kinds of hardware because there will be many different types of nodes in a ubiquitous network environment. Our prototype can use several operating systems (e.g., Linux, Windows, and Mac-OS) that contain a Java virtual machine.

Our seamless service platform can easily accommodate various application protocols because the main functions of our platform can be used in common and the only additions to the platform are a wrapper program (which starts an application and sets up its state) and the modules for each of the supported application protocols. Therefore, an existing application and application protocol can be used without modification. In fact, our prototype accommodates three application protocols (Telnet, FTP (File Transfer Protocol), and HTTP (Hyper Text Transfer Protocol) for Web Access and MPEG file delivery), each of which has individual characteristics.

A seamless proxy is uniquely specified by the proxy ID. The format of the datagram for inter-seamless-proxy transfer consists of a 36-byte header and data of variable length. The sequence number is used in the acknowledgment of received data.

3.1 Node migration

Node migration occurs when the network address of a communications device or the device itself is changed because of the node's physical movement or because the user has exchanged one communications device for another. In the IP layer, changing the communications device is equivalent to changing the network address, so support for changes in network addresses covers all types of node migration. Node migration involves a change in the relationship between the proxy ID and network address and is achieved by closing an old connection of the transport layer and opening a new one.

Since a sequence number is given to each datagram and an acknowledgment is sent during the process in which a seamless proxy accumulates transmission data in its cache, data is thus relayed without loss when node migration occurs.

3.2 Change of node being used (client service migration)

Figure 3 shows the sequence of client service migration when the node being used is changed from B to C. Figure 4 shows our prototype delivering MPEG TV to one node (left side) and then switching the destination to another node (right side). The user can continue watching the movie from the same point without having to restart from the beginning. This is achieved by recording the first node's delivery state as cookie information which is copied to the seamless proxy. Then, when the user launches a movie player on another node, this cookie information can be accessed by another seamless proxy so that the user continues to receive the movie datastream from the same point.

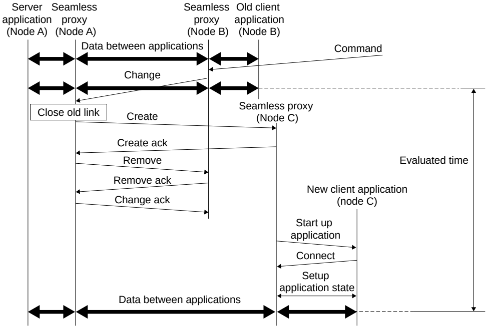

Fig. 3. Sequence of service migration.

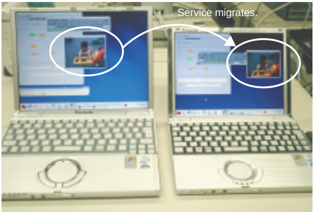

Fig. 4. Our prototype in action.

3.3 Shadow service (server service migration)

The service is set up beforehand as a duplex service consisting of a main service and a shadow service (backup service) in the network. To discover shadow services, the seamless proxy utilizes the discover protocol of JXTA [7], which is a P2P platform. Information about shadow services is advertised in the network, so each seamless proxy can obtain the sevice information (node address) by receiving this advertisement.

We are currently conducting experiments on service migration using our prototype. MPEG data is dispersed as a shadow service in the network and the MPEG datastream can be switched to another identical MPEG datastream from another terminal during delivery to a user. At that time, the seamless proxy converts an MPEG data file such that already-played data is cut and the playback time is changed, etc. This means that after you watch and record a sports event using a personal video camera equipped with a network interface (Fig. 5 (1)), you can watch images of the event from the camera through network, and if the camera is disconnected from the network, you can continue to watch it by automatically changing to a shadow service (Fig. 5 (3)) that was set up in advance (Fig. 5 (2)).

3.4 Routing method among seamless proxies

To avoid placing any burden on the user, the seamless proxies should create routes automatically without instructions from the user, so a routing method between seamless proxies is required. It must construct an appropriate route among seamless proxies when a service is started and must reconstruct it when the network topology or user requirements change.

To achieve this, a cost value is assigned to the virtual link between seamless proxies and each seamless proxy modifies it according to the stability or load. This makes it possible to select an appropriate route by reevaluating this cost value for each seamless proxy.

In addition, a user can express a preference for the interface and relaying seamless proxy in every ser-

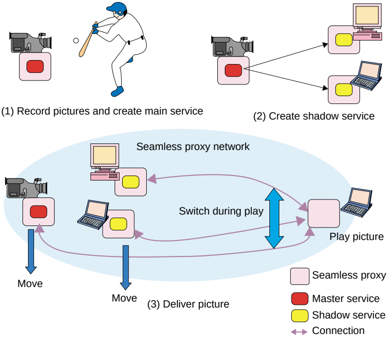

Fig. 5. Shadow service scenario.

vice. This makes it possible, for example, for a user to select a relaying seamless proxy that has a translation service and get English translated to Japanese during a video conference. That is, it is possible to assign an appropriate route to an individual service.

We are currently conducting experiments with routing methods on an actual network consisting of nine personal computers and five sub-networks and are simulating more complex situations.

4. Experiments

We conducted experiments with FTP (file transfer protocol) on our prototype. We used three notebook PCs as nodes running RedHat Linux. A seamless proxy was placed at each node. Nodes were connected by wireless and wired LANs. The wireless LAN operated in the *ad hoc* mode defined by IEEE 802.11b. Throughout the experiments, we found that the interruption time (determined as the duration of the interruption in communication between end seamless proxies) was only 400 ms for a change in IP address (node migration) and only 1 s for a change in node (service migration), which is equivalent to the evaluated time in Fig. 3. This is sufficiently short compared with the TCP-handover approach [4].

To measure the overhead for using the seamless proxy, we downloaded a 12.7 Mbyte file by FTP over a wired LAN and measured the download time for two cases: one in which we used two S-proxies and the other in which we used none. If three simultaneous downloads occurred, using the S-proxies raised the required time by about 5%. This corresponds to the overhead incurred by using the S-proxy. The main sources of overhead are the 36-byte header and acknowledgments and the delay caused by the processing on the S-proxy. A 100BASE-TX LAN was used in this experiment. If testing were in an Internet environment, the transmission speeds would be lower and other delays would be greater. Hence, the overhead would become less significant relative to the other factors.

5. Conclusion

Our seamless service platform can continue service even if networks and users requirements change dynamically. We explained node migration, service migration, duplication service, and routing between seamless proxies. We have implemented a prototype seamless proxy for the seamless service platform and experiments on it have demonstrated the feasibility of the concept. It has a wide range of applicability for future development: for example it can easily be applied to mulitipath multicast communication by branching off at a seamless proxy.

References

[1] K. Takasugi, M. Nakamura, S. Tanaka, and M. Kubota, “Seamless Service Platform for Following a User’s Movement in a Dynamic Network Environment,” IEEE PerCom 2003, pp. 71-78, Fort Worth, TX, 2003.
[2] “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” RFC2501, Jan. 1999.
[3] I. Stoica, D. Adkins, S. Ratnasamy, S. Shenker, S. Surana, and S. Zhuang, “Internet Indirection Infrastructure,” ACM SIGCOMM, pp. 73-86, PA, Aug. 2002.
[4] Perkins, C. E., “IP Mobility support,” IETF RFC2002, 1996.
[5] X. Snoeren and H. Balakrishna, “An End-to-End Approach to Host Mobility,” ACM MOBICOM, pp. 155-156, Boston, U.S.A., MA, 2000.
[6] V. Phan and A. Karmouch, “Mobile Software Agents: Ar Overview” IEEE Communication Magazine, Vol. 36, No. 7, pp. 27-37, 1998.
[7] http://www.jxta.org/

Koichi Takasugi
Research Enginneer, Network Intelligence Laboratory, NTT Network Innovation Laboratories.
He received the B.E. degree in computer science from Tokyo Institute of Technology, Tokyo in 1995 and the M.E. degree in information science from Japan Advanced Institute of Science and Technology, Nomi, Ishikawa in 1997. In 1997, he joined NTT Wireless System Laboratories, Yokosuka, Japan. He is a member of IEEE, the Institute of Electronics, Information, and Communication Engineers (IEICE) of Japan, and the Information Processing Society (IPSJ) of Japan.

Motonori Nakamura
Senior Research Engineer, Network Intelligence Laboratory, NTT Network Innovation Laboratories.
He received the B.E. and M.E. degrees from Nagoya University, Nagoya, Japan in 1990 and 1992, respectively. In 1992, he joined NTT Communication Switching Laboratories, Tokyo, Japan. He is a member of IEICE and IPSJ.

Minoru Kubota
Executive Manager, Network Intelligence Laboratory, NTT Network Innovation Laboratories.
He received the B.E., M.E., and Ph.D. degrees in mathematical engineering from the University of Tokyo, Tokyo in 1978, 1980, and 1997, resepctively. In 1980, he joined the Musashino Electrical Communications Laboratories, Nippon Telegraph and Telephone Public Corporation (now NTT), Tokyo, Japan. He is a member of IEEE, IEICE, and IPSJ.