Protocol Design for All-IP Computer Architecture

Kouji Okada
okada@sfc.wide.ad.jp

Keisuke Muda
muda@sfc.wide.ad.jp

Yoshifumi Nishida
nishida@sfc.wide.ad.jp

Hideaki Yoshifuji
yoshfuji@sfc.wide.ad.jp

Ryuji Wakikawa
ryuji@sfc.wide.ad.jp

Jun Murai
jun@sfc.wide.ad.jp

Graduate School of Media and Governance, Keio University
Faculty of Environmental Information, Keio University
Keio University Shonan Fujisawa Campus
5322 Endo, Fujisawa Kanagawa, 252-8520 Japan

Abstract

In this paper, we present a new computer architecture called All-IP computer architecture, which realize a flexible computer re-configuration environment via IP networks. Our new IP-based protocols connect devices to the host computer that were previously limited to a physical attachment. To construct All-IP computer environment, we designed a protocol to manage and discover device resources on IP networks. We also evaluated a protocol called USB/IP, which is expected to be one of the device interconnection protocols in the All-IP computer architecture.

1. Introduction

Mobile computing enables users to communicate with each other anywhere, anytime with their own mobile terminals. At the same time, users construct several application environments that depends on the location, such as the computers at home or workspace, or even cell phones with poor computing resources in the train. Thus users currently use multiple application environment for each location. In such environment, when users want to continue the work of having done at home on the train, users have to copy the data from their desktop computers to laptops or cellphones, and activate the application again. The computer environment today is restricted to its physical "box", and existing methods cannot integrate several application environments.

At the same time, approaches to utilize TCP/IP protocol stack for the communication between each computer device, such as DVTS[1], iSCSI[2], or iUSB[3] are proposed. These technologies realize the virtual computers that consist of IP networking computer devices on the Global IP networks. Those virtual computers can utilize the enormous amount of the processing capability of supercomputers on the Internet.

In this paper, we will show an architecture called All-IP computer architecture, which realizes the continuous application environment. the All-IP computers consist of IP networking computer devices connected to the global IP network. Users request All-IP computer environment according to their context. For example, When a user is running heavy task application such as scientific calculations, the user needs high calculation capability of supercomputers. When the user is watching movies at home, the user might request a rich screen and sound environment operated by a remote controller. Thus All-IP computers are constructed dynamically based on users' requests without connecting or disconnecting the physical connection cables.

2. Existing Computer Architecture

Computers are defined as the sets of computing devices and the unified controlling software called "operating systems". In this section, we discuss the existing computer architecture and how existing computer devices are controlled by operating systems.

2.1. Device types

Computers consist of several computing devices. Such computer devices can be categorized into five types: input devices, output devices, central processing unit, main memory units, and secondary storage devices.

Compuer devices are controlled by the device drivers implemented on the operation systems (OS's) (Figure 1(a)). In the current OS architecture, device drivers are implemented with OS-common application programming interfaces and behave as the universal device control programs for the same type of devices. Through the device drivers OS's can handle device common procedure of each device without any conflicts.

2.2. Device driver functionalities

Device drivers are the device controller software implemented on the OS's. Device drivers take part of device control functionarities, such as device initializations, device control methods, input-output processing, and interruptioin handlings.

Through the initialization, devices are loaded on the system and initial device configurations are set to the devices. At the first step of the device initialization, OS's scan the computer buses to probe available devices. After detecting new devices, OS's load device drivers that corresponds with each device. Devices are available on OS's after the device drivers initialize them.

Device controllers take part of controlling device unique behaviors. For examples, USB device controllers understand USB specific communication protocols and SCSI controllers for SCSI devices. The device drivers re-configure the device specific parameters set to the devices or send device specific instructions to the devices. Through the device controllers, OS's are able to control various types of devices.

The input-output function intermediates data between devices and OS's. This function enables the OS's to write control instructions to devices, and read input information from devices.

Device drivers inform OS's of the interruptions from devices. The interruptions are high priority control instructions, such as error handling commands. The interruptions have to be proceeded by OS's appropriately.

3. IP networking computer devices

As we described in the section 2, physical devices are abstracted as device drivers on OS's in the existing computer architecture. By taking charge of managing direct device accesses exclusively, OS's provide common device access methods to applications running on the OS's.

In All-IP computer architecuture, however, each computer devices are connected to IP networks, and can be accessed from any OS's connected to the Internet, which can cause device usage inconsistencies.

In this section, we describe the All-IP computer architecture overview and requirements to construct All-IP computer architecture.

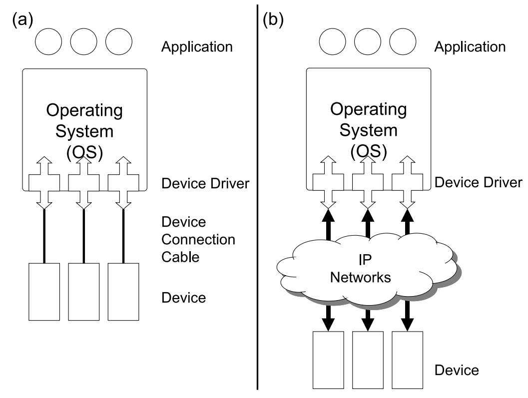

Figure 1. Device driver models

3.1. Overview of the All-IP computer architecture

The All-IP computers are the computers which consists of IP networking devices controlled by host computers. An All-IP computer consists of several computer devices and a central management computer called "host computer". Each computer devices perform as a single IP networking node that runs tiny OS's with the IP networking capability. Such computer devices are operated by the host computers running multitask OS's to control devices over IP networks, and provide user application environment (Figure 1(b)).

3.2. Requirements for the All-IP computer architecture

To construct All-IP computer, we have to consider requirements listed below.

- Device discovery
  In the existing computer architecture, the device initialization is done after scanning the computer buses to detect new devices by the OS. In the All-IP computer architecture, however, it is not realistic for a single OS to scan the global Internet for each device. We must overcome the obstacles of discovering the available devices over the Internet.
- Address assignment
  In IP networks, all the nodes are identified by their unique IP addresses. Therefore, IP addresses have to be assigned to each IP networking computer device, which is also an individual internet node. The All-IP computer architecture needs huge quantities of IP addresses, even though the IPv4 address depletion problem is issued. Therefore, it is required to utilize IPv6 for our architecture.
- Network delay and losses
  In the existing computer architecture, devices are con-

nected to each other directly through computer buses. Therefore, the scale of computer internal bus networks are small, and the network delay and packet losses are not fatal substantially. However, because the Internet covers all over the world, the communication delay between IP networking computer devices might exceed the device assumed delay and device control messages between OS's and devices might be lost unexpectedly.

- Communication reliability
  In IP networks, data are transmitted on the best-effort basis, which does not provide full reliability. In current IP networks, TCP is the main data transport protocol. However, the overhead of TCP error recovery mechanism is quite high, and it causes increasing delay and jitter. Thus, it is difficult to realize realtime data transfers for the device to device communications with TCP. The data transport protocol which provides the communication stability is necessary for the All-IP computer architecture.

4. All-IP computer architecture

In this section, we will show the detail of All-IP computer architecture.

4.1. Framework

Figure 2 shows the framework of the All-IP computer architecture.

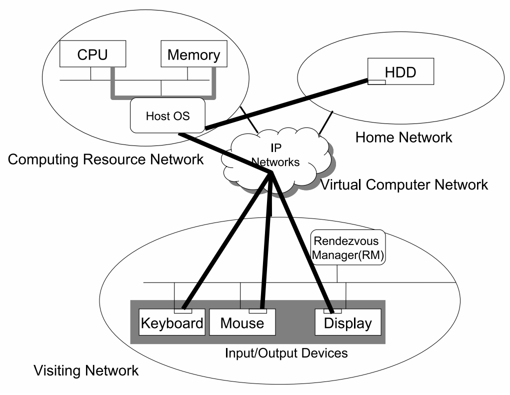

Figure 2. All-IP Computer Architecture

Users carry small computer called "user tags" which stores authentication information and device configurations. When visiting other networks, users attach their user tags to the foreign networks. RFID's and cell phones are examples of the user tags. This RM also coordinates the All-IP computer setups. We will describe the detail of how RM constructs an All-IP computer in section 4.3 .

After a user tag is attached to a network, the user tag scans the visited network to discover a RM. When the user tag is non-computing devices such as RFID's, a Rendezvous Proxy (RP), which proxies user requests, reads the device initial configuration and authentication information, and take the procedure after RM search in place of the user tag.

After the initial setup, users are able to access the All-IP computers. Each devices are connected to the host computer directly over IP networks; they are not intermediated by RMs (Figure 2). When the All-IP computer is set up, the RM informs users of the location of input and output devices for the computers.

4.2. Device Identification and Management

The information of devices connected to IP networks are registered to device management server called Rendezvous Manager (RM). RMs manages the information of the devices located in their operation domain. The information which RM's manage are listed below.

- Device name
  Device types. "Keyboard", "Display", etc...
- Device Address
  A global address assigned to the device.
- Device availability
  The operating conditions of the device.
- Physical location of device
  Required only for input and output devices. At the construction of new All-IP computers, these information are used for grouping input and output devices located nearby. After the constructions of new computers, the information are used to inform users of where to use the computers.

The configuration of devices such as CPU, memory and I/O devices, and the information of Host OS's are configured by the network operators statically. On the other hand, information about the secondary memory devices, such as hard disks, and the input device information when users carry their own keyboard or mouse, are updated dynamically based on user requests. RM's release those information when the user stops to use the All-IP computer. At a time the user shutdowns the All-IP computer, the All-IP computer sends a device release message to the RM, and the RM marks the device status as free.

4.3. Bootstrap

The figure 3 shows the All-IP computer bootstrap sequence.

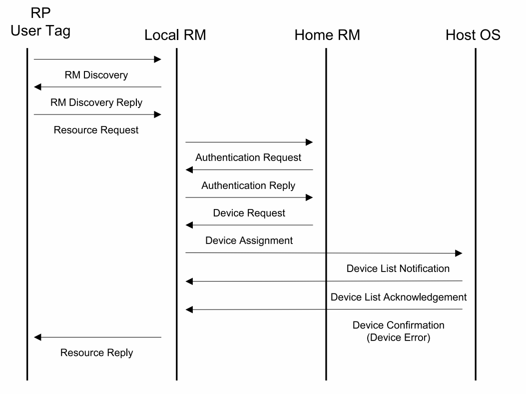

Figure 3. Bootstrap Sequence

First, users attach their user tags to visiting networks, and the user tags send RM discovery requests to the anycast address for RM discoveries to get the IP address of the RM on the visiting networks. When the user tags are non-computing devices such as RFID's, RM discoveries are proxied by RP's with RFID readers. In that case, RP's first read users' All-IP computer configuration and user authentication information from the user tags, and then the RP's proxy the RM discoveries. RM discovery messages only contain the message source IP addresses. User tags with computing resources, cell phones for example, send RM discovery requests by themselves. Upon receiving RM discovery requests, RM's send back RM discovery acknowledgement messages to the request message sources. The RM's located on the visiting networks are called "local RM". Then user tags or RP's send resource request messages, which include users' authenticatig information and computer configuration, to the local RM.

The "local RM" sends a user authentication request to user's home RM with the certification included in the resource request. The "home RM" is the one which is located on the user's home network. The home RM validates the user certificate and sends the authentication response to the local RM. If the authentication is succeeded, the local RM requests the computer resources managed by the home RM based on the configuration included in the resource requests, and home RM allocates the resources if the device statuses are marked as free. The local RM allocates to the user the devices not specified by the configuration included in the resource request.

Then the local RM negotiates with a host computer to allocate a host OS to the user. The host OS handles each IP networking devices and provide an application environment to users. First, the local RM sends a device information list of Input/Output devices, storage devices and other peripheral devices to a host OS. The device information list contains the device configurations, such as IP addresses and the protocol to control the device. Upon receiving the device list, the host OS connects each device on the list to host OS itself according to the information described on the list.

After completing the host OS allocation, the local RM sends a resource acknowledgement message which indicates the location of the I/O devices. The user tag or the RP displays the location so that the user can know where to use his or her virutual computer.

4.4. Migration

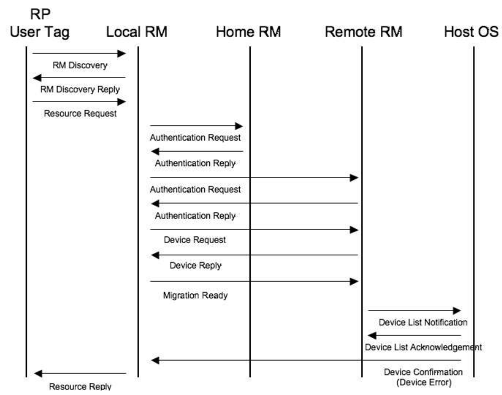

Figure 4. Migration Sequence

The figure 4 shows the All-IP computer bootstrap sequence.

The user migration occurs when a user suspend an All-IP computer and resume to use the computer on another visiting network. At a user migration, the home RM authenticates users based on the procedure described in section 4.3. Then, the remote RM located on the network on which the user construct an All-IP computer authenticates the user and authorizes the user to use the All-IP computer. After the authentication, the local RM requests the device information list to the remote RM. The device request message can contain the required device lists. Upon receiving a device information list request, the remote RM sends a device reply message to the local RM, which includes the information list of the devices the user previously used. Then, the local RM sends a "migration ready" message which specify information about devices located on new visiting network, such as input/output devices. The remote RM sends a new device list with a device list message to the host OS. After the host OS's device confirmation, the host OS sends a device list acknowledgement message to inform local RM of the device verification result. Finally, the local RM announces the input/output devices' locations.

5. Evaluation

In this section, we present USB/IP performance evaluation.

5.1. USB/IP

USB/IP[4] is a technology which realizes USB data communication over IP networks. In the USB/IP framework, the computer to which USB devices are directly connected are defined as a "USB/IP server" while the computer on which users utilize USB devices is called a "USB/IP client". The USB/IP servers capture the USB data at the USB host controller driver and encapsulate USB data into IP packet format. Then the servers send the IP packet that contains USB data to the USB/IP clients' host controller over IP networks, and the clients can handle USB data as if USB devices are connected to the client themselves directly.

We defined this USB/IP as one of the core technologies of our prototype system, and evaluated the USB/IP availability.

5.2. USB/IP Evaluation

Figure 5 shows our test environment.

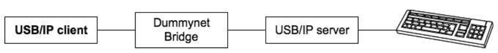

Figure 5. Evaluation environment

The evaluation environment consists of three computers, the USB/IP server which a USB keyboard is directly attached to, the USB/IP client, and the dummynet bridge for making delays by 10 milliseconds basis. We input characters with the USB keyboard and capture the IP packet at the client. Then we calculate the time difference between the keyboard "pressed" and "released" events.

Figure 6 shows the evaluation result. The bridge delay in the graph indicates the one way delay on the network. When the bridge delay was set to more than 40 milliseconds (RTT 80 milliseconds), the time-gap between "pressed" and "released" events depends on the network's round trip time, while the time-gap was almost constant when we set the network delay at less than 50 milliseconds. We estimate that this is because the delay depends on the USB event schedular when the network delay is set to less than 50 milliseconds. Thus the assumed delay of USB keyboards seems to be less than 50 milliseconds. When the network delay was

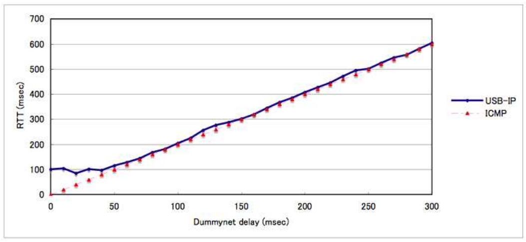

Figure 6. USB-IP Performance Evaluation

configured to more than 60 milliseconds, we could feel the delay when we type the keyboard, and when the delay was set to 230 milliseconds, the keyboard events began to be dropped by the USB clients. Thus it is desirable that the network delay between the host OS and USB devices are less than 50 milliseconds if the device needs the real-time communication.

6. Conclusion

In this paper, we described the All-IP computer architecture which realizes an IP networked virtual computer environment. All-IP computer architecture makes it possible to re-configure the computer environment flexibly. We also evaluated the network requirements using USB/IP.

7. Acknowledgment

We would like to show our appreciation to Dr. Rodney Van Meter, who gave us valuable comments and suggestions to our paper.

References

[1] A. Ogawa, K. Kobayashi, K. Sugiura, O. Nakamura, J. Murai
Design and implementation of DV stream over Internet Internet Workshop, 1999. IWS 99, February, 1999

[2] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, E. Zeidner
Internet Small Computer Systems Interface (iSCSI)
Request for Comments (Draft Standard), Internet Engineering Task Force, April 2004.

[3] K. Muda, Y. Nishida, K. Okada, H. Yoshifuji, R. Wakikawa
Problem Statement of Internet Universal Serial Bus (iUSB) (work in progress, draft-muda-iusb-ps-00).
Internet Draft, Internet Engineering Task Force,July 2, 2007.

[4] T. Hirofuchi, E. Kawai, K. Fujikawa, H .Sunahara
USB/IP - a Peripheral Bus Extension for Device Sharing over IP Networ k Proceedings of the 2005, USENIX. Annual Technical Conference, April, 2005