A conceptual framework for an IP networked 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

The networks are getting faster as 40G ethernet products appear and are expected to increase their bandwidth to terabits in the near future. Computers located all over the world are connected by those high-speed networks, and the communication between those computers are realized with lower latency and high bandwidth. In this paper, we propose a new computer architecture called "All-IP computer architecture".

The All-IP computer architecture re-configure the structure of the computer itself by attaching or detaching the computer devices connected to IP networks dynamically. Each device equips with the IP protocol and has a unique IP addresses as a device identifier. The devices are attached to operating systems running on the host computer over the IP networks.

In this paper, we developed a new computer architecture and evaluated protocols for the communication between computer devices.

1. Introduction

Mobile computing enables users to communicate with each other anywhere, anytime with their own mobile terminals. On the other hand, users construct several application environments dependant on the location such as the computers at home or workspace, or even cell phones 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 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 iSCSI[1], iUSB[2] or DVTS 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 on the All-IP computers that consist of IP networking computer devices.

2. Existing Computer Architecture

Computer are defined as the sets of computing devices and the unified controlling software, 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 computing devices can be categorized into five types, input devices, output devices, central processing unit, main memory units and secondary storage devices.

Such 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 universal device control programs for the same

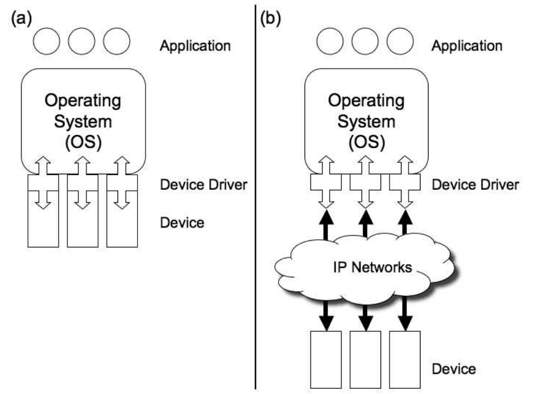

Figure 1. Device driver models

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, device initializations, device control, inputs and outputs and interruptioins.

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 obtain 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. 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

In this section, we describe the All-IP computer architecture overview and the problem statements.

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 being run on the OS's. In this paper, we propose an computer architecture model in which OS's control devices via the global IP network.

3.1. Overview

The All-IP computers are the computers which consists of IP networking devices controlled by host computers. In All-IP computer architecture, All-IP computers are constructed dynamically based on users' requests. 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 are watching movies at home, the user might request a rich screen and sound environment operated by a remote controller.

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 running 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. Problem statement

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

- Device initialization
  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.
- 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 individual internet node. The All-IP computer architecture needs huge quantities of IP addresses, while 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 connected 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, which cause 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.

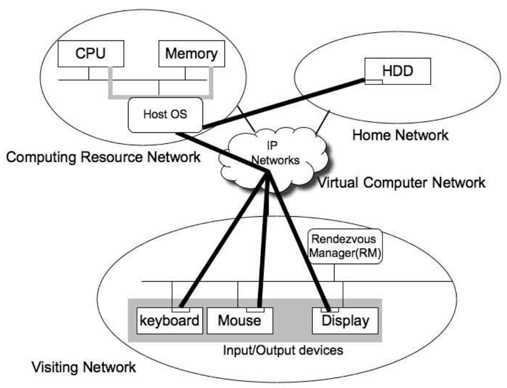

Figure 2. All-IP Computer Architecture

In out architecture, users carry small computer called "user tags" which stores authentication information and device configurations. When visiting other networks, users attach their tags to the foreign networks. RFID's and cell phones are examples of the user tags. The information of devices connected to IP networks are registered to device management server called Rendezvous Manager (RM) beforehand. This RM also coordinates the All-IP computer setups as described in 4.

After a user tag is attached to a network, an IP address is assigned to the user tag. After the address assignment, 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 RM discover completed, the user tag sends the computer setup request. When a RM receives a computer setup request, the RM searches its device lists for available computer resources such as CPUs, hard disks and input/output devices, and assigns those computing resources to the user. At the same time the RM assignes a host OS to the user if the host OS information is not specified in the device request. Those devices are controlled by the multi-task host OS that is ran on a host computer. The RM assigns a host OS for a user dynamically.

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

4.2. Computer ID

Existing computers communicate with each other using IP addresses assigned to their network interfaces. The All-IP computers are the computers without physical entities. Thus identifiers for the All-IP computers which aggregates the computing devices are necessary. There are two assumption for this identifier: they must be global unique, and must be assigned to devices dynamically.

In All-IP computer architecture, IP addresses for the All-IP computers are assigned by RM's. When a RM construct a new All-IP computer, the RM's assign an EUI-64 for new virtual computers. Global IPv6 addresses for the All-IP computers are created from the EUI-64 and the global prefixes managed by the RM's. The global IP addresses are called Computer ID (CID) in our architecture.

4.3. Rendezvous Manager

RMs manages the device information of the devices located in their operation domain. The information which RM's manage are listed below.

- Device name
  Device types. "Keyboard", "Display"
- Device Address
  A global address assigned to the device.

- Device availability
  The operating condition of the device. When the device is already used by an other computer, this field indicates the CID information of the All-IP computer.
- Device physical location
  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.

RM's manage the information listed above for each device. 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, the input device information, if 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 an All-IP computer, the computer sends a device release message to the RM, and the RM marks the device status as free.

Currently, only one RM manage the All-IP computer setup in our system. The All-IP computer setup by multiple RMs enables a flexible computer configurations. The multiple RM cooperation is a future work.

4.4. Bootstrap

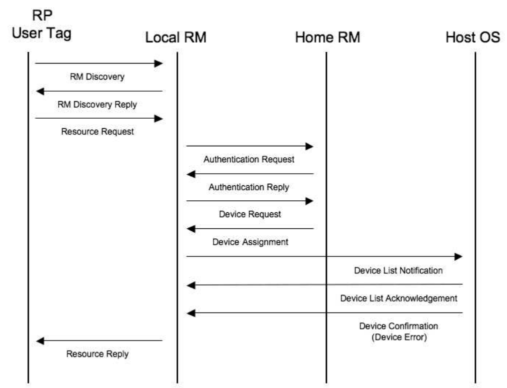

Figure 3. Bootstrap Sequence

The figure 3 shows the All-IP computer 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. When the user tags are non-computing devices such as RFID's, RM discoveries are proxied by RP's with RFID readers. RP's first read users' All-IP computer configuration stored in the user tags, and then the RP's proxy the RM discoveries. User tags with computing resources, cell phones for example, send RM discovery requests by themselves. Upon receiving the discovery requests, RM's send RM discovery acknowledgement messages to the request message sources. The RM's located on the foreign networks are called "local RM". Then user tags or RPs send resource request messages to the local RM. The resource requests contain the public keys of users.

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 succeeded, the local RM requests the computer resources managed by the home RM, and home RM allocates the resources if the device statuses are marked as free.

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 such as keyboards, mice and displays, storage devices and other peripheral devices. The device information list contains the configuration such as IP addresses and the protocol to control the device. Upon receiving a device list, the host OS connect 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.

4.5. Migration

The user migration occurs when a user suspend an All-IP computer and resume to use the computer on an other visited network. At a user migration, the home RM authenticates users based on the procedure described in Figure 4. 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 include the required device lists. Upon receiving device information list requests, the remote RM sends to the local RM a device reply message, which includes the information list of the devices the user previously used, then the local RM

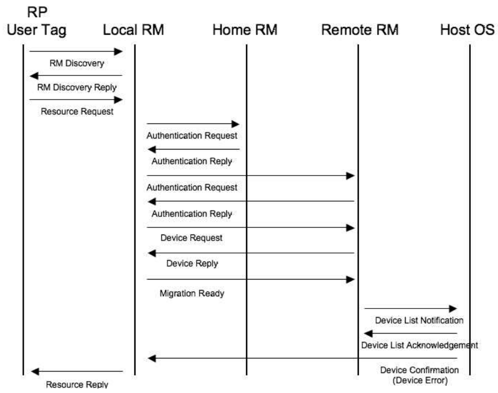

Figure 4. Migration Sequence

send 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 send a device list acknowledgement message to inform local RM of the device verification result. Finally, the local RM announces the input/output devices' location.

5. Evaluation

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

5.1. USB/IP

USB/IP is a technology which realize 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.

The test environment consists of three computers: the USB/IP server which a USB keyboard is directly attached

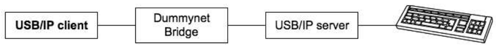

Figure 5. Evaluation environment

to, the USB/IP client, and the dummynet bridge for making delays by 10 milliseconds. 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.

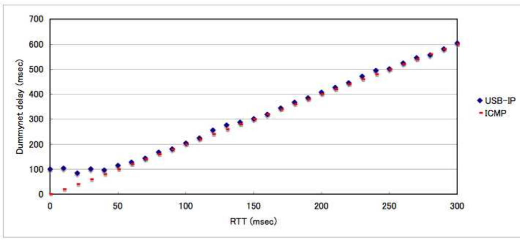

Figure 6. USB-IP Performance Evaluation

When the bridge delay was set to more than 40 miliseconds, the time-gap between two events depends on the network's round trip time, because of the relation between USB data transmission and its acknowledge. However, when we set the network delay less than 50 miliseconds, the time-gap was steady. We guess this is because the delay depends on the USB event schedular, and the USB assumed delay is less than 50 miliseconds. When the network delay was configured to values more than 60 miliseconds, we could feel the delay when we type the keyboard, and when the delay was set to 230 miliseconds, 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 miliseconds 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.

References

[1] 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.

[2] 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.