TCP/IP VULNERABILITIES OF EMBEDDED-SYSTEM IMPLEMENTATIONS

T. Sugimoto, M. Ishii, T. Masuda, T. Ohata, T. Sakamoto, R. Tanaka,
Japan Synchrotron Radiation Research Institute (JASRI/SPring–8),
1–1–1 Kouto, Sayo, Hyogo 679–5198, Japan

Abstract

Recent accelerator-control systems are based on a TCP/IP network. Embedded devices equipped with an Ethernet interface are often used in control systems. At SPring–8, since increasing the number of network-connected devices, many network problems have been caused by vulnerabilities. To improve the reliability of the accelerator-control system, the vulnerabilities of embedded devices are investigated. As a result, some vulnerabilities in TCP/IP implementations were found in the motor-control unit, which is one of the embedded devices used in SPring–8. Our preliminary plans to fix the vulnerabilities and thus improve the reliability of the control system are also presented.

INTRODUCTION

Historically, accelerator-control devices used to communicate with each other using hardwired channels. Then standard interfaces such as RS–232C and GPIB were substituted for the hardwired channels. Since these legacy interfaces provide a one-to-one (or one-to-*few*), short-range, narrow bandwidth communication channel, it is difficult to use them to control large-scale accelerators. Recently, these legacy-interface-based devices have been replaced with devices equipped with Ethernet interfaces. The Ethernet allows us to operate large-scale accelerator-control systems, because it has a many-to-many connection topology, long-range reachability, and a wider bandwidth than those of legacy devices. These Ethernet devices, in many cases, support the TCP/IP protocol.

TCP/IP is a *de facto* standard network-communication protocol. By popularizing the TCP/IP technology, network-connecting functions are supported by many devices; not only computers but also embedded devices. Since embedded devices are designed with limited hardware resources, these devices often use a subset of the TCP/IP components. The limited hardware resources and the use of a subset of components, therefore, cause many problems such as vulnerabilities in TCP/IP implementation.

At SPring–8, the accelerator-control system is based on the TCP/IP protocol. A number of network-connected embedded devices are used to monitor and control the accelerators including digital multimeters, oscilloscopes, and multichannel analyzers. Figure 1 shows a history of the number of registered hosts in the SPring–8 control segment. The number of hosts was only 342 in January 2001[1] but had increased to 1141 by March 2008. By increasing the number of devices with latent vulnerabilities, many problems have arisen such as packet flooding and unexpected delays in response. One of the most serious problem related to the embedded devices in the control system has been the hang-up of pulse-motor controller units (MCUs)[2]. This causes interruptions to the operation of injection accelerators. These events have frequently occurred since 2007, when the number of hosts was rapidly increased. It is estimated that the number of hang-up events is correlated with the network traffic.

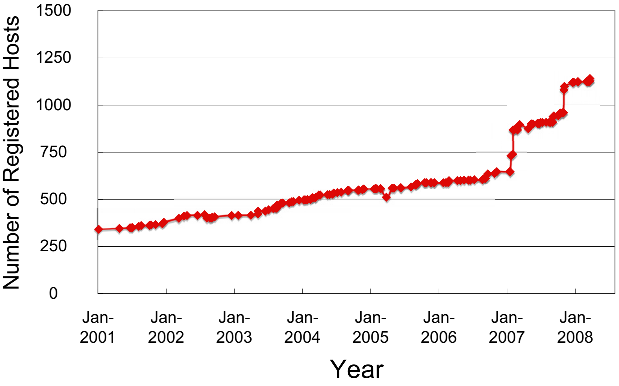

Figure 1: Number of registered hosts in the SPring–8 control system. Note that the number of hosts was rapidly increased in February 2007. Since then, hang-ups of MCUs have frequently occurred.

In this study we aimed to improve the reliability of the accelerator-control system in SPring–8. To reduce the interruption time, it is necessary to investigate the network vulnerabilities of embedded devices.

INVESTIGATION OF VULNERABILITIES IN EMBEDDED DEVICES

In the present study, embedded devices used in the accelerator control of SPring–8 are investigated. An experimental network segment separated from the actual control segment was built. By loading the devices with simulated network traffic, the vulnerabilities of the devices were inspected.

In the present paper, we show the TCP/IP vulnerability in the MCU. The basic architecture of the MCU is an embedded device with an SH–4 CPU and a NORTi4/μITRON-based real-time operating system. A number of MCUs are used in SPring–8 as beam slit controllers, RF phase ad-

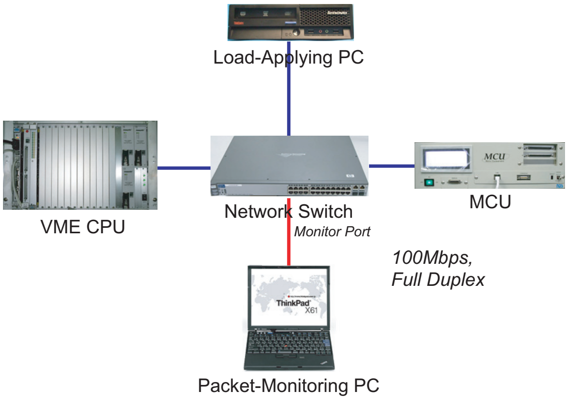

Figure 2: Experimental network environment. A VME CPU, two PCs, and a MCU are connected to a network switch with a 100 Mbps full-duplex mode.

justers and attenuators, and wire-grid monitors. The reasons why we chose the MCU for the first target of investigation as follows. 1) The MCU was developed at SPring–8; thus, fixing the implementation is relatively easy compared with fixing commercially available devices. 2) The failure of network communication in the MCU occurs more frequently than in other devices — once a week on average. 3) Once a communication failure occurs, the MCU must be restarted and initialized under the condition that the injection accelerator has stopped. Therefore, the top-up operation of the storage ring is interrupted.

The investigation procedure and methods are described below.

Experimental Environment

The experiment was performed in the separate network environment shown in Fig. 2. The experimental setup consisted of a network switch (HP ProCurve Switch 2626-PWR), a VME CPU board (Densan DVE-686/50), two PCs for applying load and monitoring packets (IBM ThinkCentre S50, Lenovo ThinkPad X61), and the MCU. Each channel between the network switch and the devices was connected with a 100 Mega bit per second (Mbps) full-duplex mode. The monitoring PC was connected to the monitor port of the network switch.

As described before, hang-up of the MCU is suspected to be caused by increased traffic, particularly broadcast packets. Therefore we investigated traffic capability of the MCU; maximum load and continuous load capabilities.

Maximum Load Capability

To investigate the maximum load capability, the MCU was loaded with an extremely large volume of traffic using the "Burst Ping" method. The load-applying PC sent a number of ping packets to the MCU at a rate of about 2000 packets per second (pps). Both the echo-request and echo-reply packets were recorded by the monitoring PC.

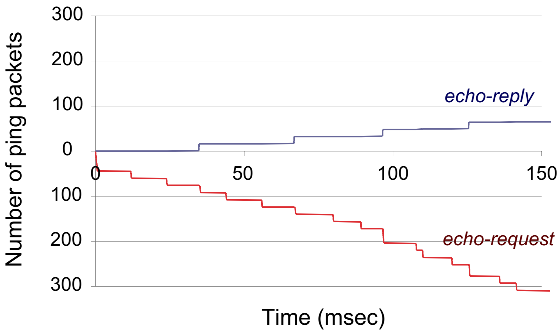

Figure 3: Number of ping packets sent to the MCU. The vertical axis shows the number of echo-request and echo-reply packets. The horizontal axis shows time in milliseconds.

Figure 3 shows the number of ping packets (ICMP echo-request and echo-reply) sent from and to the MCU. By analyzing the recorded data, some characteristic behaviors were found.

1. The MCU replies to an echo-request within 30 msec. This behavior is one of the performances specified in the real-time OS.
2. The number of echo-reply packets that can be sent within the 30 msec period is only 16.
3. When more than 16 echo-request packets are received, they are buffered and sent during the next 30 msec period.
4. In the case of receiving echo-request packets continuously, the receive buffer overflows, and the overflow packets are not replied to.

According to these behaviors, the packet-processing capability of the MCU is up to 533 pps.

Continuous Load Capability

Since it is possible that the continuous buffer overflow causes network problems in the MCU, its continuous load capability was also investigated. For this investigation, the "SYN Flooding" method was applied. While the VME and MCU communicate with each other, SYN packets were sent to the MCU continuously. By varying the rate at which SYN packets were sent, the normality of the intercommunication was monitored.

1. When the rate of SYN packets is less than or equal to 100 pps, proper communications maintained.
2. If the rate of SYN packets exceeds 100 pps, the SYN-flooding load interferes with the communication, and the control operations between the VME and MCU fail.

Thus, the upper limit of continuous SYN packet capability turned out to be 100 pps. Note that not only listening ports but also closed ports are affected by the SYN flooding.

Vulnerabilities in the MCU

According to the results of these investigations on MCU, the network-processing capability of the MCU appears to be poor. Assuming a large packet (1518 bytes), the bandwidth capability of an MCU is only 1.2 Mbps. This is much less than the connection bandwidth (100 Mbps) between the network switch and the MCU. Thus, the MCU is vulnerable to heavy traffic.

In the control network of the SPring–8, many broadcast packets flow including the Network Information Service (NIS), Syslog, and NetBIOS over TCP/IP (NBT). Since the capability of the MCU is affected by packets received at closed ports, broadcast packets with any destination port are also harmful to MCUs. Furthermore, the Windows PCs required by recent instruments send many NBT packets. By increasing the number of these broadcast-sending hosts, the network environment becomes more severe for MCUs.

PLAN TO IMPROVE RELIABILITY OF CONTROL SYSTEM

To improve the reliability of the control system in SPring–8, two approaches are considered. One is to fix the MCU implementation, and the another to fix the network environment. In this section, our preliminary plans for improvement are described.

Improvement of MCU: Higher Performance

The first approach is to fix the base system of the MCU. The input/output (I/O) bandwidth of flash memory is much narrower than that of random access memory (RAM). Since the MCU is a flash-based system, it is assumed that the hardware I/O limits the overall performance of the MCU. To confirm this assumption, the packet capability of an MCU with a RAM-based OS was examined. It was found that the continuous load capability with a RAM-based OS is clearly improved, whereas the maximum load capability remains the same value. Long-term tests on a RAM-based MCU are now being performed.

We also considered changing the OS of the MCU. Since NORTi4 is a real-time OS, its maximum load capability (16 packets per 30 msec) may be limited by the OS. Using another OS, load capability against the number of burst ping packets was investigated. It was found that an MCU with SH–Linux has a packet processing capacity of more than 2000 pps. Therefore, the maximum load capability is not limited by the hardware but by the real-time OS. To improve the maximum load capability, we consider it effective to alter the OS from NORTi4 to another OS such as Linux, if exact real-time performance is not required.

Improvement of Network: Greater Sophistication

The accelerator-control network in SPring–8 is based on a 21-bit masked segment (2048 addresses), and about 1200 hosts are currently installed in this segment. By increasing the number of hosts in the control network, MCUs are required to handle many broadcast packets. If it is supposed that each of the 2000 hosts in the 21-bit masked segment sends broadcast packets at a frequency of 20 sec per packet (0.05 pps), the continuous load capability is exceeded. The broadcast domain is hence too large for MCUs.

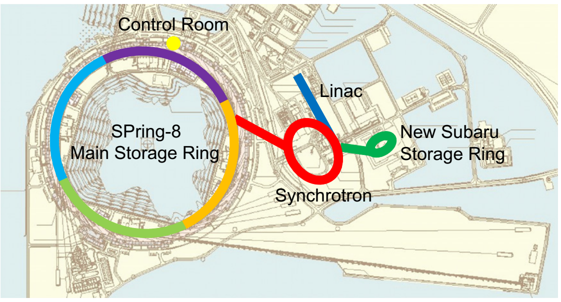

Figure 4: Preliminary plan to segment network into applicable broadcast domains. The present control network will be segmented into eight domains, each with 23-bit masked network: control room, linac, synchrotron, 4× main storage ring, and New Subaru.

It is important to segregate the network into applicable broadcast domains. Thus, we are beginning to plan the modification of network segments. Figure 4 shows a preliminary plan for a future network. Each broadcast domain is segmented into a 23-bit mask. In the case of a 23-bit masked segment, the acceptable frequency of broadcasts from each host can be reduced to 5 sec per packet (0.2 pps). Moreover, it may be effective to segregate vulnerable devices into another network.

SUMMARY

A number of network-connected devices are used in recent accelerator-control systems. At SPring–8, network problems have occurred as a result of some embedded devices. To improve the reliability of the accelerator-control system, the vulnerabilities of the MCU were investigated. As a result, we found vulnerabilities in TCP/IP implementation in the MCU: one is caused by hardware, and another is caused by the operating system. We are now planning to refine both the MCU itself and the network environment.

In the present paper, we described the vulnerabilities of the MCU. Network problems have also occurred in other devices at SPring–8: including digital multimeters, oscilloscopes, multichannel analyzers. These problems are not specific to SPring–8, but may be occurred in other facilities. We will also investigate the vulnerabilities of these devices and thus improve the reliability of the control system.

REFERENCES

[1] T. Fukui et al., Proceedings of ICALEPCS'01, THDT005 (2002).

[2] T. Masuda et al., Proceedings of PCaPAC2005, WEP30 (2005).