Development of a distributed backup system and a recovery system for telemetric seismic data

Kenji Uehira

Institute of Seismology and Volcanology, Kyushu University, Shimabara 855-0843, Japan

(Received April 30, 2008; Revised September 29, 2008; Accepted September 30, 2008; Online published February 18, 2009)

Fast, low-cost Internet lines have recently become available that rapidly transfer seismometer data. However, as in any fast transfer system, gaps frequently occur during data transmission. There are many causes for missing data packets at the receiving end, where is the data center side. I have developed a system that is meant to solve this problem. When the telemetry or the system is down at the data center end, data recovery is possible if the data are stored locally at each station. This means that the missing data can be re-sent after the required repairs have been made at the data center end. Therefore, I developed a concept of a distributed backup system that can store seismometer data for over 1 year locally at each station and developed a new protocol called "WIN Raw Data Recovery Protocol (WRRP)" for the efficient re-transmission of missing data.
Key words: WIN Raw Data Recovery Protocol (WRRP), distributed backup system, microserver, WIN system, stateful.

1. Introduction

As Internet connections have become ubiquitous, along with fast and inexpensive Internet protocol (IP) lines, such as the Integrated Services Digital Network (ISDN), the Asymmetric Digital Subscriber Line (ADSL), and fiber-optic networks, the use of leased lines and satellite lines is being replaced by IP lines for the telemetry of seismometer data.

When seismic data are sent by telemetry from stations, there are many different reasons for missing data packets at the data center end, such as a breakdown of the data center collection systems, or trouble with the telemetry lines, or the data acquisition system or the power supply is down at the station. In the first two cases, it is theoretically possible to recover these data if the data are stored distributively at each station side, because the missing data can be re-sent at a later time. However, this is currently difficult to accomplish because the telemetry lines that are used, such as radio lines, leased lines, or satellite lines, are slow and have inadequate bandwidth to simultaneously broadcast concurrent real-time data in addition to the backfill data. Recently, it has become possible to construct more robust data collection systems that do not depend on the availability of telemetry.

I have developed a system that can store the waveform data for over 1 year at the station side using a microserver and developed a protocol called "WIN Raw Data Recovery Protocol (WRRP)" for the efficient transmission of missing data. This protocol deals with an environment that needs complete data, such as networks for research purposes, which demand more complete holdings, rather than a real-time monitoring network like the Earthquake Early Warning (e.g., Kamigaichi, 2004; Hoshiba *et al.*, 2008) by the Japan Meteorological Agency (JMA).

2. Data Storage System at the Station Side End

Seismic data are stored using a microserver for storage because environmental conditions, such as temperature, humidity, and dust in the station environment, can inflict a great deal of mechanical stress on personal computers and servers. In addition, it is not necessary to have a powerful computer, because the quantity of data at one station is much less than that of the sum of all data at the data center. A compact flash (CF) card is used for storage, since it has no moving parts and is resistant to severe environmental conditions. Although the CF card has a predefined capacity—approximately 100,000 erase/write cycles per one block—recent CF card controllers distribute data to every block intelligently in order to not concentrate erase/write cycles on a specific block. In fact, no trouble has been reported during the 4 years that status files were updated once per minute. An OpenBlockS266 of Plat'Home Co., Ltd. (http://www.plathome.com/) was used as the microserver. Table 1 shows the specifications for the OpenBlockS266. As of January 2008, CF cards ranging from 128 MB to 8 GB can be installed.

The WIN System software (e.g., Urabe and Tsukada, 1991; Uehira and Urabe, 1994; Urabe, 1994; Urabe and Takanami, 1994; Uehira, 2001; http://eoc.eri.u-tokyo.ac.jp/WIN/) was used for data acquisition and storage. The WIN system deals with multichannel seismic waveform data and consists of many programs that run on UNIX. The WIN system defines a waveform data format called the "WIN format". This format has specifications, such as: (1) a variable length second block with a time label in 1-s increments; (2) a channel block with header informa-

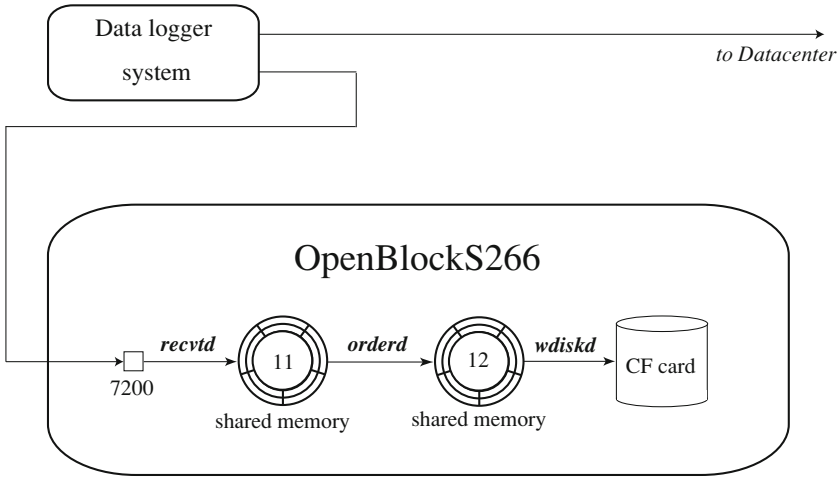

Fig. 1. Schematic diagram of the waveform data flow and processes (in italics) that can run on OpenBlockS266. Details are described in the text.

Table 1. Specifications of the OpenBlockS266.

Maker	Plat'Home Co., Ltd.
CPU	IBM PowerPC 405GPr 255 MHz
Memory	64 MB/128 MB
Flash ROM	8 MB/16 MB
Network I/F	10/100BaseTx × 2
Internal storage	CF (128 MB ∼ 8 GB)/2.5″ HDD
Dimensions	81 (W) × 114.5 (D) × 38 (H) mm
Case material	Aluminum alloy
Weight	Approx. 255 g
Power consumption	≤ 5.0 W (CF)
OS	SSD/Linux
Kernel version	2.6.16/2.4.26/2.4.20
File system	ext2/ext3

tion for each channel; (3) a dynamically variable sampling rate and sample size. The WIN system has valuable features, such as an easy channel split and integration, different sampling rates and sample sizes can co-exist, the number of available channels scales to a large number, and an effective file compression algorithm. The file compression is performed by taking the differential values from the previous sample data for 1 s of each channel data, and the smallest sample size that can express these values is selected among length at 0.5 byte, 1 byte, 2 bytes, 3 bytes, and 4 bytes (first sample is always 4 bytes long). This system is used as a data processing system in Japanese universities and as a data exchange system among universities, including between universities and public organizations, such as JMA. Uehira (2004) and Uehira (2007) improved this system so that it runs on the OpenBlockS266. Figure 1 shows one simple example of processes that run on OpenBlockS266. A data logger sends out A/D converted waveform data with user datagram protocol (UDP) packets to both the data center and to the 7,200th port of OpenBlockS266 every second. OpenBlockS266 receives these UDP packets at the 7,200th port and writes them to the 11th shared memory cyclically [recvtd]. The process names will be denoted in square brackets. The data in the 11th shared memory are then time sorted and written into 12th shared memory cyclically [orderd]. Finally, the ordered data in the 12th shared memory are cyclically written onto the CF card [wdiskd]. The data file with a YYMMDDhh.mm file name format is created once every minute. The process [wdiskd] verifies the remaining storage capacity each time the data are written to storage and ensures that it does not exceed the configuration, for example, 2 MB. When the limit is exceeded, following the removal of the oldest file, the latest file is written. It is something like a ring buffer where data can become overlapped. Table 2 shows station information regarding the channel number, sampling frequency, capacity of the CF card, and the number of waveform data files. Data compression ratios differ depending on the noise level. In turn, the difference in compression ratios affects the duration of data storage. On average, it is possible to store three channels of waveform data at a sampling rate of 100 Hz in a 1 GB CF card for 30–50 days.

3. Development of the New Protocol "WIN Raw Data Recovery Protocol"

A new protocol was developed for sending stored data from the station to the data center. This protocol will be referred to as the "WIN Raw Data Recovery Protocol (WRRP)". "WIN Raw data" means usual waveform data, i.e., WIN format data, but sometimes called WIN format data to distinguish it from the "MON form" data (Urabe, 1994). The protocol is constructed as follows. The data processing should be as simple as possible, and the data traffic should be kept to a minimum. This is necessary because the OpenBlockS266 at the station is less efficient than the servers at the data center and because the communication speed between the station and the data center is usually 64 kbps, which is much slower than a typical LAN, but many times faster than that of the previous telemetry lines connected to the station. When a line experiences trouble for a long period of time, which is followed by the need to transmit a lot of data, all of the channel data must be retransmitted. Therefore, this protocol supports time selection but not channel selection. Goto (2005) also developed a similar and more multifunctional protocol for sending requested

Table 2. List of the observational specifications, the capacities of the CF card, and the numbers of data files that are stored on a single CF card. A data file is made every minute, so that the number of files shows how many minutes can be stored on a single CF card.

Station code	Channel number	Sampling frequency [Hz]	Capacity of CF	Number of data files
ITK	3	100	256 MB	19684 (13.7 days)
MZH	4	100	2 GB	92113 (64.0 days)
TAI	3	100	1 GB	75080 (52.1 days)
NKT	3	100	1 GB	50053 (34.8 days)
NITA	6 (SP), 6 (LP)	100 (SP), 1 (LP)	1 GB	35857 (24.9 days)
USB	3	100	1 GB	73686 (51.1 days)
SMT	6	100	4 GB	150843 (104.8 days)
SBR	3	100	1 GB	67742 (47.0 days)
KJU	3	100	1 GB	43994 (30.3 days)
FUK	3	100	512 MB	38905 (27.0 days)
TKD	3	100	2 GB	96113 (66.7 days)
TMO	3	100	2 GB	124178 (86.2 days)
HIR	6	100	512 MB	15091 (10.4 days)
FKN	6	100	1 GB	22970 (15.9 days)
SWA	4	100	1 GB	38994 (27.0 days)
KRA	2	100	8 GB	742990 (516.0 days)
FRIQ	4	200	2 GB	41581 (28.8 days)
IMA	4	100	1 GB	49272 (34.2 days)
CJA	4	100	1 GB	42923 (29.8 days)
SGS	3	100	4 GB	255777 (117.6 days)
STO	6	100	4 GB	137916 (95.8 days)
KTK	6	100	4 GB	145162 (100.8 days)
IKE	3 (SP), 3 (LP)	100 (SP), 1 (LP)	4 GB	226079 (157.0 days)
YTE	3 (SP), 3 (LP)	100 (SP), 1 (LP)	4 GB	195425 (135.7 days)

data from a server to a client, but WRRP is specialized for transmitting backfill data.

A client-server system was developed that takes into account the reliability of data transmission and reception. For this reason, the transmission control protocol (TCP) was used for the transport layer, while UDP was used for real-time data transmission and reception. The server program runs on OpenBlockS266 at each station. The server program starts from the 'inetd' daemon. The advantages of using 'inetd' are that the machine resources can be used efficiently because the programming is simple and the access configuration allows or denies hosts. In addition, how many hosts can have access to the server at any given time is controlled. These functions can be entrusted to existing programs like 'inetd' and 'tcp_wrapper'.

The client program, which runs on the data center machine, checks every minute for missing data, and if data are missing, then the program requests the server program at each station to send the missing data. Since multiple station data are gathered at the data center, the client program can request multiple servers at any time. The default setting is 100 stations. The requested data are written in a file or sent to the network as UDP packets. When data are sent to the network, the data are handled as delayed packets (Uehira, 2001). They are inserted into existing continuous files and/or event detected files.

Figure 2 shows a schematic of the WRRP. First, the client connects to the server, and the server replies to the handshake message that includes the information about its IP address, port number, and protocol version. Then, the server waits for the data request. Client and server communicate with each other using 128-byte frames, except for the waveform data. Next, the data requests are sent. First, the client sends a packet that begins in 'REQ', followed by the waveform file name, and finally by a 60-byte flag which indicates the required seconds. Then, the server receives this packet and tries to prepare the data. If the server can prepare the data, the server answers with a packet that begins with 'SIZE OK' followed by the data size in 4 bytes, written in big endian word order. If the server cannot prepare the data, the server answers 'SIZE ERR'. If a server answers 'SIZE OK', the client allocates the buffers, and the server transmits the data to the client. This process describes one data request cycle. In addition, there is a 'STAT' command that checks the status of the server. When the server receives this command, it sends the following information to the client: the data storage directory name, the file name of the oldest data, the file name of the most recent data and its size (in bytes), how much data is currently stored, and the configuration of the remaining storage capacity (Fig. 3). Finally, at the end of the session, the client disconnects from the server or sends a 'QUIT' command to the server.

Figure 4 shows two examples of data recovery. Figure 4(a) is an example of a telemetry line that was interrupted due to a thunderstorm, and Fig. 4(b) shows that the carrier communication network was unstable and that data from many stations were missing at the same time. In both cases, the missing data were completely recovered using this system.

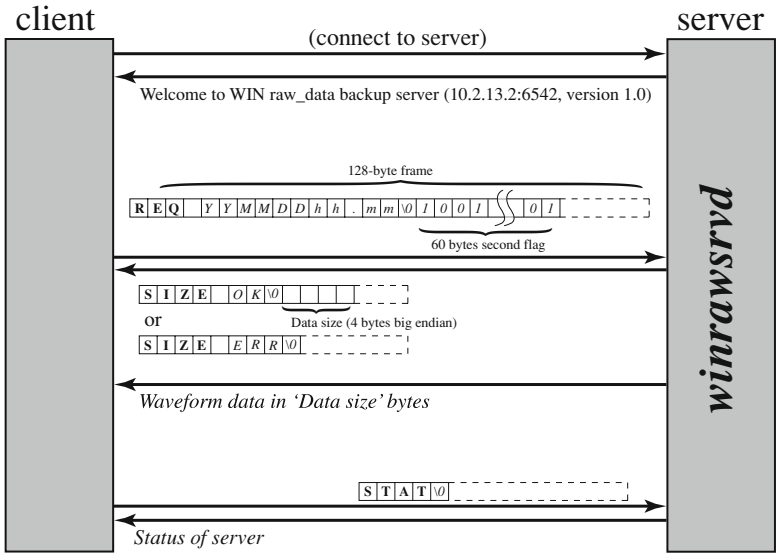

Fig. 2. Outline of the WRRP system. Left shaded rectangle shows the client, that is, the data center side machine, while the right shaded rectangle shows the server, that is, the station side OpenBlocks266. '[winrawsrvd]' represents the name of the WRRP server daemon. The arrows show the flow of the packets. Each square between client and server represents 1 byte. The bold characters in the squares are commands, and the italic characters in the squares are arguments of the command.

```
ITK-OBSS266:6542 /mnt/raw : OLDEST=08032016.26 LATEST=08040308.28 (11040) COUNT=19684 MAX=2 MB
MZH-OBSS266:6542 /mnt/raw : OLDEST=08013009.17 LATEST=08040308.28 (17460) COUNT=92113 MAX=2 MB
TAI-OBSS266:6542 /mnt/raw : OLDEST=08021105.10 LATEST=08040308.28 (15303) COUNT=75080 MAX=2 MB
NKT-OBSS266:6542 /mnt/raw : OLDEST=08022814.17 LATEST=08040308.28 (19860) COUNT=50053 MAX=2 MB
```

Fig. 3. Examples of the reply messages to the 'STAT' command. The left column shows the server name, its port number, the data storage directory name, the file name of the oldest data, the file name of the latest data and its size in bytes, how much data is currently being stored, and the configuration of the remaining storage capacity.

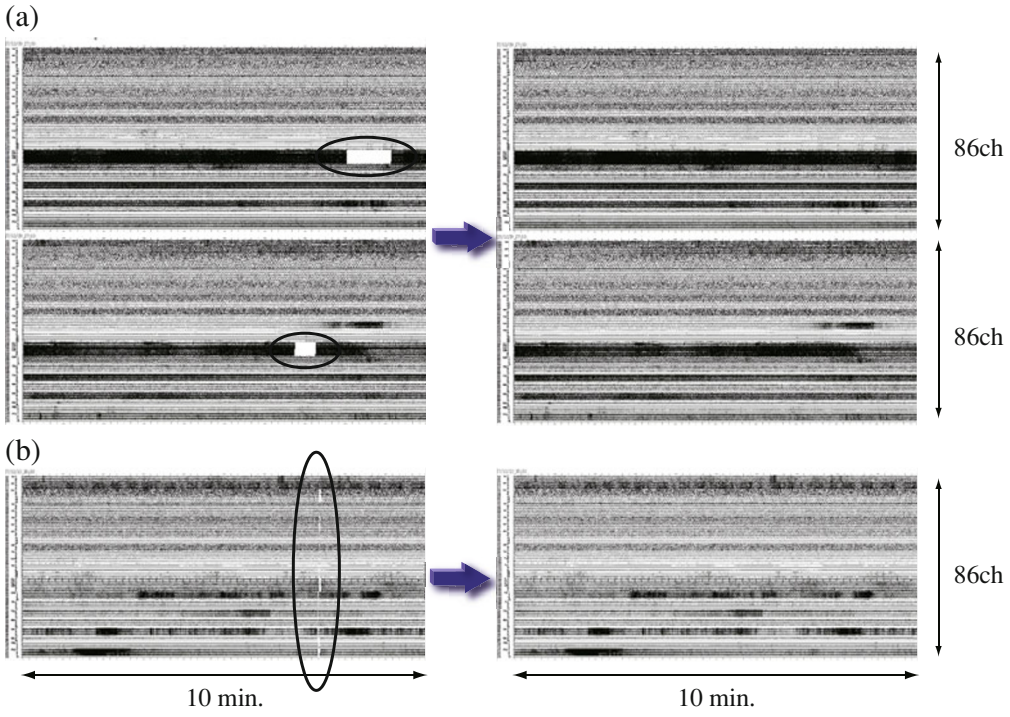

Fig. 4. Examples of data recovery. (Left) Before data recovery. Solid ellipses show the times when data were missing due to (a) telemetry lines that were unstable because of a thunderstorm, and data from one station (six channels) were missing for tens of seconds; and (b) A communication network of the carrier was unstable, and the data from multiple stations were missing at the same time for several seconds. (Right) After recovery of the data using the WRRP. In each case, all data were completely recovered.

4. Conclusions

A distributed backup system and a recovery system were developed in order to recover missing seismic data that results from a system failure at either the data center or the telemetry line. The data can be stored for over 1 year at each station using a microserver with a CF card. The data center can request the necessary data from each station when the missing data are detected. In addition, a new protocol called WRRP was developed for communication between each station and the data center. This protocol allows the resources of the server and the communication line to be used efficiently. WRRP supports only time selection—and not channel selection. In the future, a similar system may be constructed that can delimit data by channel, since the number of channels at a data center is much greater than that at a single station.

The server and client programs were developed on a FreeBSD system, and they were compiled and tested on FreeBSD and SSD/Linux systems. However, these programs are expected to run on many UNIX flavors because they consist of common functions, not special ones.

Acknowledgments. Kazunari Uchida helped set up and install OpenBlockS266s. Comments from Rick Benson and Taku Urabe were helpful in improving the manuscript.

References

Goto, H., Time-Series Data Server Optimized for Multichannel and Real-Time Processing, *IEICE Trans. Inf. Sys.*, **J88-D-1**, 316–325, 2005 (in Japanese).

Hoshiba, M., O. Kamigaichi, M. Saito, S. Tsukada, and H. Hamada, Earthquake Early Warning Starts Nationwide in Japan, *EOS Trans. AGU*, **89**, 73–74, 2008.

Kamigaichi, O., JMA Earthquake Early Warning, *J. Jpn. Assoc. Earthq. Eng.*, **4**, 134–137, 2004.

Uehira, K., Improvement of WIN system, *Abst. Jpn. Earth Planet. Sci. Joint Meeting*, Ss-P002, 2001.

Uehira, K., Let's run the WIN system on the OpenBlockS2666!, http://www.sevo.kyushu-u.ac.jp/uehira/WIN/OpenBlockS266.html, 2004 (in Japanese).

Uehira, K., Data backup system that is worked at seismic station and resending protocol of backup data, *Abst. Fall Meet. Seismol. Soc. Jpn.*, D21-07, 2007 (in Japanese).

Uehira, K. and T. Urabe, Real-Time Data Combination System among Seismic Networks Using IP Network, *Abst. Fall Meet. Seismol. Soc. Jpn.*, P26, 1994 (in Japanese).

Urabe, T., A common Format for Multi-Channel Earthquake Waveform Data, *Abst. Fall Meet. Seismol. Soc. Jpn.*, P24, 1994 (in Japanese).

Urabe, T. and S. Tsukada, A workstation-assisted processing system for waveform data from microearthquake networks, *Abst. Spring Meet. Seismol. Soc. Jpn.*, p70, 1991 (in Japanese).

Urabe, T. and T. Takanami, Real-Time Transmission of Seismic Waveform Data on IP networks, *Abst. Fall Meet. Seismol. Soc. Jpn.*, P25, 1994 (in Japanese).

K. Uehira (e-mail: uehira@sevo.kyushu-u.ac.jp)