Regular paper http://dx.doi.org/10.13067/JKIECS.2013.8.7.1003

소프트웨어방식을 이용한 근해 정박 부이의 기계간의 데이터손실의 최소화

Tan She Young* · 박수홍**

Minimizing Machine-to-Machine Data losses on the Offshore Moored Buoy with Software Approach

Tan She Young* · Soo-Hong Park**

요 약

본 연구에서는 데이터통신을 사용하기 위하여 기계간의 통신을 기초로 한 TCP/IP는 CDMA/GSM을 사용한다. 이 통신방식은 근해정박부이에서는 시스템 서버에 데이터 백업을 위하여 광범위하게 사용된다. 기후나 신호적용범위 때문에 TCP/IP M2M 통신방식은 종종 데이터 전송이 실패하거나 서버에서의 데이터손실을 유발한다. 해양통신이나 기상학적인 해석을 위해서는 데이터 손실을 줄여야 한다. 본 연구에서는 데이터 전달 손실을 최소화하고 데이터복구에서 사용되는 재시도를 위하여 M2M 데이터손실을 최소화하는 소프트웨어방식을 연구하였다. 이 연구의 유용성을 입증하기 위하여 근해에 위치하는 기상용 부이에서 통신을 이용하여 연구의 유용성을 보여주었다.

ABSTRACT

In this paper, TCP/IP based Machine-to-Machine (M2M) communication uses CDMA/GSM network for data communication. This communication method is widely used by offshore moored buoy for data transmission back to the system server. Due to weather and signal coverage, the TCP/IP M2M communication often experiences transmission failure and causing data losses in the server. Data losses are undesired especially for meteorological and oceanographic analysis. This paper discusses a software approach to minimize M2M data losses by handling transmission failure and re-attempt which meant to transmit the data for recovery. This implementation was tested for its performance on a meteorological buoy placed offshore

키워드

Machine-to-Machine, Meteorological Buoy, CDMA/GSM Network, Transmission Failure
기계간통신, 기상용부이, 무선통신네트워크, 근해정박부이

Ⅰ. INTRODUCTION

Meteorological and oceanographic buoy normally placed further away from the shore to gather in-situ environmental data. Transmission of the data back to the server is normally through satellite communication, Very High Frequency (VHF)/ Ultra High Frequency (UHF) transmission

* 동서대학교 대학원 메카트로닉스공학과(sheyong5499@gmail.com)
** 교신저자(corresponding author) : 동서대학교 메카트로닉스공학과(shpark@dongseo.ac.kr)
접수일자 : 2013. 04. 21 심사(수정)일자 : 2013. 06. 20 게재확정일자 : 2013. 07. 23

or Global System for Mobile Communication (GSM)/ Code Division Multiple Access (CDMA) communication [1].

Among most recent methods, CDMA communication with TCP/IP stack or in other words TCP/IP based Machine-to-machine (M2M) communication is one of the preferred methods used by coastal buoy due to its feature that allowing direct data transmission from CDMA modem on the buoy to data server. CTIA - The Wireless Association define M2M as “application or mobile units that use wireless communicate with other machines” [2]. The usage of M2M is expected to grow rapidly in near future and cover more and more applications where the cellular network is available [3]. However, this communication method is highly dependent on the signal reception. As the buoys are placed offshore and have significantly weaker signal compare to civilized land, sea surface condition and weather became factors to affect CDMA reception [3]. As offshore moored buoys are normally working 24 hours a day 7 days a week unattended for long period of time, a robust system is required to ensure every data gathered can be transmitted to the receiving personnel.

As the M2M service on ocean buoy was just started years ago and the firmware to interface with the modem are still not good enough after long term operation. The current firmware is still imperfect when dealing with some special cases such as, TCP/IP port opening failed, unable to close port, slow modem respond, interrupted by incoming text message and some other uncertainties. These interfacing issues directly affect the interval data reporting and transmission.

However, frequent changes in modem are not a good way to encounter the problem and it takes another long period to test its reliability. On board memory card logging is also an option to retrieve lost gathered data but this is not an option for application such as weather forecast which requires real-time data.

Therefore, system firmware solution is proposed to reduce the data loss which caused by modem’s undesirable status. The main idea in this solution is to minimize the risk which able to caused transmission failure or pre-transmission failure. In another word, the algorithm halts transmission procedures if it ran into any uncertain event. The output message will be saved and wait for re-attempt. This solution is tested with Telit BSM-856 CDMA modem which operating in an offshore meteorological buoy for 2 months.

The main idea in this solution is to minimize the risk which able to caused transmission failure or pre-transmission failure. In another word, the algorithm halts transmission procedures if it ran into any uncertain event. The output message will be saved and wait for re-attempt. This solution is tested with Telit BSM-856 CDMA modem which operating in an offshore meteorological buoy for 2 months. Benefits of this approach include no hardware changes in the whole buoy system and the firmware of the system can be updated during maintenance work on the buoy. The buoys need not to be retrieved from the sea for the update.[6,7,8]

Ⅱ. PLATFORM OVERVIEW

This approach is embedded in a meteorological buoy communication board firmware. Meteorological buoy communication board is responsible to handle all communication of the buoy with the outside world. On the test bed buoy, the communication board receives data message from the combiner board continuously and at the same time, accumulate the data for averaging process. When it reaches the transmission interval, the output string will be sent through ST2500 Orbcomm Communicator through satellite and Telit BSM-856 CDMA Modem through M2M service over CDMA network. Besides, the system also provides data logging on SD memory card as backup and reference. Figure 1 below shows the integration between several important devices and modules in the testing platform.

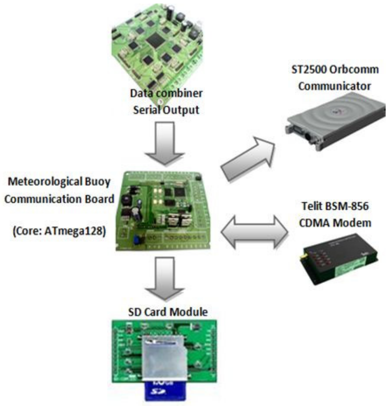

Fig. 1 Platform overview

The output messages transmit through satellite and M2M are in the same format with date and time, GPS location, and all the gathered data. Date and time are important especially for re-attempted message. This enables the server program to sort the data with respect to the date and time or data sequence. The sample format of the M2M output message is shown Figure 2 below. CDMA M2M modem handles the message in ASCII form which is more convenient for error detection. The test bed buoy hardware structure is a standard 3 meters discus buoy. The appearance of the 3 meters discus buoy is shown in Figure 3.

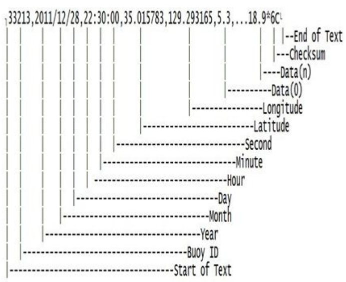

Fig. 2 Message format

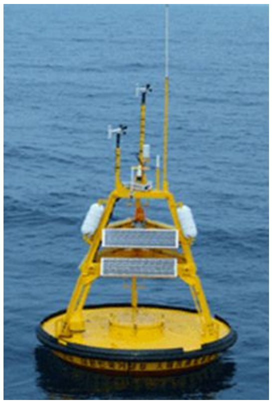
Fig. 3 3 meters discus buoy

III. METHODS

In order to minimize the data losses, method used in this project considered the following key points. Firstly, avoid all undesired cases. In other words, when the data transmission facing difficulties, the firmware will abandon the transmission process and end it up nicely for future attempt. Another key point is assuming no acknowledgement return within an allowable period of time is a failed transmission. Besides, by introducing backup and re-attempt, the transmission which affected by undesired circumstances can be halted and retry again after a short while. This might cause the system losing real-time data but it able to keep the data set losses to minimum. With date and time in the message, the receiver server computer program able to sort the data back to the original sequence for long time evaluation and analysis.

A. Transmitting

Procedures need to be followed strictly for the modem to transmit an output message out to the receiver. Due to some practical and external factors, modem might not respond to the system as expected. Failure may happen at any point between start and end of data transmission. Modem's acknowledgement is used to judge whether the procedure is passed successfully. The standard procedures of TCP/IP M2M data transmission using BSM-856 CDMA modem is shown in the Figure 4.

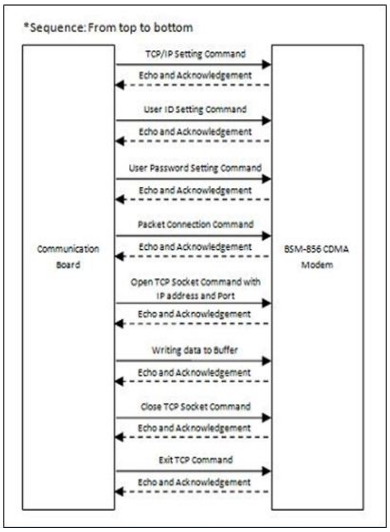

Fig. 4 Standard procedures of TCP/IP data transmission

Due to some undesirable factors such as weak signal, server problem or busy status, the modem might return error which hardly occurred in good signal coverage environment or sometimes very slow in acknowledgement respond. Errors are commonly occurring at packet connection command and open TCP socket command. BSM-856 Programmer's Guide did include the possible error acknowledgements which might happen during undesired situation. For example, NO CARRIER means unable to activate the service. NO DIALTONE means no dial tone detected within time-out period and ERROR if the command is not recognized or could not be executed [4]. But it did not have detailed note on the possible acknowledgement for each command respectively. With lacking information of the acknowledgement and response during weak signal operation, set of experiments were done. However, without a proper experiment and testing facilities to reduce the modem signal reception, the only possible experiment can be done is to modify the modem antenna to cause the modem move from good signal to bad signal status.

Experiment was done with good antenna, bad antenna and no antenna and the procedures for TCP/IP M2M transmission was carried out. By using a serial terminal utility in the computer with an USB to Serial Converter connected with the CDMA modem, acknowledgement returned after each command was saved as the experiment result. Sudden signal lost during transmission procedures also carried out to test the modem acknowledgement and modem's response to the host. The summary of the result shows all the setting command will not affected by the signal level. But all the other commands after the setting stage are all affected with the signal reception. The findings are summarized as the following.

1) Packet Connection Command
- It takes longer time to get connected if the signal level is low.
- Acknowledge CONNECT for successful connection but usually slow.
- Acknowledge NO DIALTONE if no antenna is connected to the modem (fast acknowledgement).
- Acknowledge NO CARRIER twice with 2 sets of carriage return (0x0D) and new line (0x0A) in between them during weak signal reception.

2) Open TCP Socket Command with IP address and Port
- For successful connection, it replies OK and $TCPOPEN. There are 2 set of
- It replies ERROR if previous step had failed or unable to perform the operation.

3) Writing Data to Buffer

- It replies OK and $TCPSENDDONE with 2 sets of carriage return (0x0D) and new line (0x0A) between them if the operation is successfully performed.
- ERROR if previous step failed, connection disconnected and etc.

4) Close TCP Socket
- It replies OK and $TCPCLOSED with 2 sets of carriage return (0x0D) and new line (0x0A) between them if the operation is successfully performed.
- ERROR if previous step failed, connection disconnected and etc.

5) Exit TCP Command
- The positive response will be OK and NO CARRIER with 2 sets of carriage return (0x0D) and new line (0x0A) between them.
- Negative response will be simple ERROR.

Method used in this paper considered both slow response (more than 2 seconds) and error are failure in transmission. Depending on the stage of failure, different levels of rollback process need to be done to configure the modem back to the normal operation mode for next attempt. Figure 5 shows the flowchart for data transmission and failure reporting

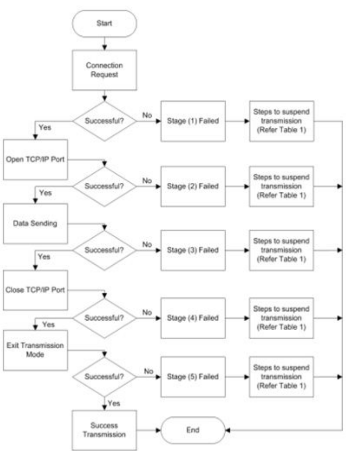

Fig. 5 Transmission and failure report

B. Failure Stages and Solutions

This solution main concern is to avoid interfacing failure which could ends up with transmission failure. There were five different failure points in the flowchart shown in Figure 5, each of them indicates the stage where failure occurred. For failure in Stage (1), Stage (2) and Stage (3), command to exit transmission mode need to be sent to end the session. The output data string will be saved into microcontroller (MCU) memory for re-attempt and a re-attempt counter increment by one.

The output string is saved in last in, first out (LIFO) manner in MCU memory where the latest data has higher priority than older data.

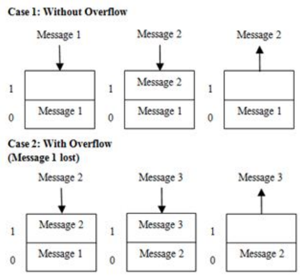

Fig. 6 LIFO stack with size 2

There are few strong reasons by implementing LIFO in this application. Buoy is an offshore device which transmits its status and collected data back to the onshore base routinely. Therefore, if the onshore base does not receive any data for few intervals, the latest buoy status will become the highest priority data for the service provider to know what is happened to the buoy. With the latest status being transmitted first, the buoy service provider able to take action if the status more likely to be collision or sinking. By using this manner, the system able to reports most up-to-date status of the buoy then only re-attempts to transmit the others. However, if overflow occurs in the stack, the oldest data will be cleared and to allow the latest data to be store into the memory. Figure 6 illustrates the stack with size of 2.

On the other hand, failure at Stage (4) and Stage (5) showed the TCP/IP port unable to close and exit the transmission mode. For failure in Stage (4) and Stage (5), the only operations required are re-attempting to close TCP/IP port and exit transmission mode. Resending the output message is not required because the failures are occur right after the sending step.

C. Re-attempt

If the stack is not empty, re-attempt will be carried out to transmit the message(s) stored. The firmware will check the connectivity routinely for the re-attempt. In last implementation, the re-attempt connectivity check was carried out every 1 minute (configurable) while the buoy reporting interval can be from 10 minutes to 1 hour. By choosing the re-attempt check interval less than normal reporting interval, data traffic congestion can be reduced during normal reporting interval. Once the connectivity check shows good signal, the algorithm will send the saved message(s) one by one. A fully filled small size stack can be totally cleared before reaching the next reporting time.

Figure 7 below illustrates the idea of re-attempt where the data interval is set to 10 minutes and re-attempt interval is set to 2 minutes with size of the stack is 2 to hold the data.

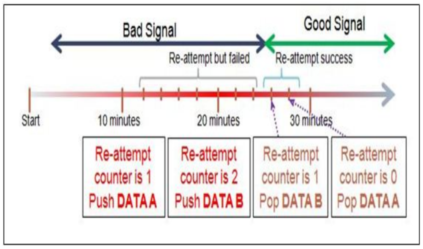

Fig. 7 Illustration of re-attempt process

The illustration started with bad signal period and when it reached the first 10 minutes interval, DATA A was not able to transmit. This causing the DATA A being pushed into the memory stack and the counter increment by 1. The counter was not zero anymore therefore the firmware check for the signal every 2 minutes in order to re-transmit the data in the memory but failed. When the time reached the second interval, again the signal was bad and DATA B was not able to be sent. It was pushed into the memory stack and the stack was then carrying 2 data (DATA A and DATA B) with the counter incremented again to 2. Again the counter was not zero, every 2 minutes the command was sent to check the possibility to re-transmit. Once the signal turned good and the re-transmit interval was reached, DATA B was popped first. The counter was then decrement to 1. The same thing happened at the next re-transmit interval which DATA A was transmitted and lastly all data were successfully sent.

Ⅳ. RESUTLS

By fixing the buoy reporting interval at 30 minutes, data collection had been carried out from 1st May 2012 until 30th June 2012.

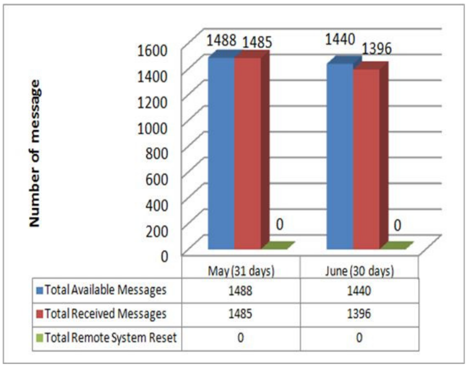

Fig. 8 Result

Total number of messages in month and total received messages by the server are presented in the chart shows in Figure 8 below. Besides, in both May and June 2012, system reset was not performed by the administrator.

In May 2012, only 3 data losses recorded (approximate 0.2%). On the other hand, 44 data losses recorded (approximate 3.06%) in June 2012. Remote system reset is normally being performed by the administrator if the buoy did not transmit any message to the server for more than 2 hours. This method is to clear any hardware and software issue which causing the system not actively communicate with the server. However, during 2 months of operation, the remote reset was not performed even once.

Due to the left over memory in ATmega128 and external EEPROM do not support high capacity stack, the implementation uses only stack with size 1. The stack can only hold one message for re-attempt due to the long formatted message string.

Therefore, if the system unable to transmit the message successfully for two times in a row, the first message will be lost. For better result, hardware modifications or memory management need to be done to accommodate more messages for re-attempt. Another possible method is to store the pending data into a structure rather than ASCII string which costs 1 byte for each character or digit.

However, for the modem TCP/IP transmission failure, the solutions shows good result in preventing mishandling in the CDMA modem which might need remote system reset to solve the problem.

Memory limitation is one of the main concerns in this implementation. Low memory issue need to be solved either by code optimization or hardware upgrade for better performance. This project is currently powered by Atmel ATmega128 microcontroller with 32Kbytes external memory. This microcontroller able to support up to 64Kbytes of external memory which means the memory expansion is possible without changing the microcontroller but is only limited to 64Kbytes[5]. Another possible way to upgrade the performance is by upgrading the 8-bit microcontroller to either 16-bit or 32-bit microcontroller to allow more memory space allocated for backup or retransmission. There is also possible to convert the message from ASCII array to hexadecimal type for data storage and convert back to ASCII code for transmission. This requires another software function to convert the data but certain amount of memory can be saved.

V. CONCLUSION

As a result, this method is proven to be useful in dealing with TCP/IP M2M data communication in minimizing data losses. However, the short term test might not sufficient for a very persuasive outcome because the environment and weather are the factors which might disturb the signal level. We knew that South Korea weather condition is greatly depend on the four seasons of the year. Therefore, a long term testing and analysis need to be done to further test the robustness of this solution.

감사의 글

"This Research was supported by Dongseo Universty Annual Special Research Grants 2012"

REFERENCES

[1] Wonbu Lee, Soo-hong Park, "A study on the Automatic ocean wave observation buoy system", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 6, No. 2, pp. 268-273, 2011.

[2] CTIA Consumer Info- Wireless Glossary of

Terms[Online] : http:// www.ctia.org/consumer_info/ service/index.cfm/AID/10408
[3] The Effects of Humidity on Cell Phone Signals http://upcomingcellphones201.com/the-effects-of-humidity-on-cell-phone-signals.html
[4] BCM(BSM)-860S/1860 Programmer's Guide Version 1.0.6
[5] Atmel ATmega128 Datasheet [Online]. Available : www.atmel.com/images/doc2467.doc
[6] Woon-Taek Hwang, "A Design and Implementation of Maritime Data Communication System ", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 6, No. 6, pp. 815-822, 2011.
[7] Sang-Young Kim, Soo-Hong Park, Jee-Yeun Park, "Development of the Winch Control System for Floating Island and Statistics Analysis for data", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 7, No. 2, pp. 309-315, 2012.
[8] Soo-hong Park," Electronics System Design of a Generic Meteorological Buoy", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 5, No. 1, pp. 51-57, 2010.

저자 소개

탄시영(Tan She Young)

2011년 7월 말레지아 MultiMedia University 졸업(공학사)
2013년 8월 동서대학교 대학원 메카트로닉스공학과 졸업예정
※ 관심분야 : 마이트로프로세서응용

박수홍(Soo-Hong Park)

1986년 2월 부산대학교 정밀기계공학과 졸업 (공학사)
1989년 2월 부산대학교 대학원 기계공학과 졸업(공학석사)
1993년 2월 부산대학교 기계공학과 졸업(공학박사)
동서대학교 메카트로닉스공학과 교수
※ 관심분야 : 제어 자동화, 로봇공학