Development of an Ultra-small Sensor Information Remote Monitoring System with an Embedded VPN and Linux Microcomputer Operation

著者	SHIBATA Kouji, HANADA Kazuma
著者別名	柴田 幸司, 花田 一磨
journal or publication title	The Bulletin of Hachinohe Institute of Technology
volume	34
page range	93-99
year	2015-03-31
URL	http://id.nii.ac.jp/1078/00003534/

Development of an Ultra-small Sensor Information Remote Monitoring System with an Embedded VPN and Linux Microcomputer Operation

Kouji SHIBATA[†] and Kazuma HANADA[†]

ABSTRACT

In this study, an encrypted closed line was created over the Internet by building an embedded VPN with a Linux microcomputer and a data communication terminal for combination with mobile telephone networks. It was also found that information on temperature/humidity and camera images from sensor-remote locations could be acquired through a web browser using a smart device such as a tablet computer by connecting the sensor information reception/transmission equipment to the VPN. The system can be set up anywhere because the Linux microcomputer is connected to a mobile network. The results showed that this technology can be used to transmit information on sensors and switch circuit control over a public line with no risk of sniffing. The technique can be applied for a range of purposes, including electricity consumption monitoring and remote crop management. The authors also believe the system has strong potential in education relating to information, communication and computer network technology on campus.

Keywords: *Internet, VPN, mobile telephone network, cellular network, NAT transversal, remote monitoring system, Raspberry Pi, Linux, microcomputer*

1. Introduction

Telemeters and other systems adopted to transmit sensor information from remote locations previously required a costly dedicated line connection, but can now be set up more economically thanks to the growth of the Internet.

The recent rapid proliferation of data communication and cellular networks along with the associated increasing trend of data transmission speeds has also enabled the transmission of large volumes of data online from mobile units. As a result, a wealth of information (such as camera data from sensor-connected devices in remote mountainous areas) from mobile objects can be acquired anytime, anywhere with only a browser and an Internet connection.

平成27年1月8日受付

† Department of Electrical and Electronic Systems, Lecturer

In previous work, a stand-alone VPN (virtual private network) designed to transmit sensor information with a VPN access router and a cellular network or WiMAX network combination was created[1, 2]. The system can also be operated via a radio-based Internet connection in mountainous areas and other locations where fixed phone lines cannot be installed.

The authors further previously confirmed the feasibility of receiving data from information acquisition systems in sensor-remote locations via a web browser using a smart device such as a tablet computer by connecting the equipment used to transmit and receive sensor information for the Ethernet to this VPN[1, 2]. However, conventional remote monitoring systems require a costly dedicated VPN router, which accounts for a substantial portion of system costs. Conventional VPNs must also be assigned a fixed global IP address by the Internet service provider to enable the connection of equipment in remote locations via cellular networks. These limitations represent a significant economic burden in VPN operation.

In this study, an extremely compact and inexpensive remote monitoring system without the need for sensor information acquisition equipment was developed based on a combination of a Linux microcomputer and various USB-connected sensors. The proposed system enables the acquisition of camera data and information on temperature/humidity at low operational cost with the assignment of a dynamic private IP address by the Internet service provider thanks to a firewall-traversing VPN program and a NAT (network address translation) function simply installed on the microcomputer. A USB-based mobile broadband modem for LTE networks via the cellular network of Japanese provider NTT Docomo was also connected and controlled directly with the Linux microcomputer, thereby eliminating the need for a mobile wireless router. This technological combination represents an extremely simple and inexpensive system that enables the acquisition of information on temperature/humidity and data from sensor-remote locations via a web browser using a note PC and a tablet computer. This paper outlines the realization of the proposed system.

2. System summary

The system consists of a Linux microcomputer, a USB modem to enable cellular network connection and USB-connected sensors to enable the collection of data on temperature/humidity, camera images and other information (**Figs. 1, 2**). The Raspberry Pi model (RS Components) originally developed for educational purposes was chosen as the ARM-based Linux microcomputer due to its cost-effectiveness and its known versatility in terms of device driver development. An L-02C modem (LG Electronics) was chosen to allow interfacing with the microcomputer using a USB Internet connection via the cellular network of Japanese provider NTT Docomo, and a standard USB web camera was used to import camera images to the microcomputer. A USBRH-FG USB-connected temperature and humidity sensor combination module (Strawberry Linux) was also chosen for its track record in measurement using a Linux PC. In the proposed system, connection between the Linux microcomputer with a data communication unit for remote information acquisition and the client terminal is secured using VPN software built into the Raspberry Pi. As a result, the system does not require a separate VPN router. In addition, the Linux microcomputer maintains a constant web server-to-terminal connection via the dedicated VPN program, easily negotiating NAT and firewall protection with a private IP address assigned dynamically to the computer without static from the Internet service provider. This combination of technology significantly reduces the cost of the Internet connection required for secure transmission of data from a remote server to a terminal without the risk of third-party intrusion inside the VPN, thereby supporting enhanced confidentiality. It was found that information from remote sensors could be acquired using an intelligent device such as a smartphone via cellular networks and the Internet anywhere in the world by connecting the set-up used to transmit and receive sensor information in the proposed system to an arbitrary location.

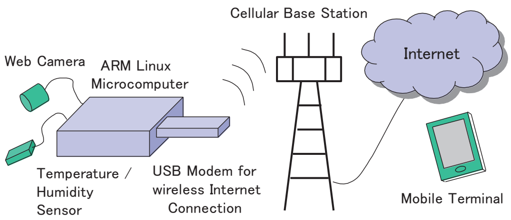


Figure 1 System configuration

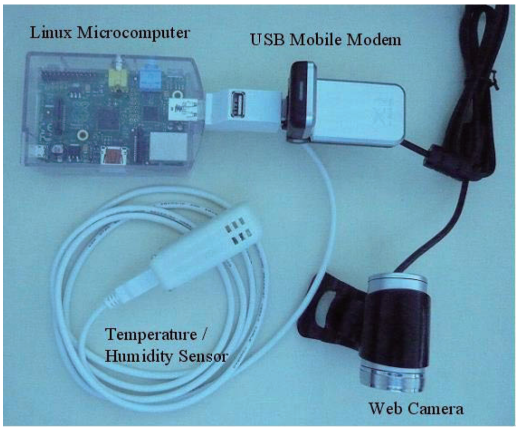


Figure 2 System configuration

3. VPN connection to the Internet from a Linux microcomputer via Wi-Fi

Firstly, connection to a number of sensor information remote monitoring systems using the Raspberry Pi linked to a wireless LAN router via radio from another place over the VPN was confirmed with the basic set-up shown in **Figure 3**. Next, sensor information was obtained from a remote location by connecting environmental sensors and a web camera to the microcomputer with the proposed system. The procedure implemented is as follows:

1. Installation of Linux on a Raspberry Pi
2. Setting of LAN conditions (IP address, gateway, DNS)
3. Setting of a wireless LAN adaptor connected to the Raspberry Pi
4. Installation and setting of the VPN program
5. Setting of the firewall (connection admission control for ports 22, 80)
6. Installation of the web (HTTP) server and PHP software
7. Installation of Motion software for the web camera
8. Installation of the program for collection of sensor information on the environment
9. Rewriting of the file stored as HTML information (index.php)

For the system, a Buffalo WLI-UC-GNM unit was chosen as a wireless LAN adapter for physical connection to the Raspberry Pi, and a Buffalo WZR-450HP wireless LAN router was chosen for radio connection. A dynamic private IP address was also assigned to the Linux microcomputer from the wireless LAN router using DHCP to enhance scalability for connection to another variety of access lines under these set-up conditions.

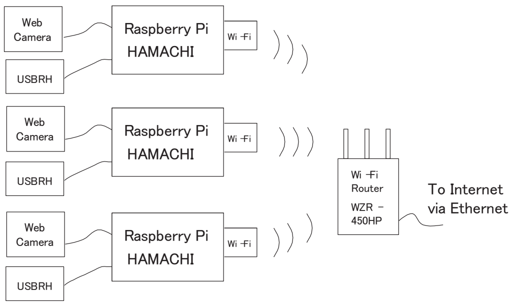


Figure 3 Configuration for VPN connection via Wi-Fi

VPN method selection is very important in reducing costs for the operation of such remote monitoring systems. The proposed system's VPN connection is achieved using an application called Hamachi (LogMeIn), which enables the operation of a variety of servers using a private IP address and allows easy VPN construction across NAT and firewall barriers. In addition, a dedicated program to link both a Linux microcomputer as the server and a PC as the terminal is needed to operate the VPN. **Figure 4** shows the state of connection between Linux microcomputers and a Windows XP client PC after the completion of 1 – 9 above. The results show a number of Raspberry Pi connections to the wireless LAN router over the radio on the VPN constructed using Hamachi.

Figure 4 Connection of individual computers using Hamachi

Next, Apache2 was installed for the website to allow the microcomputer to work as an HTTP server, and PHP5 was installed on the system to allow the indication of temperature and humidity information through a web browser. The motion-capture program allowing web camera usage was adopted from Motion software, whose installation allows long-distance monitoring based on USB web camera images through a web browser on a microcomputer. As a standard feature, it also allows image data to be stored on the microcomputer's SD card when the web camera's internalized landscape changes. As a result, image data from a remote device can be obtained easily by accessing the sensor system from the terminal over the VPN using FTP.

The program adopted to receive temperature and humidity information from the sensor module (USBRH-FG) using the Raspberry Pi was generated via the following procedure. Firstly, the source file of a program edited in C language to work for a CPU with different architecture was adapted to work on the Raspberry Pi. Next, an executable file was generated using the *make* command for the Raspberry Pi's Debian OS. Finally, a function to display temperature and humidity data from the sensor was mounted using this command for the Linux OS. The index.php file code was also edited as shown in **Fig. 5**. The /var/www directory was then located to check the sensor information using a mobile terminal device from a server via a web browser.

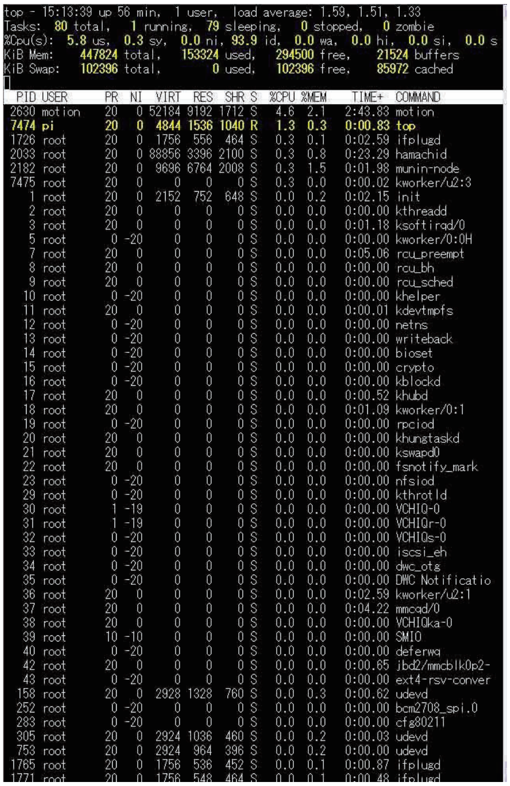
```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta http-equiv="Content-type" content="text/html;charset=UTF-8">
<title> Temperature and Humidity in the Laboratory </title>
</head>
<body>
<font size="7">
Temperature Humidity
<br>
<?php
echo exec('/usr/local/bin/usbrh');
?>
<br>
<a href="http://25.42.***.***:8081"><img class="motion"
src="http://25.42.***.***:8081"/></a>
<br>
</font>
```

Figure 5　Content of the edited index.php file

Figure 6 shows the display of remote information in a Firefox web browser for the web server located on-site on a Windows XP PC via VPN after the completion of 1 – 9 above. Camera images and temperature/humidity information are shown on the screen. Note that a private IP address is assigned dynamically to the microcomputer using DHCP by an access router connected to the Internet. The system thus stays linked to the VPN by switching back and forth among fixed-line, Wi-Fi and cellular phone network connections when any of these paths is interrupted.

Figure 6　Sensor information obtained via the Internet and a VPN

Figure 7 shows the screen detailing OS operational status using the *top* command based on the installation of various programs. The statuses of Hamachi, Apache2, Motion, USBRH and other programs are displayed.



```
top - 15:13:39 up 56 min,  1 user,  load average: 1.59, 1.51, 1.33
Tasks:  80 total,   1 running,  79 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.8 us,  0.3 sy,  0.0 ni, 93.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 s
KiB Mem:    447824 total,   153324 used,   294500 free,    21524 buffers
KiB Swap:   102396 total,        0 used,   102396 free,    85972 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 2630 motion    20   0 52184 9192 1712 S   4.6  2.1   2:43.83 motion
 7474 pi        20   0  4844 1536 1040 R   1.3  0.3   0:00.83 top
 1726 root      20   0  1756  556  464 S   0.3  0.1   0:02.59 ifplugd
 2033 root      20   0 88856 3396 2100 S   0.3  0.8   0:23.29 hamachid
 2182 root      20   0  9696 6764 2008 S   0.3  1.5   0:01.98 munin-node
 7475 root      20   0     0    0    0 S   0.3  0.0   0:00.02 kworker/u2:3
    1 root      20   0  2152  752  648 S   0.0  0.2   0:02.15 init
    2 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S   0.0  0.0   0:01.18 ksoftirqd/0
    5 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0     0    0    0 S   0.0  0.0   0:05.06 rcu_preempt
    8 root      20   0     0    0    0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      20   0     0    0    0 S   0.0  0.0   0:00.00 rcu_sched
   10 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 khelper
   11 root      20   0     0    0    0 S   0.0  0.0   0:00.01 kdevtmpfs
   12 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 netns
   13 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 writeback
   14 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 bioset
   15 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 crypto
   16 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 kblockd
   17 root      20   0     0    0    0 S   0.0  0.0   0:00.52 khubd
   18 root      20   0     0    0    0 S   0.0  0.0   0:01.09 kworker/0:1
   19 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 rpciod
   20 root      20   0     0    0    0 S   0.0  0.0   0:00.00 khungtaskd
   21 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kswapd0
   22 root      20   0     0    0    0 S   0.0  0.0   0:00.00 fsnotify_mark
   23 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 nfsiod
   29 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 kthrotld
   30 root       1 -19     0    0    0 S   0.0  0.0   0:00.00 VCHIQ-0
   31 root       1 -19     0    0    0 S   0.0  0.0   0:00.00 VCHIQr-0
   32 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 VCHIQs-0
   33 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 iscsi_eh
   34 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 dwc_otg
   35 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 DWC Notificatio
   36 root      20   0     0    0    0 S   0.0  0.0   0:02.59 kworker/u2:1
   37 root      20   0     0    0    0 S   0.0  0.0   0:04.22 mmcqd/0
   38 root      20   0     0    0    0 S   0.0  0.0   0:00.00 VCHIQka-0
   39 root      10 -10     0    0    0 S   0.0  0.0   0:00.00 SMIO
   40 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 deferwq
   42 root      20   0     0    0    0 S   0.0  0.0   0:00.65 jbd2/mmcblk0p2-
   43 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 ext4-rsv-conver
  158 root      20   0  2928 1328  760 S   0.0  0.3   0:00.62 udevd
  252 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 bcm2708_spi.0
  283 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 cfg80211
  305 root      20   0  2924 1036  460 S   0.0  0.2   0:00.03 udevd
  753 root      20   0  2924  964  396 S   0.0  0.2   0:00.00 udevd
 1765 root      20   0  1756  536  452 S   0.0  0.1   0:00.87 ifplugd
 1771 root      20   0  1756  548  464 S   0.0  0.1   0:00.48 ifplugd
```

Figure 7 Details of Linux microcomputer operational status

4. Connection to a VPN via a cellular network

Connection to the Hamachi VPN via a cellular network using the Raspberry Pi was also confirmed by connecting a USB modem to a microcomputer directly. For this purpose, software called WvDial was installed on the microcomputer to enable Internet connection via the cellular network of Japanese operator NTT Docomo using a USB dongle modem. This program allows connection to the Internet using PPP (Point-to-Point Protocol) by sending AT commands to a modem. In this case, a configuration file (wvdial.conf) of the Raspbian OS as shown in **Fig. 8** was edited and located as /etc/wvdial.conf for connection to Japanese mobile virtual network operator (MVNO) Excite LTE through modem control. **Figure 9** shows the USB modem connection status with a link to the Docomo network via the process of WvDial. The LED turns blue to indicate that the USB modem is connected to the LTE network. **Figure 10** also shows the status of connection from the Raspberry Pi logged-in via a SSH to a Google DNS whose IP address is set to 8.8.8.8 using the *traceroute* command. It can be seen that the connection path to the Google DNS is via the IIJ Internet service provider. In this case, Excite LTE as the MVNO assigns a private IP address to a terminal. The results show that inexpensive and stable VPN connection across NAT and firewall environments was achieved using only the Linux microcomputer.

```
[Dialer Defaults]
Init1 = ATH
Init2 = AT&F
Init3 = ATZ
Init4 = AT&F
Init5 = ATZ
Init6 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init7 = AT+CGDCONT=1,"IP","vmobile.jp"
Dial Attempts = 3
Stupid Mode = 1
Modem Type = Analog Modem
Dial Command = ATD
Stupid Mode = on
Baud = 460800
New PPPD = yes
APN = vmobile.jp
Modem = /dev/ttyUSB2
ISDN = 0
Phone = *99***2#
Password = excite
Username = bb@excite.co.jp
Carrier Check = off
Check Def Route = on
Abort on No Dialtone = off
```

Figure 8 /etc/wvdial.conf file setting

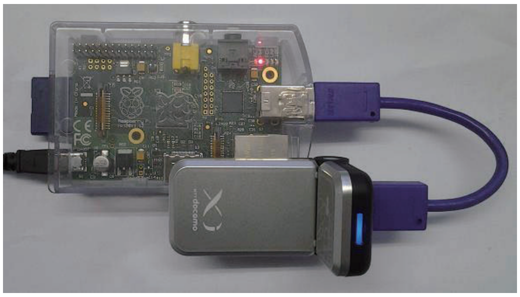


Figure 9 Confirmation of connection via the NTT Docomo network

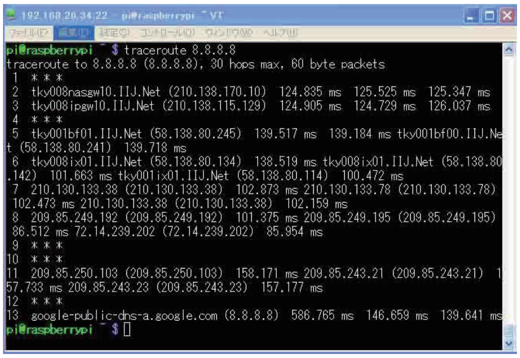


Figure 10 Confirmation of connection to the Google DNS via a cellular network

5. Connection using a mobile terminal device

Based on this setup, android 4.0.4 Samsung Galaxy Tab 10.1 (SC-01D) tablet computer with an Android OS was assigned to the Hamachi network. The status of mobile terminal connection to the VPN using this device and PPTP (Point-to-Point Tunneling Protocol) is shown in **Fig. 11**. These outcomes verify that temperature/humidity information and camera images can also be retrieved from a remote location using a mobile terminal with the proposed system.

Figure 11 Display on an Android OS tablet computer

Munin software was also installed on the Raspberry Pi to monitor the server's operation status, and a script was written to import environment information. **Figure 12** shows the results of monitoring performed using Hamachi to determine temporal variations in temperature and humidity. It can be seen that longitudinal sensor information was obtained continuously over a period of a week, thereby indicating the stability of the proposed system.

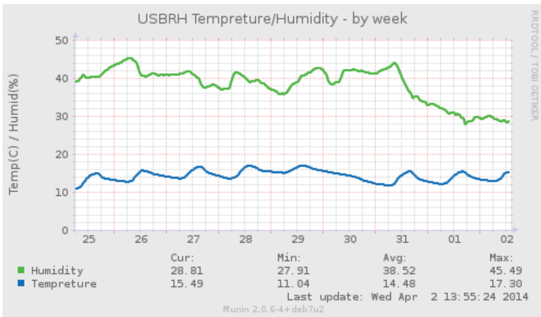


Figure 12 Temporal variations in temperature and humidity

6. Conclusion

In this study, an encrypted closed line was created over the Internet by building an embedded VPN with a Linux microcomputer and a data communication terminal for mobile telephone network combination. The result was an extremely compact and popular system for the remote monitoring of sensor information on temperature/humidity, web camera data and other information based on a combination of the microcomputer and various USB-connected sensors. It was also confirmed that such data could be acquired in sensor-remote locations through a web browser using a smart device such as a tablet computer by connecting the equipment to this VPN. The results further verified that this technology can be used to transmit information on sensors over a public line with no risk of sniffing via a cellular network and a VPN.

The technique can be applied for a range of purposes, including the monitoring of electricity consumption, remote management of crops, off-site evaluation of growth in coastal aquaculture, and disaster prevention.

The authors also believe the system has strong potential in education relating to information, communication and computer network technology on campus.

Plans for future study include outdoor remote monitoring based on a combination of a stand-alone electricity supply (such as solar cells) and an electricity storage system, the connection of atmospheric sensors and radar for motion detection, and on-off control for various instruments.

References

1) Kouji Shibata et al., "A Remote Monitoring and Control System for Sensor Information Based on a Stand-Alone VPN over a Mobile Telephone Network," The bulletin of the Hachinohe Institute of Technology, 31, pp. 115-120, 2012-3.

2) Kouji Shibata et al., "A Remote Monitoring System for High-Speed Sensor Information Based on a Stand-Alone VPN over a WiMAX Network," The bulletin of the Hachinohe Institute of Technology, 32, pp. 129-134, 2013-3.

3) Kouji Shibata et al., "A Remote Monitoring System for Sensor Information Based on a Stand-Alone VPN Using a Linux Micro Computer," The bulletin of the Hachinohe Institute of Technology, 33, pp. 115-120, 2014-3. SSN: 0287-1866; NCID: AN00205113.

4) Kenji Watanabe, Makoto Otani, "The TANADA-Online Project" IEICE Technical Report Vol. 108 (74), IA2008-9, pp. 43-48, 2008-5.

5) Raspberry Pi website: http://www.raspberrypi.org/.