ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 9, September 2015

Real Time Communication with Client/Server Architecture Using Secure Shell Protocol

Leeja Joseph[1]
[1] **M.Tech Student, IIET, Kothamangalam,**

Doniya Jose[2]
[2] **Asst. Prof., IIET, Kothamangalam**

Abstract— Nowadays, we need to interface to the distant devices is one of the significant task for supervising the distributed systems. The vogue and opportunity of the Internet enables the uncomplicated designing of distributed control systems. We often presume that the communication over the internet is just as firm as traditional forms of personal communication. We expect that what we are saying will reach the recipient without being altered. Unfortunately there are many security problems in distributed control systems and also there are many approaches to transmission security problem. This paper focuses on low cost solution - TCP correspondence tunneling using familiar SSH protocol. In objection to private enterprise solutions, the use of SSH client approach requires almost no user privileges on client system. Hence we prove that the SSH protocol is more efficient than tcp protocol. The Network Control System experimental structure used in this research involves real-time control of multiple plants connected to the controllers over the ZigBee network.

Index Terms— **Client–Server Architecture, ZigBee, Network Control System, Real- Time System, SSH**

I. INTRODUCTION

With the expanding scale of mastery systems in manufacturing or factory field, the web server as a key to attach subsequent apparatus and nexus to attain the minimal cost. By introducing internet in to mastery network, it is feasible to attain remote sensing monitoring and supervising for equipments. The data can be interchanged between the computers which are prolonged extent across the internet connected to the World Wide Web. The mandatory requirements of transmission on to computer networks are fortified by cryptographic indemnity. Encryption is what furnishes communication with privileged; the guarantee that imparts data is only read by the legatee and not by an intruder. Authentication of users and data is transferred by message-authentication codes and digital signatures. The certainty of these ramifications hangs on the fact that an authorized person knows some undisclosed information, a key not familiar or unknown to assailants. If attackers by any means figure out this key, they can fully infringement the system's reliability.

***Leeja Joseph**, M.Tech Student, Department of ECE,MG University, Kottayam*
***Doniya Jose**, Asst.Professor, Department of ECE, MG University, Kottayam*

Secure Shell (SSH) ([2]-[6]) is a protocol for assured network transmissions depicted to be comparatively uncomplicated and economical to implement. The SSH, concentrated on furnishing a reliable inaccessible logon provision to substitute Telnet and other remote logon strategy that impart no security. SSH also offer an additional extensive client-server proficiency and can be used to secure such network roles as file transfer and e-mail. SSH client and SSH server approaches are far apart accessible for most operating systems. It has get the technique of alternative for inaccessible or remote login and is promptly be transformed into one of the most ubiquitous approaches for encryption technology outside of embedded systems. SSH is well ordered as three protocols specifically, Transport Layer Protocol, User Authentication Protocol, and Connection Protocol that typically run on top of TCP.

In this research, real-time control of multiple clients connected to the controllers over the ZigBee network. ZIGBEE ([7]-[8]) is a wireless technology established as an open global standard to undertake the peculiar needs of low-cost, low-power, wireless sensor networks. The ZigBee protocol was delineated to carry data through the aggressive RF environments that commonly exist in commercial and industrial applications. ZigBee enables broad-based arrangement of wireless networks with low-cost, low-power solutions. It extends the ability to run for some years on inexpensive batteries for a host of examining applications.
The paper is organized as follows: in Section II the related work about client server architecture. Section III will describe the proposed system architecture. In Section IV presents the results and discussion and Section V will describe the conclusion and future work
.

II. RELATED WORK

In this section, the existing client/server network architecture is discussed.

This paper shows the client- server architecture based on SSH [9] protocol can be used as a real time communication for possible implementation in factory automation.

Ambike [11] designed a single-client-single-server real-time architecture that had the ball maglev system attached to the client computer and the controller program running on the server computer.

Lee [12] additionally researched and expanded the test bed to a multi-client-single server. Increasing the number of clients but sustaining one server led to divergent challenges in the system and Lee proposed distinct sampling periods based on bandwidth utilization and performance of the plant.

Fig. 2. Block Diagram of C/S System

Kim [1] examined and enlarged the test equipment to a multi-client-multi server architecture. Increasing the number of servers show to reduce the obstacles introduced in the multi client single server architecture and performs better than other two architectures.

Compared to the Existing research, this paper aims to easier the users to monitor and control the motor in industries by computer using TCP/IP connection. By using SSH protocol we can log into a remote machine and execute commands and can provide a secure path over the Internet, through a firewall to a virtual machine. Hence expanding a cost effective and high efficiency controller is designed in this system.

III. PROPOSED SYSTEM ARCHITECTURE

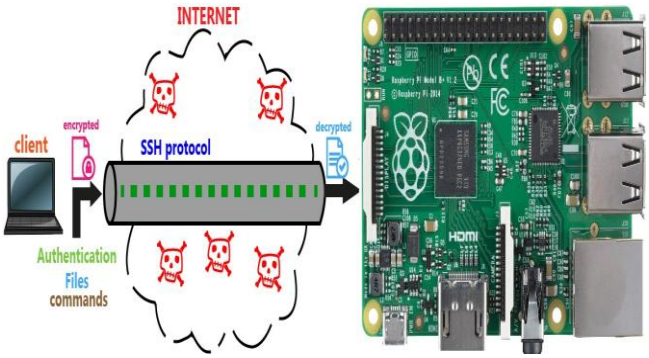


Fig. 1. Architecture of C/S System using SSH protocol

Secure Shell (SSH) is a protocol for reliable network transmissions designed to be comparatively manageable and economical to implement. SSH client and server applications are extensively accessible for almost all operating systems. Fig 1 shows the communication between the client system and server system. In this paper the client is a pc and the server is a raspberry pi board. Before consequential communication can occur, the SSH client and server must establish a reliable connection. This lets them share keys, passwords, and finally, whatever data they transmit to each other

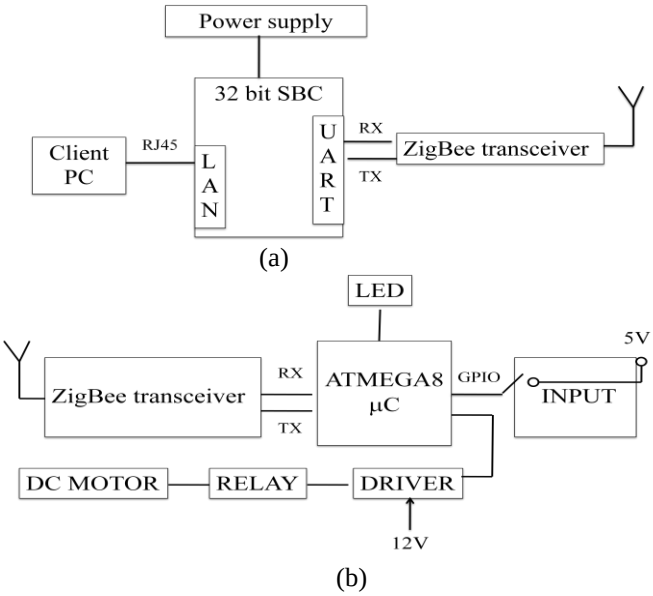


Fig 2 shows the block diagram of the client server architecture using SSH protocol. In this architecture the client PC communicates with server (raspberry pi board). In this the client server communication is demonstrated using a PCB connected with dc motor and a led. By using a wireless ZigBee protocol architecture, the PCB and the C/S system communicates

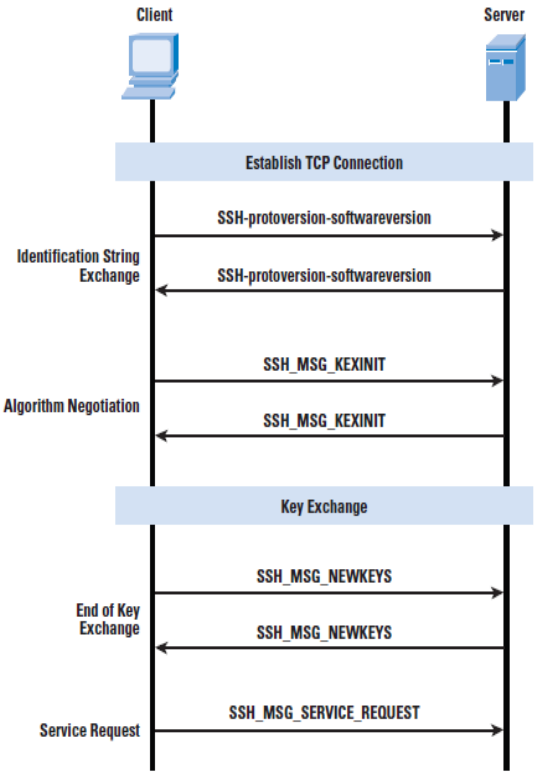


Fig. 3. Packet Exchanges of SSH TLP

Figure 3 illustrates the pattern of occurrence in the SSH Transport Layer Protocol (TLP). First, the client establishes a reliable TCP connection to the server with the TCP protocol and is not part of the TLP. When the connection is fixed, the client and server exchange data, referred to as packets.

A. Software Architecture

Linux Ubuntu- The operating system manages the communication between your software and your hardware
TShark- TShark is section of the Wireshark distribution. Wireshark is a network "sniffer" - that captures and analyzes the packets off the wire. TShark is a network protocol analyzer.
Qtiplot- Qtiplot is a program for two- and three-dimensional graphical representation of data sets and for data analysis

B. *Hardware Architecture*

Fig 4 shows the hardware structure of our client/ server system based on SSH protocol. The hardware modules used in our system are:

Atmega8- The low-power Atmel 8-bit AVR RISC-based microcontroller

Regulated Power Supply- It is an embedded circuit; this converts unregulated AC into a constant DC.

Raspberry Pi 1 Model B+ - is a series of credit card–sized single-board computers

Cc2500 Transceiver Rf22 2.4 GHz (ZigBee) - is a low-cost 2.4 GHz transceiver designed for very low-power wireless applications.

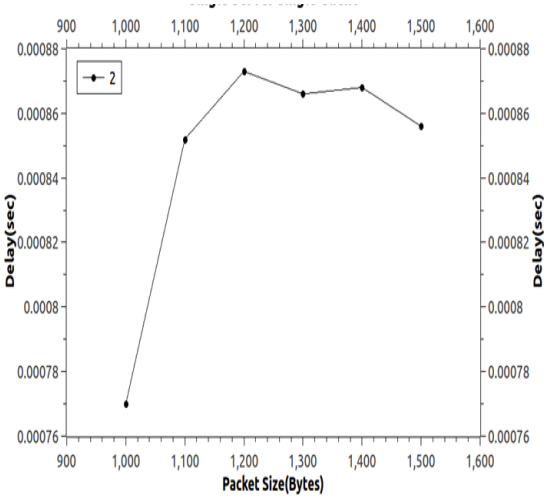
Fig. 4. Hardware Unit of System Architecture

IV. OBSERVATION AND RESULTS

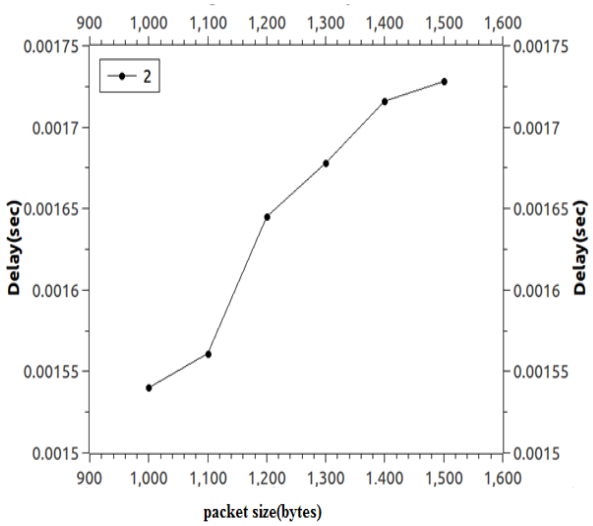
Experiments were delineated to bear maximum coverage with respect to network load, utilization, and packet loss. To create scenarios of real-time and non-real-time network utilization, the experimental procedure is divided into two types of experiments. A non-real-time scenario is conducted by sending larger packet sizes at smaller transmission rates, and a real-time scenario, by transmitting smaller sized packets at higher packet rates.

A. *Increasing the Packet Length While Keeping the Packet Rate Constant*

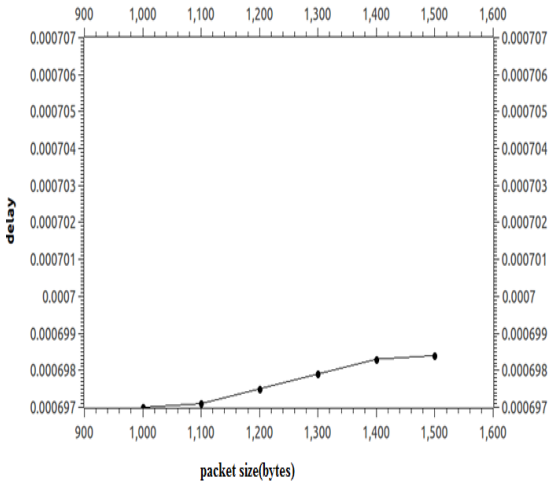
In each section of experiments, the packet rate was permanent, and the packet length was increased for every successive iterations. The packet length was increased from 900 bytes to 1400 bytes in steps of 100 bytes. This testing method verifies a framework of non-real-time network utilization. Experiments were applied to three architectures, and Fig. 5. (a-c) are labeled accordingly



(a)



(b)

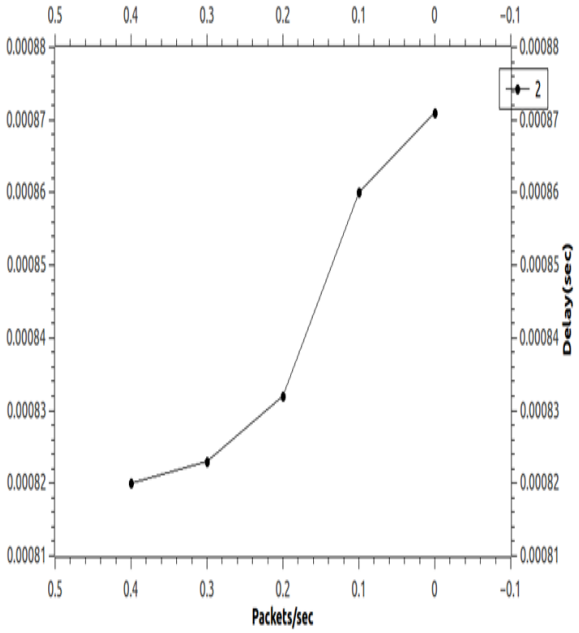


(c)

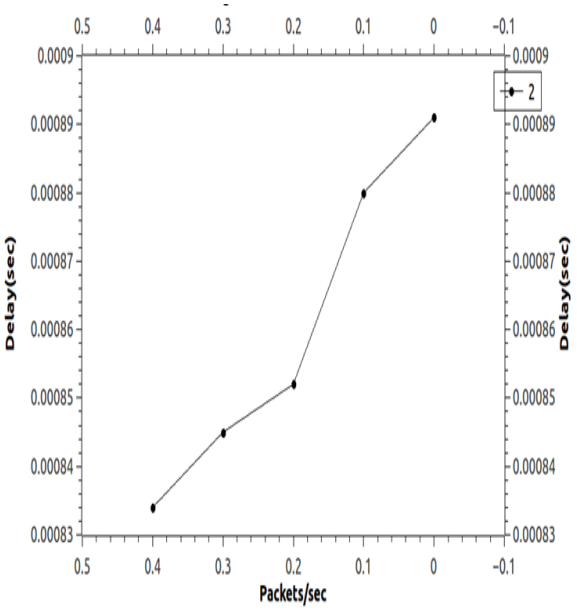
Fig. 5. Average Delay Vs Packet Size of (a) SC-SS (b) MC-SS (c) MC-MS

B. Increasing the Packet Rate While Keeping the Packet Length Constant

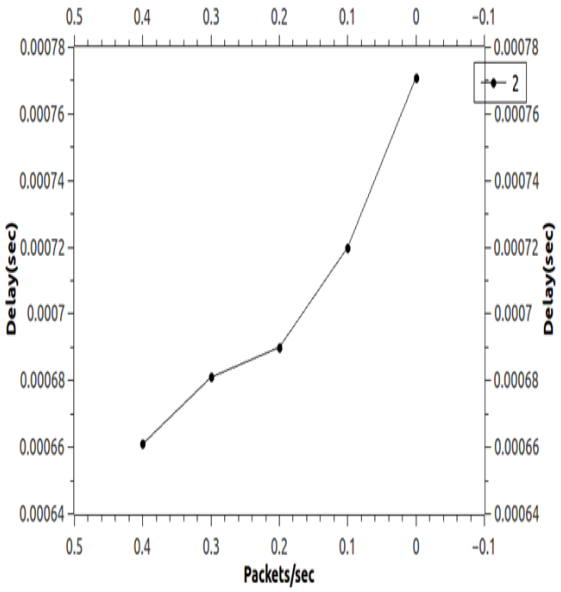
The packets are transmitting at an increasing packet rate but a constant length of 64 bytes per packet is retained. On every successive iteration, the packet rate was increased from 1000 to 5000 packets/s in steps of 1000. Due to the physical limitation of the sending and receiving computers the maximum packet rate was fixed at 5000 packets/s. This testing method verifies a scenario of real-time network utilization. The experiments were applied to three architectures, and Fig 6. (a-c) are labelled accordingly



(a)



(b)

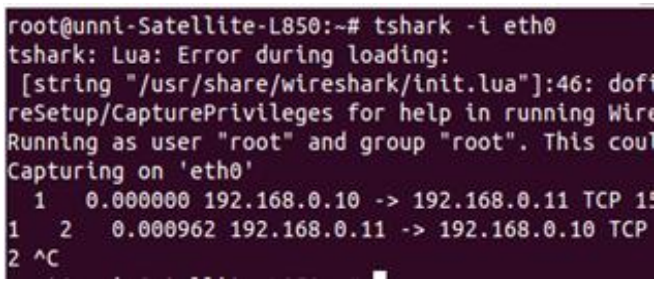


(c)

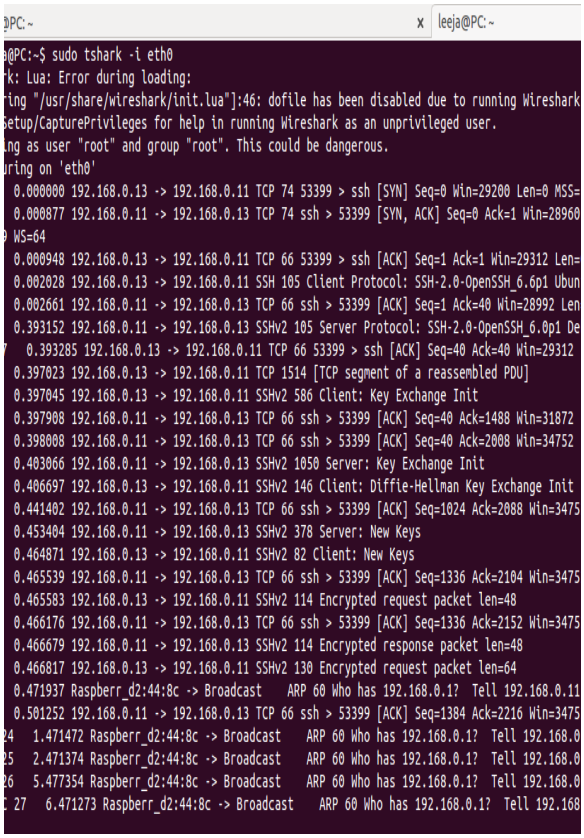
Fig. 6. Average Delay Vs Packet Rate Of (a)SC-SS (b) MC-SS (c) MC-MS

C. Client Server Communication Using SSH Protocol

Fig 7 a shows the communication between the server and client using TCP protocol and fig 7 b shows the communication between client server using ssh protocol. From this we can compare and analyze that the how secure the ssh protocol is. Finally fig 8 shows the screenshot of communication between the zigbee and the client server system,



(a)



(b)

Fig .7. C/S Communication Using (a) TCP (b) SSH

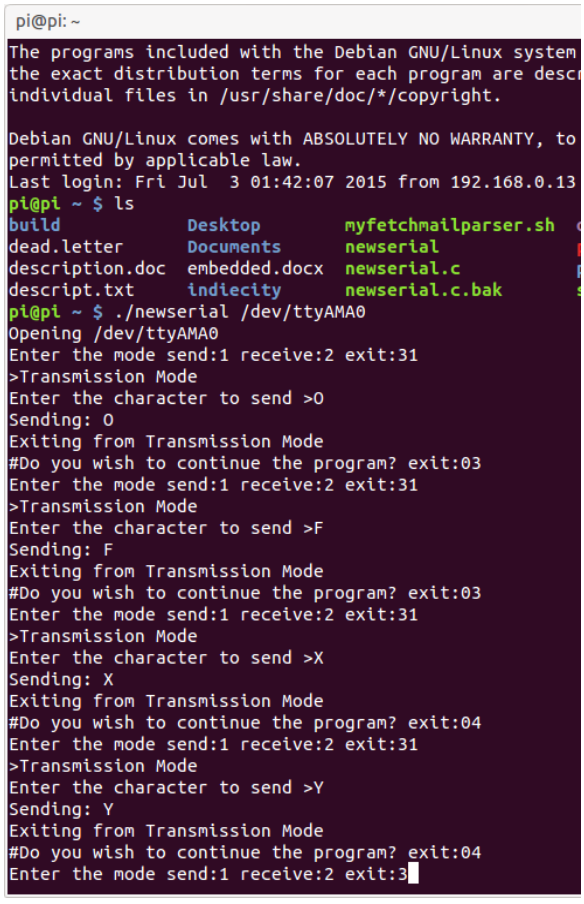


Fig 8. Screenshot of C/S Communication

V.CONCLUSION & FUTURE SCOPE

In this paper, we introduced a summary of distributed control system security using SSH protocol. Secure Shell is a protocol used to log into another computer over a network, to accomplish commands in a distant machine, and to transfer files from one machine to another. It issues powerful authentication and firm communications over uncertain channels. The SSH protocol consists of 3 components: The Transport Layer Protocol, which provides server authentication, confidentiality, and integrity with ultimate forward privateness. The User Authentication Protocol, which authenticates the client to the server. The Connection Protocol that multiplexes the encrypted subway into some logical channels. Also in this research, zigbee is used for the communication between the PCB and the client/ server system. ZigBee is a wireless matrix standard that is focused at remote control and sensor applications which is acceptable for functioning in coarse radio environments and in unreachable locations. The future scope for this thesis is to analyse the average delay of the various client server architectures, i.e. SC-SS, MC-SS, MC-MS architectures using SSH protocol and compare the results with client server architectures using TCP protocol.

REFERENCES

[1] Naveen Bibinagar , Won-jong Kim, —"Switched Ethernet-Based Real-Time Networked Control System with Multiple-Client–Server Architecture", IEEE/ASME Trans. Mechatronics, vol. 18, no. 1, pp. 104-112, February 2013.

[2] Oliver Gasser, Ralph Holz, Georg Carle," A deeper understanding of SSH: Results from Internet-wide scans", Network Operations and Management Symposium (NOMS), 2014 IEEE.

[3] Maurizio Dusi, Francesco Gringoli, Luca Salgarelli,"A Preliminary Look at the Privacy of SSH Tunnels", Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference.

[4] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture," RFC 4251, IETF, Jan. 2006.

[5] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol," RFC 4253, IETF, Jan. 2006.

[6] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Connection Protocol," RFC 4254, IETF, Jan. 2006.

[7] Li Zheng,"ZigBee Wireless Sensor Network in Industrial Applications",SICE-ICASE, 2006. International Joint Conference.

[8] Chengbo YU1, Yanfei LIU1, 2, Cheng WANG ,"Research on ZigBee Wireless Sensors Network Based on Modbus Protocol".

[9] Shiguo Lian, Guest Editorial "Secure Multimedia Communication".

[10] "Open SSH." [Online]. Available: http://www.openssh.org.

[11] A.Ambike, "Closed Loop Real-Time Control on Distributed Networks," M.S. Thesis, Texas A&M University, 2004.

[12] M. Lee, "Real-Time Networked Control with Multiple Clients," M.S. Thesis, Texas A&M University, 2009.