CHAPTER ONE

INTRODUCTION

1.1 Introduction:

In computer networks, communication occurs between entities in different systems. An entity is anything capable of sending or receiving information. However, two entities cannot simply send bit streams to each other and expect to be understood. For communication to occur, the entities must agree on a protocol. A protocol is a set of rules that govern data communications. A protocol defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics, and timing. The term syntax refers to the structure or format of the data, meaning the order in which they are presented. For example, a simple protocol might expect the first 8 bits of data to be the address of the sender, the second 8 bits to be the address of the receiver, and the rest of the stream to be the message itself. The word semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation? For example, does an address identify the route to be taken or the final destination of the message?

The term timing refers to two characteristics: when data should be sent and how fast they can be sent. For example, if a sender produces data at 100 Mbps but the receiver can process data at only 1 Mbps, the transmission will overload the receiver and some data will be lost [1].

A Protocol Converter is a device used to convert standard or proprietary protocol of one device to the protocol suitable for the other device or tools to achieve the interoperability. Protocols are software installed on the routers, which convert the data formats, data rate and protocols of one network into the protocols of the network in which data is navigating. There are varieties of protocols used in different fields like Power Generation, Transmission & Distribution, Oil & Gas, Automation, Utilities, AMR, and Remote Monitoring applications. The major protocol translation messages involve conversion of data messages, events, commands, and time synchronization.

The general architecture of a protocol converter includes an internal master protocol communicating to the external slave devices and the data collected is used to update the internal database of the converter. When the external master requests for data, the internal slave collects data from the database and send it to the external master. There will be different schemes for handling the spontaneous reporting of events and commands. There can be different physical medium for communication on protocol-X & Y, which include RS-232, RS-485, Ethernet, etc.

Protocol Converter applications vary from industry to industry and it can be a software converter, hardware converter, or an integrated converter depending on the protocols. Some of the key applications are: Substation Automation, Building Automation and Process Automation. The major protocols used in each area of application are listed under list of automation protocol.

Protocol Converters are generally used for transforming data and commands from one device or application to another. This necessarily involves transformation of data, commands, their representation, encoding and framing to achieve the conversion.

There are simple and complex types of conversions depending on the application and domain in which this is being used. The simplest and most commonly used conversion is protocol conversion between modbus RTU and Modbus TCP. In this conversion, there is no change in the overall framing. it is easy to take the Serial Modbus RTU frame and encapsulate it in a TCP/UDP socket and send it over Ethernet. Since both the protocol framings are the same, except for the actual physical layer transmission, both the application layers will interpret data similarly as long as the communication interfaces are made transparent.

1.2 Problem Statement:

The problem of this project is incompatibility between Ethernet and RS232. Incompatibility means differ of packet format so that In serial communication (R-S232) need additional fields for ‘ start of packet’ and ‘end of packet’.

1.3 Proposed Solutions:

Protocol conversion has become a necessity for the internetworking in a heterogeneous network environment. In this project implementation of protocol converter to solve incompatibility between protocols is implemented in order to facilitate networking and serial communication without having to worry about programming at hardware level.

1.4 Research Aim and Objectives:

Aim: to implementation protocol converter between Ethernet and RS232.

Objectives:

1. To develop conversion software to enable two system with different protocols to communicate.
2. To convert packet type form Ethernet to R232 and vise verse to enable data and command transmission.

1.5 Methodology:

A segment containing three pc devices .on pc1 a java program will be running which transmits file over Ethernet link to pc2 then pc2 runs a C program that captures the Ethernet packets and converts them into serial format, pc3 is just to check whether software is running correctly or not.

1.6 Research Outline:

Chapter one: Introduction.

Chapter two: Literature Review.

Chapter three: Methodology.

Chapter four: Results and Discussion.

Chapter five: Conclusion and Recommendations.