GMIT
GALWAY-MAYO INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

Implementing TCP/IP in SCADA Systems

In One Volume

Denis McHugh B. Eng.

August 2003

Submitted for the Degree of
Master of Engineering

Submitted to:	Galway-Mayo Institute of Technology, Galway, Ireland.
Research carried out at:	Galway-Mayo Institute of Technology.
Research Supervisor:	Dr. John Owen-Jones.

Dedicated to my Parents

Declaration

I hereby declare that the work presented in this thesis is my own and that it has not been previously used to obtain a degree in this institution or elsewhere.

Denis McHugh.

1st August 2003.

Statement of Confidentiality

The material contained in this thesis should not be used, sold, assigned, or disclosed to any other person, organisation or corporation without the permission of:

Galway-Mayo Institute of Technology: Contact: Dr. John Owen-Jones
Tel: +353 91 742202
Dr. John Owen-Jones email:
john.owen-jones@gmit.ie

Contact: Mr. Michael Murray
Tel: +353 91 742298
Michael Murray email:
michael.murray@gmit.ie

Datac Control International Ltd., Dublin Contact: Mr. Michael O' Gara
Tel: +353 1 6717377
Michael O' Gara email:
mogara@datac-control.com

Prologue

The research described in this thesis has been carried out over a 20 month period as part of a college/industry partnership project. This project, the 922 FlashNet RTU, was funded under the Innovation Partnerships research grants scheme administered by Enterprise Ireland. The project is a partnership between the Galway-Mayo Institute of Technology and Datac Control International Ltd., a Supervisory Control and Data Acquisition (SCADA) company. The aim of the project was to research and develop a Remote Terminal Unit (RTU) which makes use of the TCP/IP suite of protocols and common web applications.

Abstract

The development and popularity of the Internet and its applications over the past number of years has made Transmission Control Protocol/Internet Protocol (TCP/IP) one of the most commonly used protocols in communicating data across networks. Supervisory Control and Data Acquisition (SCADA) systems are used for gathering and analyzing of real-time data from remote locations. SCADA systems are used to monitor and control plant and equipment in industries such as telecommunications, water, waste control, energy, oil and gas refining and transportation. However SCADA is an industry to which TCP/IP has had limited impact.

This thesis looks at the use of TCP/IP in a Remote Terminal Unit (RTU). An RTU is a device installed at a remote location that collects data, codes it into a format suitable for communications and sends it back to a master station. The master station can also control any of the outputs on the RTU. An RTU contains input channels for sensing and metering and output channels for control. The developed system will contain all the normal components of a normal RTU such as digital and analog inputs and outputs, a database and communications ports. In addition the RTU will also act as an IP router. It will have an Ethernet interface, a radio interface and two Point-to-Point Protocol (PPP) interfaces. A network of RTUs can be created using a radio link. One or more of the RTUs can have a link to a master station either over Ethernet or over PPP using a modem. The RTU will also support distributed I/O using Modbus.

This thesis also looks at uses of common web applications such as Hyper-Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), Email and Telnet in an RTU. The use of SMS and Email for alarm notification directly from an RTU will also be examined.

Acknowledgments

There are a number of people to whom I am extremely grateful for their help and encouragement these past twenty months…

Dr. John Owen Jones and Michael Murray, my research supervisors for all your assistance, ideas, guidance and the opportunity to study at GMIT. You were always very approachable and I wish you all the best in the future.

Robert Farrell of Datac Control International Ltd. for all your help and support with the printed circuit board layout and manufacture.

Stewart Buchanan and Cyril Kerr of Datac Control International Ltd., for the initial ideas and for supporting the project to the end.

Thanks to all of the staff in GMIT, in particular thanks to the following people: Gerard Mac Michael, Frank Mc Curry, Tom Roche, Caroline Joyce, Ann Duggan John Sherry, Tom Frawley, and Mary Creaven.

A special acknowledgement to my postgraduate friends Cristi Cocor and Valerie Butler. Sincere best wishes to you all in your personal and professional lives.

Finally, and most importantly my wonderful parents, Willie and Patricia. Thank you both for the guidance and encouragement you have given me throughout my life.

Table of Contents

List of Figures and Tables

Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Tables

Chapter 3

Chapter 4

Chapter 5

Glossary

ADSL: Asymmetric Digital Subscriber Line, a technology which allows high speed data (up to 7Mbps) such as Internet access, video on-demand and interactive TV over a normal phone line. It's called asymmetric as the download speed is typically greater than the upload speed.

AIP: Analog Input, an input on a Remote Terminal Unit (RTU) which measures voltage. Analog inputs can be used to measure temperature, pressure and flow using appropriate sensors.

AM: Amplitude Modulation, technique used in radio communications which varies the amplitude of a carrier signal in proportion to the voltage of a modulating baseband signal. AM allows signals such as voice and data to be sent over radio.

AMR: Automatic Meter Reading, emerging technology used to replace the "meter man" by retrieving customer usage data of electricity, gas and water over a phone line.

AOP: Analog Output, output voltage signal from a Remote Terminal Unit (RTU) which can be used to control a threshold input of an alarm sensor.

ARP: Address Resolution Protocol, a protocol used within a TCP/IP stack to resolve IP addresses into MAC addresses.

ASCII: American Standard Code for Information Interchange, system which allows numbers, letters of the alphabet,

punctuation and control characters to be represented using seven or eight bit numbers.

BPSK: Binary Phase Shit Keying, digital radio communications technique which changes the frequency of a carrier signal depending on the logic state of the digital data stream.

Binary: Used in computers and digital systems to represent numbers using 1s and 0s.

Broadcast: A message which can be sent on a network and is received by all stations.

CDMA: Code Division Multiple Access, a form of wireless multiplexing which uses a "frequency hopping" technique. Frequency hopping is the continuous changing of the carrier frequency at random.

CHAP: Challenge Handshake Authentication Protocol, an authentication protocol used in Point-to-Point Protocol (PPP).

CSMA/CD: Carrier Sense Multiple Access/Collision Detect, multiple access technique in which stations wait until a medium is "quiet" before transmitting. If more than one station transmits at the same time it is detected and both station "back-off" before trying again at a random number of milliseconds.

Datagram: Internet Protocol (IP) message which is sent between IP modules on different hosts.

Datapoint: A record in an RTU database which represents an input or an output. The I/O can be local (built in) or distributed.

DFWMAC: Distributed Foundation Wireless Medium Access Control, multiple access technique used in wireless networks.

DHCP: Dynamic Host Configuration Protocol, used by computers on a network to obtain an IP address from a DHCP server.

DIP: Digital Input, an input of a Remote Terminal Unit (RTU) which can read a logic 0 or 1 value.

DNP3: Distributed Network Protocol version 3, SCADA protocol used to control and monitor I/O on remote devices.

DNS: Domain Name System, a mechanism for resolving domain names such as www.domain.com into IP addresses.

DOP: Digital Output, an output of a Remote Terminal Unit (RTU) which can have a logic 0 or 1 state.

Email: Based on Simple Mail Transport Protocol (SMTP) and Post Office Protocol version 3 (POP3), allows messages and files to be sent over the Internet to selected recipients.

Ethernet: Network physical/data layer technology used to transfer frames of up to 1500 bytes.

FAT16/32: File Allocation Tables, file system used to store data on DOS and Windows PCs.

FDDI: Fibre Distributed Data Interface, physical/data-link layer network technology which uses fibre optic cabling.

FDMA: Frequency Division Multiple Access, technique in where the electromagnetic spectrum is split in to channels. To allow

multiple access, each user or pair of stations use a different channel.

FM: Frequency Modulation, radio communications technique in which the frequency of a carrier is changed in proportion the voltage of a modulating baseband signal and is used for voice and data communications.

FSK: Frequency Shift Keying, digital radio communications technique which changes the frequency of a carrier signal depending on the logic state of the digital data stream.

FTP: File Transfer Protocol, used to transfer files over the Internet.

GPRS: General Packet Radio Service, always-on high speed Internet connection which uses Global System for Mobile communications (GSM).

GPS: Global Positioning System, system for determining the geographical location (longitude, latitude and altitude, speed) using the triangulation of signals received from satellites.

GSM: Global System for Mobile communications, digital mobile phone communications standard.

HART: Highway Addressable Remote Transducer uses a traditional 4-20mA measurement or control signal with a superimposed bi-directional digital communications signal that provides additional information about the field device.

HDLC: High-level Data Link Control, data-link layer protocol used to transfer frames using a "byte stuffing" mechanism.

HMI: Human Machine Interface, software used in a master station to monitor and control I/O on a remote device such as a Remote Terminal Unit (RTU)

HTML: Hyper Text Mark-up Language, language used to create web pages.

HTTP: Hyper Text Transfer Protocol, used for transferring HTML files and media over the Internet to a web browser.

ICMP: Internet Control Message Protocol, protocol used to detect, report and resolve host and routing problems on the Internet.

IDE: Integrated Device Electronics, used in PCs to connect motherboards to mass storage devices such as hard disk drives and CDROMs.

ISDN: Integrated Services Digital Network, digital dual phone line which allows up to 128 kbps data transfer.

ISP: Internet Service Provider allows users to access the Internet using a PC, a modem and phone line.

LAN: Local Area Network, several computers connected on one segment of a network.

LCP: Link Control Protocol, used in Point-to-Point Protocol (PPP) to negotiate link parameters such as compression and authentication protocol to use.

MAC: Medium Access Control, methodology that allows devices on a LAN to share their interconnecting media.

Modbus: SCADA master/slave protocol used to control and monitor the input and output channels on devices such as Remote Terminal Units (RTUs).

Multicast: Similar to a broadcast, a packet can be sent to several computers which have joined a "multicast group".

NAT: Network Address Translation, used in Internet routers by many companies and educational institutions to allow several computers with private IP addresses to access the Internet using just one public IP address.

NCP: Network Control Protocol, used in Point-to-Point protocol to configure network layer protocols such as IP.

NIC: Network Interface Card, a Industry Standard Architecture (ISA) or Peripheral Component Interface (PCI) card with an on-board Ethernet controller which enables a PC to access a network.

OSI: Open Systems Interconnect, a model on which multi-layer communications protocols such as TCP/IP can be modelled on.

OSPF: Open Shortest Path First, dynamic routing protocol used in routers to generate routing tables automatically.

Packet: Short message sent and received by network hardware.

PAP: Password Authentication Protocol, used in Point-to-Point Protocol (PPP) for user authentication.

PLC: Programmable Logic Controller, a device with in which its outputs are logic functions of its inputs.

PPP: Point to Point Protocol, used over serial cables or a modem and a phone line to connect a PC to a remote network such as the Internet.

PSTN: Public Switched Telephone Network, the analog phone system used in most homes.

QAM: Quadrature Amplitude Modulation, similar to AM except the amplitude can take either one of four values. This allows greater bit rates for a given bandwidth.

QNX: Highly stable Linux variant operating system used by RealFlex.

QPSK: Quaternary Phase Shift Keying, similar to BPSK except the angle (frequency) can take either one of four values. This allows greater bit rates for given bandwidth.

RealFlex: Human Machine Interface (HMI) software used in master stations to control and monitor inputs and outputs on devices such as Remote Terminal Units (RTUs).

RFC: Request For Comments, documents which specify protocols such as TCP/IP, HTTP, FTP, etc. and are widely distributed on the Internet

RIP1/2: Routing Information Protocol version 1/2, dynamic routing protocol used in routers to generate routing tables automatically.

Router: A device connected to one or more networks such as Ethernet or FDDI. It allows the routing of IP datagrams between interfaces.

RTOS: Real Time Operating System, an operating system that has been developed for real-time systems and is typically used for embedded applications.

RTU: Remote Terminal Unit, a SCADA device with digital and analog inputs and outputs and a communications link to a mater station. Using a Human Machine Interface (HMI), the I/O can be controlled and monitored over the communications link.

SCADA: Supervisory Control And Data Acquisition systems are used in industry to monitor and control plant status.

SMS: Short Message Service, allows small messages to be sent and received using Global System for Mobile communications (GSM) mobile phones.

SMTP: Simple Mail Transfer Protocol, protocol used for transfer of email messages from a client program such as Microsoft Outlook to a server for delivery.

SNMP: Simple Network Management Protocol, used by administrators to monitor, debug, control and configure network devices remotely.

Tag: A record in an RTU database which represents an input or an output. The I/O can be local (built in) or distributed.

TCP/IP: Transmission Control Protocol/Internet Protocol, protocol suite which forms the foundation of the Internet.

Telnet: A version of Terminal which runs over a TCP socket instead of a serial communications link and hence can be used over a network or the Internet.

Terminal: Simple ASCII application that allows users to connect into a computer over a serial communications line such as RS232. The user can type and execute commands (such as Linux or DOS) and view the responses on a terminal screen.

Token Ring: Network topology in which computers are connected in a ring. To transmit a packet, a computer must have a "token" which is passed around the ring from node to node.

UDP: User Datagram Protocol, simple connectionless transport-layer protocol.

WAN: Wide Area Network, two or more Local Area Networks (WANs) connected via routers.

WAP: Wireless Application Protocol, technology used in Global System for Mobile communications (GSM) mobile phones to access the Internet.

WAP: Wireless Access Point, a device which acts as a bridge between a wireless network (e.g. IEEE 802.11) and wired network such as Ethernet. Packets sent on the wired network are transferred to the wireless network and vice versa.

Chapter 1

Introduction

1.1 Thesis Motivation

1.2 Thesis Objectives

1.3 Research Methodology

1.4 Thesis Structure

1.1 Thesis Motivation

The last decade has seen the explosion of the Internet and it associated technologies. The Internet is based on the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of communications protocols. The use of these protocols has had some impact on the Supervisory Control and Data Acquisition (SCADA) industry which has been around for much longer. SCADA systems are used to control and monitor plant and equipment in industries such as telecommunications, water, waste and manufacturing control, energy, oil and gas refining and transportation. The basic components of a simple SCADA system are a Remote Terminal Unit (RTU), a communications link and a Human Machine Interface (HMI – master station) as given in figure 1.1.

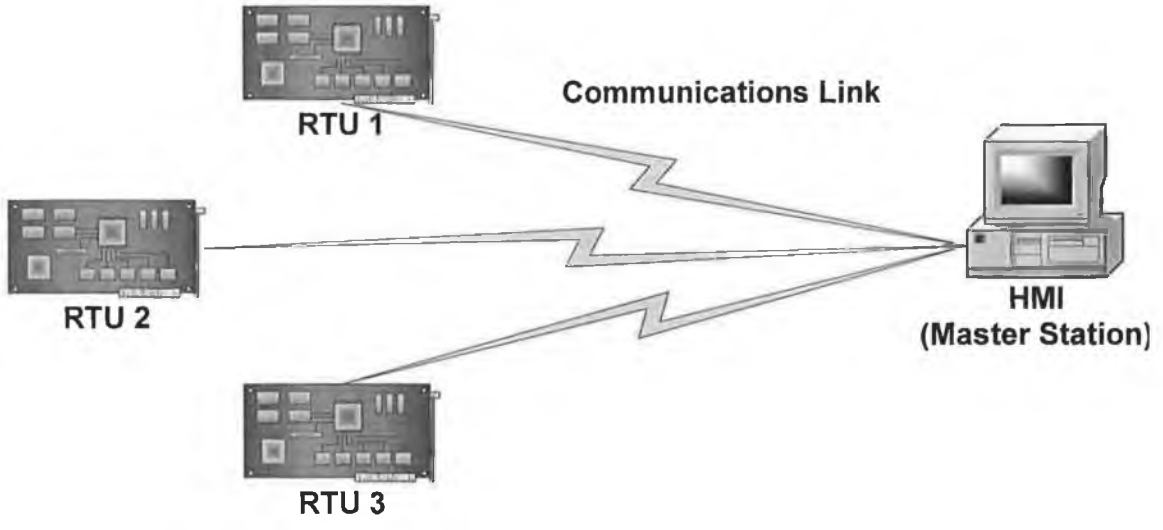

Figure 1.1 A basic SCADA system.

An RTU is a device with analog and digital I/O and a communications link to a master station. The master station through a Graphical User Interface (GUI) allows an operator to

control and monitor the I/O on any of the RTUs in real time. The simplest form of communications link can be a serial cable. Many non-IP based protocols exist such as Modbus and Distributed Network Protocol (DNP) for polling the I/O from the master station over the communications link. The impact of TCP/IP in SCADA has primarily been in the control room through HMIs which can be accessed over a network. For example the HMI can have a built in web server displaying over a web browser the status of the I/O polled from RTUs. However communications from the master station to the RTU still uses non-IP protocols such Modbus, Distributed Network Protocol (DNP) or some other propriety protocol.

Very few RTUs on the market today support TCP/IP. TCP/IP has found its way in to many embedded systems and devices such as digital cameras and mobile phones. Given the explosion in devices with Internet connectivity, it is inevitable that TCP/IP will become one of the main communications protocols in many RTU systems. There are so many applications today associated with the Internet such as Web, File Transfer Protocol (FTP), Telnet and Email. Some of these applications may be of more use than others in an RTU. An RTU that supports these applications would be easier to configure, use and maintain. Given most PC users today are familiar with the Internet and its applications; it would take less time to train operators and installation personnel of such a system.

1.2 Thesis Objectives

This thesis is based on the work done in developing the 922 FlashNet project. The main objective of this project is to develop three prototype RTUs that support TCP/IP. Each RTU will be a fully functional and will contain analog and digital input and output and will support the ability to generate an alarm when an input goes outside preset limits. The RTU will contain an Ethernet port, radio and Point-to-Point Protocol (PPP) interfaces. Internet applications such as Web, FTP, Email and Telnet will also be implemented along with the SCADA protocol Modbus and its TCP variant. The system will contain a mass storage device complemented with a file system.

The main objective of the thesis is to examine the use of TCP/IP and the most common Web applications in an RTU. The thesis also looks at the use of TCP/IP over radio in an RTU. The issues of implementing TCP/IP in an RTU which periodically goes to a low power sleep mode will also be examined. Low power sleeping RTUs are used in applications where the system runs from battery in a remote location and there is no mains power source. The RTU can spend most of the time in a low power sleep mode and then

wake to do processing and report any alarms. None of the few RTU systems currently available with IP connectivity support going to sleep periodically.

An RTU that supports TCP/IP, its associate technologies and a mass storage device has many applications. The system could easily be used for mass data logging and given support for radio links, this can be done remotely. It can replace many of the non-IP based RTUs on the market today and has the advantage that it can be easily connected to company networks and the Internet. Furthermore since it can be configured using common applications such as a web browser, it simplifies installation and maintenance of the system.

1.3 Research Methodology

The approach to the research is as follows:

Literature Review

- **TCP/IP**
  - Introduction, history, concepts and protocols.
  - Wireless IP.
  - Embedded Networking.
  - Security.
  - Dynamic routing.
  - Future of IP.
- **SCADA**
  - Introduction.
  - Protocols.
  - TCP/IP in SCADA systems.

Requirements Analysis

- Analyze current RTU systems and common SCADA protocols.
- Identify short comings in the RTU systems which could be resolved using IP connectivity.
- Develop a list of requirements for the proposed TCP/IP RTU.

System Development

- Develop three fully functional RTUs that support TCP/IP and common Web applications.

Test and Validation

- Evaluate the implemented system.
- Compare with the old system.

- Obtain SCADA expert feedback.

1.4 Thesis Structure

A brief description of the structure of the thesis is given below and is illustrated in figure 1.2.

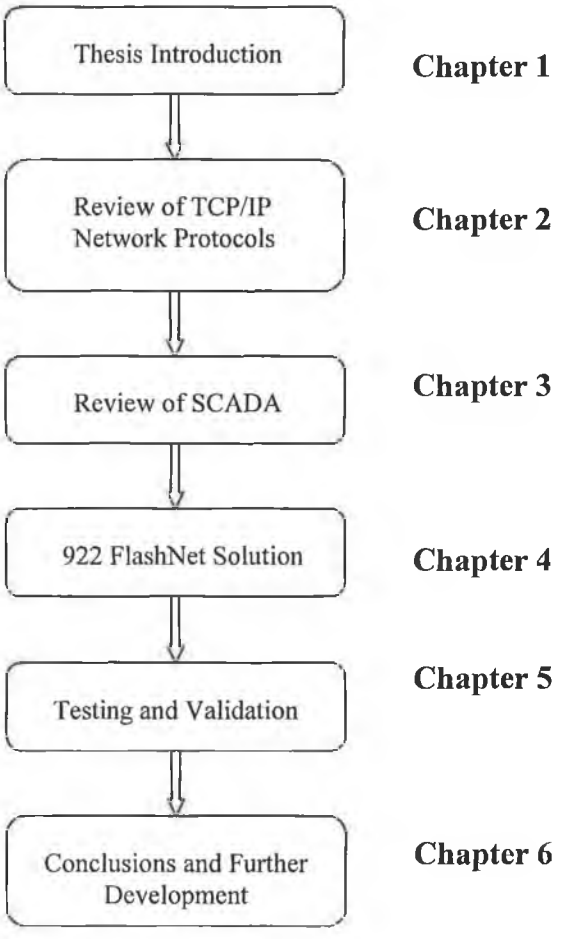

Figure 1.2 Thesis structure.

Chapter 1 presents the thesis motivation, objectives, methodology and structure.

Chapter 2 contains a literature review of TCP/IP network protocols. The chapter looks at the TCP/IP suite of protocols in detail. Network technologies such as Ethernet and PPP will be discussed. The use of TCP/IP over radio links will also be discussed. A review of TCP/IP in embedded systems will be given. The security of the protocols will be examined along with security technologies. Two dynamic routing protocols will also be examined.

Chapter 3 gives an introduction to SCADA and defines some of the common terms. The main SCADA protocols will be examined as well as a proprietary one. The use of TCP/IP in an RTU will also be investigated.

Chapter 4 details the solution, the 922 FlashNet RTU. Firstly improvements using TCP/IP that could be made to the RTU systems discussed in the literature review will be listed. From this a set of functional and non-functional requirements are derived. Then all aspects of the design and implementation of the solution are given.

Chapter 5 looks at the testing and validation of the 922 FlashNet RTU. The methods used to test the system are presented along with results and feedback from SCADA experts.

Chapter 6 concludes with a summary of the work done, the conclusions drawn from the research and recommendations for further development of the solution.

Chapter 2

Review of TCP/IP Network Protocols

2.1 Introduction

The development and popularity of the Internet and its applications over the past number of years has made TCP/IP one of the most commonly used protocols in transferring data between networks both local and international. This chapter focuses on TCP/IP; in particular issues which would affect its use in SCADA based systems will be identified. An overview of the operation of TCP/IP and network technologies will be given along with its use in embedded systems. The chapter also looks at security issues of TCP/IP. Usage of the protocols over radio will be discussed along with a discussion on some of the various radio technologies. Finally the future of TCP/IP will be discussed.

2.2 History of TCP/IP

Although to many people the Internet is a relatively new technology, the underlying protocols TCP/IP however have been around many years. The need for such a protocol was recognized in the late 1960s by the U.S. Department of Defense (DOD). The

department had a growing problem of communicating large amounts of information between its staff, research labs, universities and contractors. The various groups had different computer systems from different manufacturers which had different operating systems, network topologies and protocols. The department decided a system would have to be developed to allow everyone to share information regardless of which type of network or computer system they had. The Advanced Research Project Agency (ARPA) was given the task of designing and implementing such a system. ARPA formed an alliance with several universities and computer manufacturers to develop a set of protocols which could be platform independent. The result was a network which is the basis for the Internet today and the TCP/IP suite of protocols. The number of computers connected to the Internet has increased at a phenomenal rate due to explosion of the World Wide Web (WWW) [Tackett *et al* 1998].

Towards the late 1970s the Internet Protocol suite was completed. TCP/IP was later included with the Berkeley Software Distribution (BSD) of UNIX. These protocols form the basis for the World Wide Web. Internet protocols specifications are documented in technical reports called "Request for Comments" or RFCs. These documents are published by people who are involved in development of the protocols and are revised when required. Updated versions of the documents are published in new RFCs [Cisco *et al* 2000]. Today the Internet is used in almost all educational institutions, corporations, research labs and businesses. It has become an invaluable communications tool which allows users to send messages such as Electronic Mail (Email) around the world is seconds. The Web can allow users to retrieve information from a website almost instantly.

2.3 Network Hardware and Software

"*A network is a number of computers connected together to share information and hardware*" [TekMom 2002]. The information can be anything from files such as word documents to emails, the hardware could be typically something like a printer or a scanner. There are two types of communications networks; connection-oriented (circuit switched) and connectionless (packet switched). An example of a connection-oriented network is that of the telephone system. When a user makes a call they have a dedicated line to the other end, their voice is continuously sampled, with the 64kbps data stream continuously delivered to the other end. The circuit switched connection can be made over many different telephone companies' equipment. An analogy of a connectionless network is that of the postal system. A letter (packet) is sent from one location to another, it travels

through various postal sorting offices until it reaches its destination. Unlike a connection-oriented system the time taken for the letter or packet to be delivered isn't fixed. In fact the letter may not be delivered at all. In both cases an address is used to indicate where to deliver the message; the phone system uses a phone number and the postal system uses the postal address.

Packet switched networks can be divided into two broad categories; Wide Area Networks (WANs) and Local Area Networks (LANs). LANs provide the highest speed connection and lowest delay. LAN technologies used in most company networks such as Ethernet can operate at speeds between 10Mbps and 1Gbps. The delay between nodes can be as little as a fraction of a millisecond. A computer is connected to a LAN using a Network Interface Card (NIC) while connecting to a WAN requires connecting the computer to a packet switch. The switch can be connected to many other switches in the WAN, hence the reduced speed and increased delay. Figure 2.1 shows the topology for a simple LAN; all computers are connected on the same physical network and can communicate with each other directly. WANs provide communications over longer distances, typically across continents. They operate at lower speeds (1.5Mbps to 155Mbps) than local area networks and have greater delays (few milliseconds to several tenths of a second). Figure 2.2 shows a topology for a simple WAN; two separate physical networks (LANs) are linked together using a router.

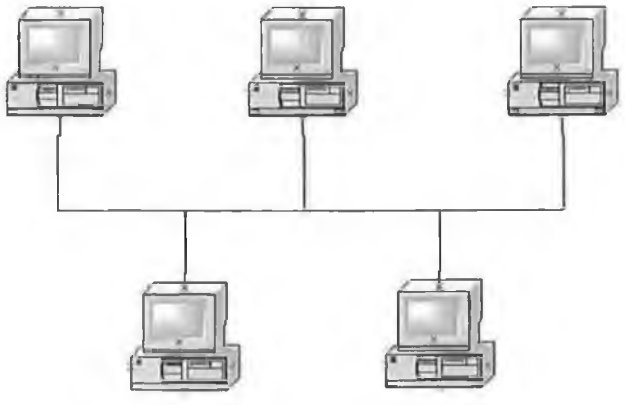

Figure 2.1 A Local Area Network [Tanenbaum 2003].

If any computer on subnet 1 wants to communicate with subnet 2 it must do so through the router. The same applies for computers on subnet 2 wanting to communicate to computer on subnet 1. The most common example of a WAN is the Internet, which is made up of millions of computers and many thousands of routers [Comer 2000].

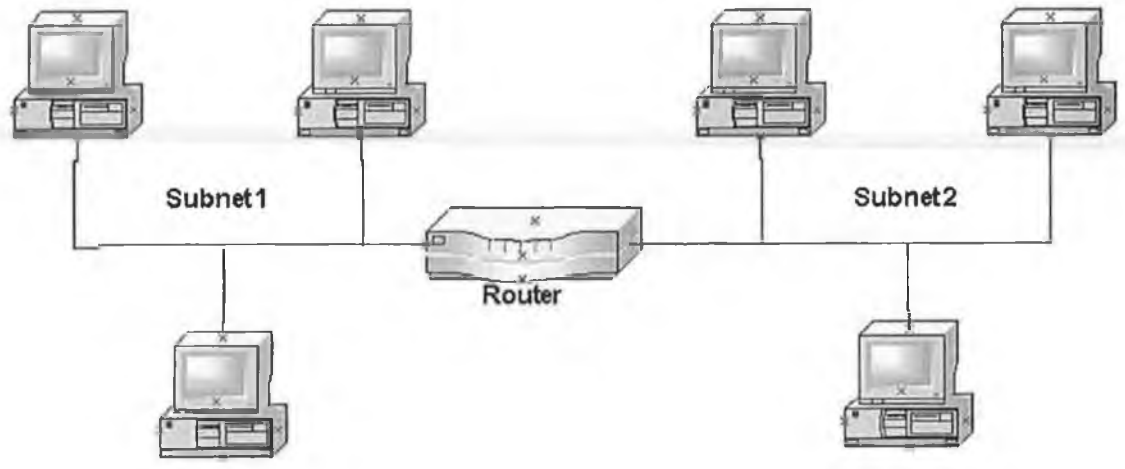

Figure 2.2 A Wide Area Network [Tanenbaum 2003].

Many different network hardware technologies exist such as Ethernet, Token Ring and Fiber Distributed Data Interface (FDDI). As well as the network hardware, there is also the requirement in a network for software tasks on each host. This software comes in the form of protocol stacks. A protocol is an agreement between communicating parties on how communications is to proceed. It is analogous to a language; in order for two computers to communicate they must speak the same language. A protocol stack is a group of protocols layered on top of one another. The protocol in layer N provides functions to the protocol in the upper layer N + 1. Layering protocols like this simplify the design of networking systems. The purpose of the network hardware is to transfer packets. Packets are short messages which are sent and received by network hardware. Note packets are sometimes referred to as datagrams in the context of IP or as segments in the context of TCP [Tanenbaum 2003].

2.4 The Open Systems Interconnect Model

As explained in the previous section, in the late 1960s many different types of computers existed, each with different operating systems and network technologies. Hence communicating between different systems became a problem. In 1977, the International Organization for Standardization (ISO) created a subcommittee to develop data communications standards to promote multi-vendor interoperability. The result is the Open Systems Interconnection (OSI) model. The OSI model should not be confused with the development of TCP/IP, the model doesn't specify any protocol or communication standard. The model instead provides a framework of the functions to be performed; it doesn't describe how the functions are to be performed. The ISO subcommittee took the

"Divide and Conquer" approach. This breaks down a communications process into smaller subtasks which can be more easily solved [Tackett *et al* 1998].

All layers are independent of each other. Each layer provides functions to the layer above it. For example the physical layer provides services to the data-link layer. It should be noted however that not all communications processes are split into seven sub-tasks, i.e. there is not a one-to-one correspondence between the OSI Reference Model and the Internet Protocol Suite. Figure 2.3 shows the OSI model and its correspondence to TCP/IP. In some cases the functions in the lower layer which are software tasks have been replaced by optimized hardware. One of the disadvantages of the seven layer model is that each layer adds overhead in the form of the header. If all seven layers have headers, less than 15% of the message is source data. The functions of the different layers in the hierarchy are given on the below.

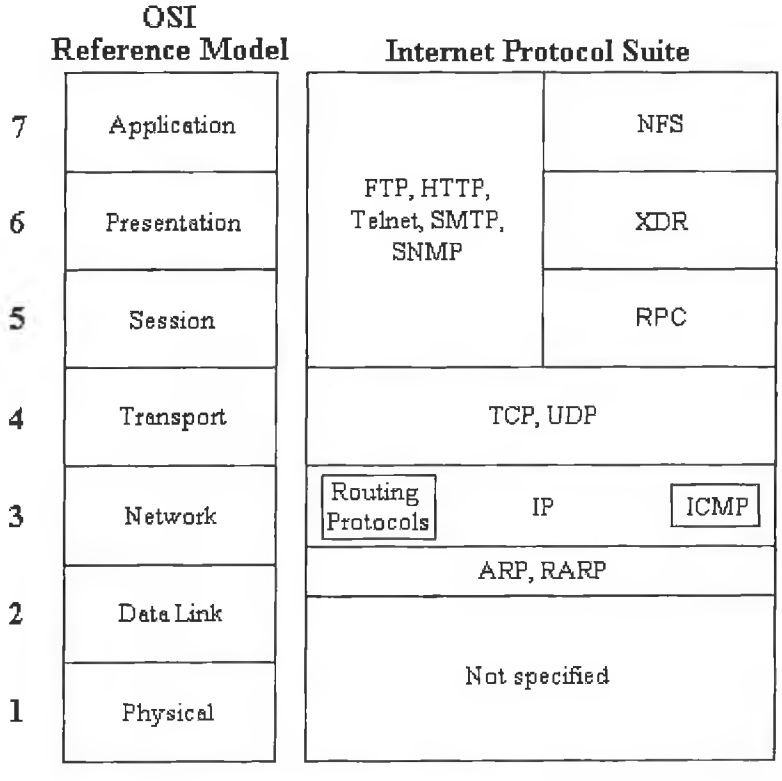

Figure 2.3 OSI Model and TCP/IP suite [Cisco *et al* 2000].

1. *Physical Layer.* This is the lowest layer in the hierarchy. This layer defines physical, electrical, functional and procedural standards for accessing the data network.
2. *Data-link Layer.* The data link layer provides functions to activate, maintain and deactivate the link. Framing and error correction and detection are also handled.
3. *Network Layer.* The network later defines the mechanism used for routing the message through the network.

4. *Transport Layer.* The transport layer acts as an interface between the network layer and session layers. Layers above the transport layer are independent of network technology. The layer controls the end-to-end integrity of the message which includes routing, segmenting and error recovery.
5. *Session Layer.* The session layer is responsible for network availability (i.e. buffer storage and processor capacity). A session is a temporary condition which exists when data is in the process of being transferred. Session responsibilities include network log-on, and log-off and user authentication.
6. *Presentation Layer.* The presentation layer is responsible for data conversion and ensures data is in a universal format. Presentation functions include data file formatting, encoding (ASCII, EBCDIC, etc.), compression, encrypting and decrypting.
7. *Application Layer.* The application layer is the highest in the OSI hierarchy and is the interface between the user application and the network communication process. Note the application layer shouldn't be confused with the user application program which is executed on a computer. The application program is an implementation of the application layer.

Layers 3, 4, 5, 6, 7 allow for hosts to communicate directly (peer-to-peer), the lower layers are concerned with the transfer of data (at bit level) from one host to another [Tomasi 1998].

2.5 The Physical Layer

The physical layer, the lowest in the OSI hierarchy defines mechanical, electrical and timing interfaces to the network. This section looks at the different types of physical layer media as well as some modulation techniques used to transfer data over the media [Tanenbaum 2003].

2.5.1 Media

The purpose of the physical layer is to transport a raw bit stream from one host to another. This is done over a medium which in most cases uses a band in the electromagnetic spectrum. Media can be divided into two categories: guided media such as copper wire and fiber optics, and unguided media such as radio and lasers. The most commons types of guided media are as follows:

- Magnetic media such as floppy disks, hard disks and tape drives.

- Twisted pair such as the copper wire used in the phone system.
- Coaxial cable, used for connecting televisions and radio equipment to antennae.
- Fiber optics, used for high speed data transfer over long distances.

The most common types of unguided media are as follows.

- Radio, transmission on frequencies in the electromagnetic spectrum below 1GHz such as broadcast television and radio and data communications through free space.
- Microwave, used in mobile phones, microwave television and satellite communications.
- Light waves, infra-red and visible light can be used to transfer data through free space. A common example is a television remote control [Tanenbaum 2003].

Wireless communications uses various bands of the electromagnetic spectrum to send data across a medium. The most common bands are radio waves, microwaves and infrared. Different bands are suited to different situations, for example infrared it typically used for remote controls or IrDA ports on laptops where there is line of sight between the sender and receiver. Radio waves are used for long distances and are not as susceptible to obstacles as infrared. Many SCADA systems use radio links in situations where RTUs are placed in remote locations and wired connectivity is unsuitable. Such radio links generally don't use IP and instead use a propriety protocol for controlling and monitoring the device.

2.5.2 Modulation Techniques

The two most common basic types of modulation used in everyday applications such as radio and television are Amplitude Modulation (AM) and Frequency Modulation (FM). Amplitude Modulation combines two signals; a radio frequency (RF) carrier and the baseband signal (information). The output is a radio signal with a frequency centered on that of the carrier but whose amplitude changes in proportion to the baseband signal. Frequency Modulation is a similar technique that varies a carrier frequency according to the baseband signal.

Both techniques can be easily adapted for digital modulation. Instead of using a sinusoidal baseband signal, a binary pulse stream is used instead. Frequency Shift Keying (FSK) uses constant amplitude angle modulation similar to conventional FM except the modulation signal is a stream of 1s and 0s. Binary Phase Shift Keying (BPSK) is similar to FSK except instead of changing the frequency according the baseband digital signal, the phase is changed instead. In order to increase the amount of data that can be sent through a given amount of bandwidth, the number of angles that the carrier can be modulated to can be

increased. If four phases are used then two bits can be sent at a time and this is known as Quaternary Phase Shift Keying (QPSK). Quadrature Amplitude Modulation (QAM) is a digital modulation technique where the phase and amplitude can take multiple values. These modulation methods allow several bits to be transferred as one symbol (baud) [Tomasi 1998].

2.5.3 Public Switched Telephone Network

The Public Switched Telephone Network (PSTN) was originally designed for the transmission of voice over copper wires. It allows circuit switched calls and is now commonly used for data communication using a modem and can be used to connect a computer in a remote location to the Internet. PSTN using a modem gives a maximum speed of 56 kbps; this makes it only suitable for home or small office use. Integrated Service Digital Network (ISDN) is a double line digital telephone system that can use the same type copper wires as PSTN and offers data transfer speeds of up to 128 kbps. Asymmetric Digital Subscriber Line (ADSL) can also be used with PSTN lines and allows transfer speeds of up to 1 Mbps. The transfer speeds however are asymmetric; the download speed is typically more than the upload speed. For example the download speed could be 512kbps and the upload speed could be 128 kbps. This is quite acceptable for broadband as most Internet users download data as apposed to upload. ISDN and ADSL have limitations over PSTN lines, the installation site must be within a certain distance of the local exchange and the copper wire must meet certain quality criteria.

2.5.4 Mobile Phone Network

The most common mobile phone technology in use today is Global System for Mobile Communications (GSM). GSM allows secure voice and data communications over radio waves in three different bands of the electromagnetic spectrum. GSM can also be used for linking a computer to a network, like PSTN however it provides low communications speeds [Tanenbaum 2003].

2.6 The Data Link Layer

The data link layer provides functions for the sending a receiving of frames to the upper network layer. The data link layer allows the network layer to send packets to any other machine on the network. It also passes packets received up to the network layer. The functions provided to the network layer include:

- Provide well defined service interface to the network layer.
- Deal with transmission errors.
- Control the flow of data, so receiver buffers aren't overrun with data.

The data link layer is responsible for framing, error control and flow control. It is possible that these may also be provided by the upper layer protocols. The data link takes the data payload from the upper network layer and encapsulates it in a frame. A frame is made up of a bit stream, figure 2.4 shows a frame for a simple data link protocol.

Figure 2.4 A Simple Data Link frame [Tanenbaum 2003].

The flags are used to indicate the start and end of the frame, the header contains control and addressing information such as the destination machine and the payload is the network layer data. The trailer contains a checksum for error control; this is generated by an arithmetic calculation on the frame, e.g. a simple checksum could be the sum of the value of all bytes in the frame. When a frame is received, the checksum is calculated and compared with that in the trailer, if they are different, the frame is either discarded or the receiver may request the sender to retransmit it. To control the amount of data a sender may transmit to a receiver, a flow control mechanism is used. The simplest flow control is having the receiver sending a simple message to say either "*you can transmit N bytes now*" or "*you cannot transmit now*". This is known as software flow control, the other type is hardware flow control which in its simplest form uses an extra line (wire) which when it is logic 1 allows transmission and when it is logic 0 inhibits transmission [Tanenbaum 2003]. Figure 2.5 shows how information is encapsulated into a data link layer frame.

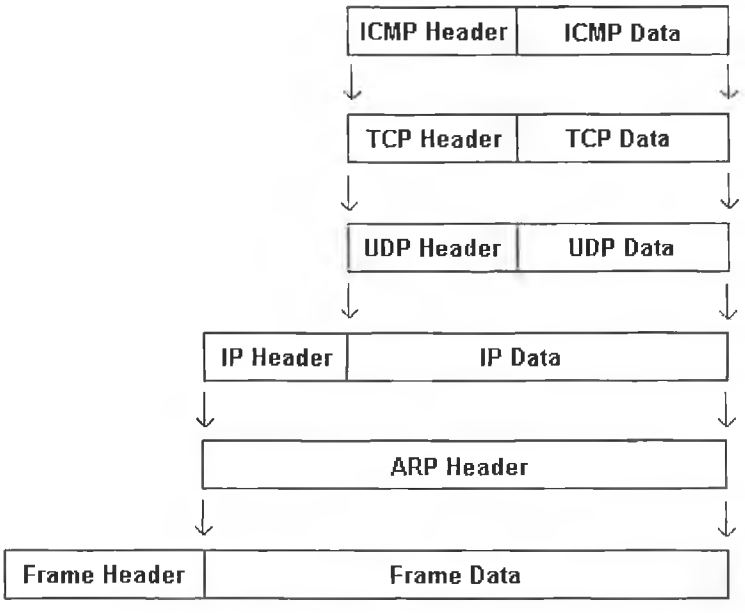

Figure 2.5 TCP/IP Protocol encapsulation [Hall 2000].

The data field of the frame for example could be an Address Resolution Protocol header or an IP packet. The IP packet itself also has a header and data. Within the IP packet data is another header and data for the protocol in the next upper layer. The Type field in the frame identifies the type of frame data. In the case of Ethernet, 0x0800 indicates an IP packet while 0x0806 indicates an ARP packet. The same principle applies to Token Ring, FDDI, and (PPP) frames [Hall 2000].

2.6.1 Point to Point Protocol

Point-to-Point Protocol is a common data link layer protocol used for computers in remote locations that don't have a direct connection to a LAN or WAN such as home Internet users. Point-to-Point Protocol (PPP) is commonly used to connect the remote computer to a network over a phone line (serial communications link). Typically it uses lower speeds than that found in LANs. Home Internet users use PPP over a modem to connect through an Internet Service Provider (ISP). PPP provides a standard method for transporting multi-protocol datagrams over point-to-point links. It has the following characteristics:

- It allows multi-protocol datagrams to be encapsulated.
- A Link Control Protocol (LCP) is used for establishing, configuring and terminating the link.
- Authentication of the user who is dialing in. Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP) are the two most common authentication protocols.
- A set of Network Control Protocols (NCP) for configuring the upper layer protocol to be transported over the link such as IP. Internet Protocol Configuration Protocol (IPCP) is used to configure options such as IP address and Domain Name System (DNS) servers [Simpson 1994].

Figure 2.6 gives common PPP frame types and a description of the fields is given on the next page.

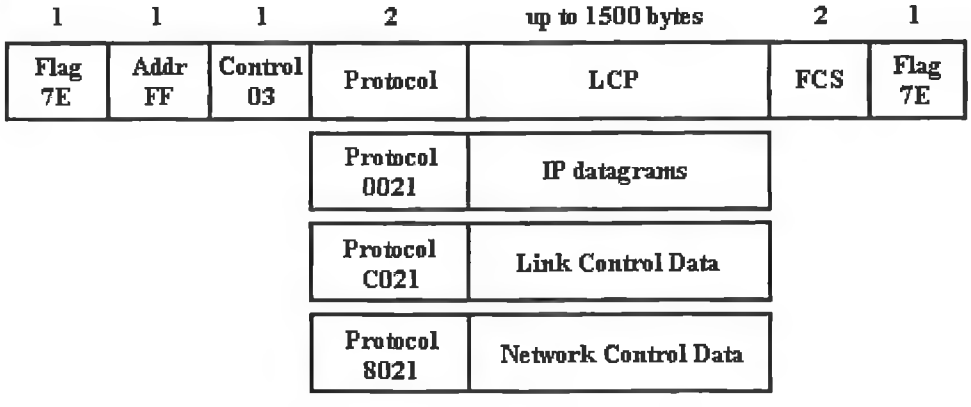

Figure 2.6 Point to Point Protocol frame types [Simpson 1994].

- Flag sequence (01111110 binary). The flag sequence is used for synchronization and indicates the beginning and end of the frame. To ensure the flag sequence does not appear within the frame, a form of byte stuffing is used. There only needs to be one flag sequence between consecutive frames.
- The address identifies the station that should receive the frame. PPP does not use this field and it is always 0xFF.
- The control field for PPP is always 0x03.
- The protocol field is one or two bytes depending on whether compression is used, and its value identifies the type datagram encapsulated in the information field of the packet (such as IP or LCP, etc.).
- The data field is the same as the Ethernet data and contains the upper layer protocol header and data.
- The FCS is a cyclic redundancy check on the frame to verify its integrity and is carried out over all fields except the flag sequences [Simpson 1993].

2.7 The Medium Access Control Sub-Layer

Networks can be divided into categories: point-to-point connections and broadcast channels. The previous section dealt with the most common point-to-point network: Point-to-Point Protocol (PPP), this section looks at common broadcast type networks [Tanenbaum 2003].

2.7.1 Multiple Access Techniques

In order for multiple stations to share the electromagnetic spectrum or the bandwidth on a copper wire, several techniques are used which slit the bandwidth between users. These techniques are discussed in the following sections.

Time Division Multiple Access (TDMA) is a technique in which all stations use a single channel. Each station is allocated a time slot and can only transmit on that slot. Figure 2.7 shows a TDMA system. Station 1 only transmits on Slot 1, etc [IEC 2003].

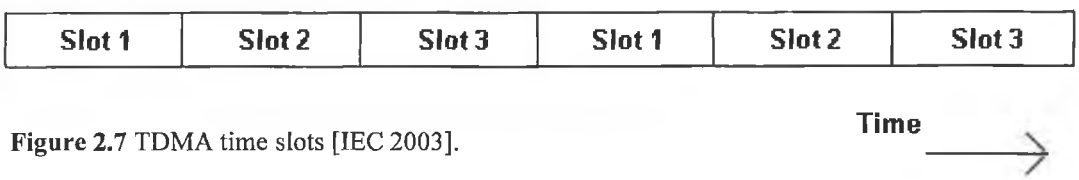

Figure 2.7 TDMA time slots [IEC 2003].

Frequency Division Multiple Access (FDMA) splits the RF spectrum in to channels. Each pair of stations is allocated a channel. Commercial FM radio is a typical example of

TDMA; various radio stations operate on different frequencies [Tomasi 1998]. *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD) is a system is which each station continuously listens for traffic. When a station wants to send a frame, it can only do so when the network is quiet. If a station which is sending a frame detects that another station is also sending a frame, it must stop a wait a certain amount of time determined by a back-off algorithm [Cisco *et al* 2000]. *Code Division Multiple Access* (CDMA) uses *"frequency hopping"* in which the carrier signal changes frequency at random. Hence only the receiver that knows the pattern of the frequency hopping can decode the bit stream. Because of the continuously changing frequency, CDMA provides better security. The result is a signal which is spread across the radio spectrum and looks like white noise and hence causes less interference to other radio services or adjacent channels. Because the power density is spread across the spectrum higher bit rates can be achieved [Yen *et al* 2000].

2.7.2 Ethernet

Some of the most common LAN technologies are Ethernet, Token Ring and Fiber Distributed Data Interface (FDDI). Ethernet is the most popular for most companies as it is cheap and relatively easy to configure and setup. It offers speeds between 10Mbps and 1000Mbps (1Gbps). Ethernet uses Carrier Sense Multiple Access/Collision Detection (CSMA/CD) to allow multiple stations to send frames on the network. Each station continuously listens for traffic on the network, a station which wants to send a frame can do so when the network is quiet. If a station which is sending a frame detects that another station is also sending a frame, it must stop and wait a certain amount of time determined by a back-off algorithm. The Basic IEEE 802.3 Media Access Control (MAC) data frame format in shown in figure 2.8. The number of bytes contained in each field in given in the upper part of the diagram.

Ethernet						
	1	6	6	2	46-1500	4
Preamble	Start of frame delimiter	Destination Address	Source Address	Type	Data	Frame Check Sequence

Figure 2.8 Ethernet frame format [Cisco *et al* 2000].

A description of the fields in an Ethernet frame as shown in figure 2.8 is given below:

- The preamble (7 bytes) is used for synchronizing the receiver and to indicate a frame is about to be received.
- The Start-of-frame delimiter (1 byte) is used by the receiver to know when the destination address is the next block to be received.
- The destination (6 bytes) address indicates which station(s) should receive the frame. It is possible depending on the address that the frame can be received by several stations. A broadcast (all 1s) frame is received by all stations while a multicast frame is only received by stations which have joined a multicast group.
- The source address (6 bytes) identifies the sending station.
- The Length/Type field (4 bytes) depending on the frame formation indicates either the length of the frame or the protocol type in the data field such as IP.
- The data field (46 – 1500 bytes) contains the upper layer information such as an IP packet. If the data field is less than 46 bytes padding is added.
- The frame check sequence (4 bytes) is a cyclic redundancy check on the frame excluding the preamble and start-of-frame delimiter and is used to check to integrity of the frame [Cisco *et al* 2000].

2.7.3 IP over radio

The wireless use of TCP/IP allows computers and devices to communicate with each other and other hosts over the Internet using radio or infrared based technologies. In recent years standards like IEEE 802.11 and Bluetooth allow computers and devices to interact without the use of any cables. With the massive growth of wireless networks, wireless providers still have many obstacles to overcome before true wireless Internet becomes a reality. However the use of IP over radio links poses many problems. Such wireless technologies are still under review and being further developed, the speed and distance at which they can operate is being improved. One problem with implementing wireless devices has been cost. The development costs of such technologies along with the high cost of the end user devices have impeded the deployment of wireless data services. Broadband access using radio over long distances isn't always practical as it requires a lot of bandwidth. However slow speed radio links may not be suitable for IP due to the headers imposed at several layers of the TCP/IP stack. Also the fact that a host or device requires an IP address which is associated with its physical location causes problems in routing packets to its destination if the host or device moves. Other concerns with the use of radio include RF interference to other radio technologies, health risks and network security [Makki *et al* 2002].

2.7.4 IEEE 802.11

IEEE 802.11 is a standard developed for wireless local area networks and is packet based. It implements the data link and physical layers. The cost of such technology has dropped in recent years to the point where it can be used in place of traditional wired LANs in the office. IEEE 802.11 uses Distributed Foundation Wireless Medium Access Control (DFWMAC) as it's a MAC (Media Access Control layer) protocol. It is medium independent and allows communication between wired and wireless networks. However the physical layer is not independent and there are different PHY (physical layer) specifications for different frequency bands used such as radio and infrared along with modulation type. DFWMAC uses Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) With Acknowledgement which is similar to CSMA/CD used by Ethernet in that if the medium is free transmission begins immediately. Unlike Ethernet there is no way of detecting a collision due to the nature of radio, so DFWMAC waits for an exponential time called back-off it the medium is busy. DFWMAC also expects an acknowledgement from the destination device or devices and if it is not received the frame is retransmitted after the back-off time. This gives IEEE 802.11 robustness for managing the possible loss of data [Moreira *et al* 1997].

There are various 802.11 standards with different link speeds and distances. 802.11b allows operation up to 11Mbps and a distance of up to 1500 feet. The physical layer uses frequency hopping or direct sequence spread spectrum (DSSS) to transfer frames. TCP performs well on 802.11; however it doesn't perform as well on low bandwidth wireless networks compared to wired networks, this mainly due to the large percentage overhead caused at low communication speeds. To make TCP/IP operate efficiently over such links some modifications have to be made to TCP such as header compression and timeout values appropriate to the speed of the link [Avancha *et al* 2002]. Wireless LAN standards like 802.11 do provide higher bit rates, however this isn't necessary in many SCADA applications as typically a device is polled every few seconds which can be done using slower bit rates.

2.7.5 Bluetooth

Bluetooth is a low cost, low range radio based protocol used to link electronic devices. It operates in the unlicensed Industrial, Scientific and Medical (ISM) band at 2.4 GHz. It has a maximum range of between 10 and 100 meters. Bluetooth uses Gaussian Frequency Shift Keying (GFSK) radio modulation. Logic 1 is represented by a positive frequency deviation and logic 0 is represented by a negative frequency deviation. Frequency hopping

is used to avoid fading and interference. Bluetooth has a symbol rate of 1MBps and a normal time slot length of 625μs. Information is exchanged through packets using a Time-Division Duplex (TDD) which can take from one to five time slots. Every packet is sent on a different frequency hop. Bluetooth supports both packet and circuit switching which allows voice as well as data to be transferred. The functional blocks of Bluetooth are given in figure 2.9.

Figure 2.9 Functional blocks in a Bluetooth system [Bluetooth 2001].

Figure 2.10 shows the different Bluetooth topologies.

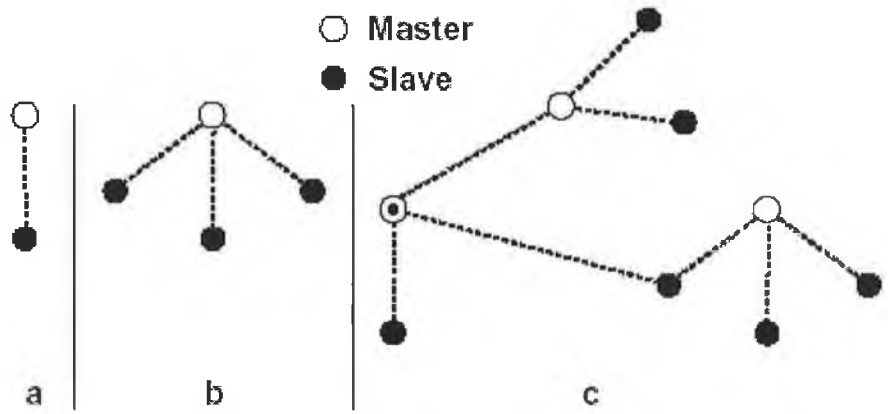

Figure 2.10 Bluetooth topologies [Bluetooth 2001].

Bluetooth allows point-to-point (A) or point-to-multipoint (B) communications. In the case of multipoint, several units can share the same channel. Two or more units using the same channel form a *piconet* (B). This is master/slave type configuration. Up to seven slaves can be active on a piconet. Multiple piconets overlapping form a *scatternet* (C). In a scatternet a slave or master in one piconet can be a slave in another piconet [Bluetooth 2001]. Bluetooth has the advantage of being relatively inexpensive. Manufacturers believe it will add around $5 to the cost of a product by the end of 2002. Secondly as it radio based, it requires no cables. Thirdly it requires little user interaction; Bluetooth devices can detect and communicate with each other when they come in range [Erasala *et al* 2002]. Bluetooth has a limited range less than 100 meters, this would make it unsuitable for many SCADA applications which can require radio links over very long distances.

2.7.6 General Packet Radio Service

The General Packet Radio Service (GPRS) is a standard derived by the European Telecommunications Standards Institute (ETSI) for packet data in GSM systems. It allows GSM operators to provide subscribers with a packet switched and hence resource efficient, access to external Internet protocol based networks [Lindemann *et al* 2002].

Global system for mobile communications (GSM) has been widely adopted in over 100 countries and offers data and voice services. GSM is typically used for mobile phones communications. The basic GSM service offers data rates up to a maximum of 14.4 kbps. This is in the form of a circuit switched data call. To increase the data rate High Speed Circuit Switched Data (HSCSD) allows multiple time slots can be allocated for a subscriber giving a maximum of 43.2 kbps. Such circuit switched techniques are unsuitable to many burst data applications such as Web browsing where a user may only utilize the channel when they visit a web page and the connection is idle when they are looking at it. Circuit switched systems are unattractive for both the user and GSM service provider. The user doesn't want to have to pay for keeping a channel active even though they only use it a fraction of the time, hence an always-on connection isn't economically viable. Similarly the service provider doesn't want to have to provide expensive radio resources which may be idle most of the time.

GPRS is a packet based technology which requires modifications to the GSM network. It uses the same frequency bands, TDMA frame structure and the same modulation technique. Since it is packet switched it uses bandwidth only when needed and using eight time slots can provide a bit rate of up to 170 kbps. However operators may limit the number of time slots allocated to a user and hence reduce the maximum bit rate [Ghribi *et al* 2000].

As the GSM network is widely available and GPRS allows an always-on connection, it is well suited to SCADA applications which require remote links. It's fast, inexpensive and as it can be always on, it doesn't have any connection setup delays like normal modems.

2.8 The Network Layer

The network layer is used to transfer packets or datagrams from the source to the destination nodes. These packets are passed down to the network later for transmission by the transport layer. The following section discusses Internet Protocol (IP), which is the network layer for the TCP/IP suite of protocols [Tanenbaum 2003].

2.8.1 Internet Protocol (IP)

The Internet protocols are the world's most popular open-system because they are used to communicate information across different types of interconnected networks. The protocols are suited to both Local Area Networks (LAN) and Wide Area Networks (WAN). The Internet protocol is a network-layer (Layer 3) protocol which contains addressing information and is designed for use in interconnected systems of packet-switched computer communication networks. The Internet protocol implements three basic functions: addressing, routing and fragmentation. The IP header contains fields which are used to route the packet to its destination and reassemble a fragmented datagram when it passes though a network which only allows small packets [Cisco *et al* 2000].

The IP header length is always a double word (32-bit) multiple. The minimum and typical IP header is 20 bytes, though if options are included it can be up to 60 bytes. The frame format for of an IPv4 header is given in figure 2.11 and a description of the fields are given below. Note all fields are in network byte order (Big Endean):

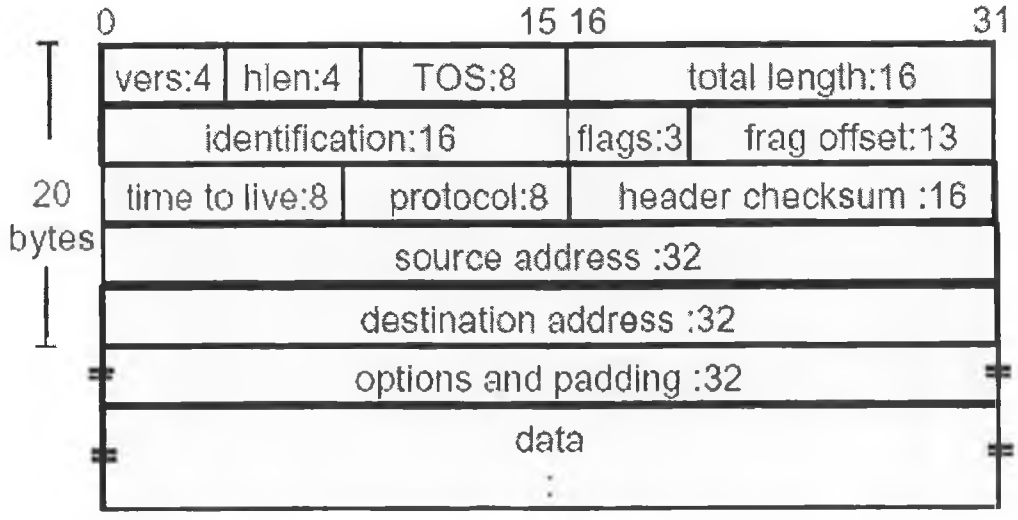

Figure 2.11 Internet Protocol header format [Postel2 1981].

- VERS: version number, always 4 for IPv4.
- HLEN: header length, in 32-bit words, normally 5.
- TOS: how the datagram should be handled.
- Total length of the packet (header + data)
- Identification, Flags, Fragment Offset: provides fragmentation of datagrams to allow different MTUs in the internetwork.
- TTL: Time-To-Live, maximum lifetime of datagram.
- Protocol: upper-layer (Layer 4) protocol sending the datagram.
- Header checksum: integrity check on the header.

- Source address and destination address: 32-bit IP addresses.
- Options: network testing, debugging, security, and other options [Postel2 1981].

2.8.2 Internet Protocol Addressing

In order for computers from different network technologies and topologies to communicate, a system of addressing which is independent of the underlying network was devised. This addressing is known as IP addresses. Each host is assigned a unique 32-bit Internet address (number) that is used for all communications for that host. The number is chosen so as to make routing between networks efficient. The number is of the same type as the destination or source address in the IP header. IP addresses are normally expressed in Dotted Decimal Notation, e.g. "10.1.1.1". IP addresses fall into one of five different classes: A, B, C, D or E. The class depends on the numerical value of an IP address. Figure 2.12 shows how IP addresses are spilt into classes depending on the upper five bits.

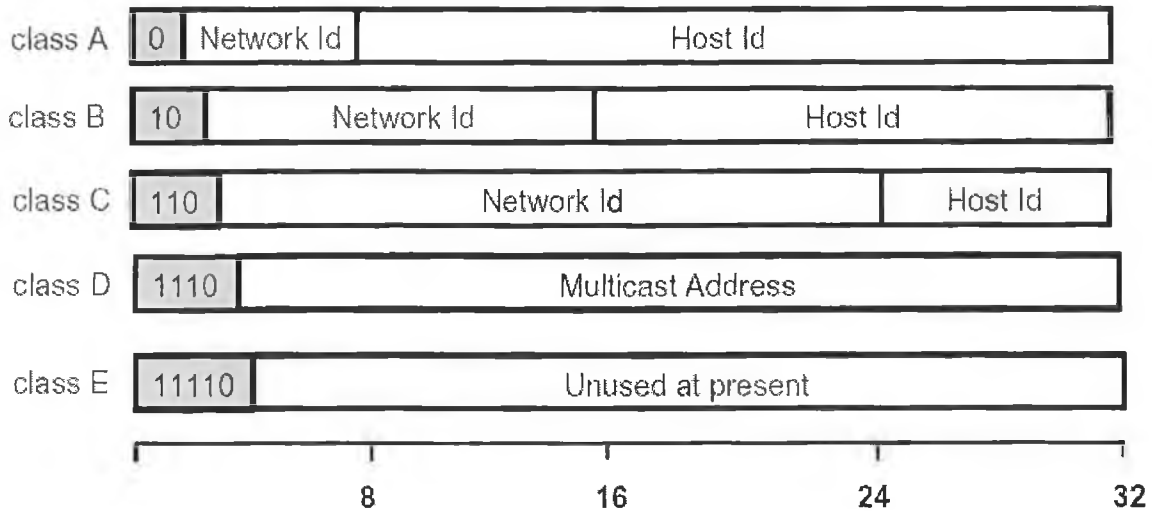

Figure 2.12 The five classes of IP address [Comer 2000].

Each address is a pair (netid and hostid) where netid identifies a network and hostid identifies a host on that network. In the original addressing scheme know as "*classful*" the boundaries between netid and hostid were fixed as above for classes A, B and C. However in practice the boundaries are not uniform throughout the Internet. This results in classless routing. Multicast addresses (D) allow a host to send a datagram to multiple hosts which are part of a "multicast group", similar to a broadcast which is sent to all hosts. Hence the five IP address ranges are:

Class	Lowest Address	Highest Address
A	**1.0.0.0**	**126.0.0.0**
B	**128.0.0.0**	**191.255.0.0**

C	**192.0.0.0**	**223.255.255.0**
D	**224.0.0.0**	**239.255.255.255**
E	**240.0.0.0**	**255.255.255.254**

From looking at the above, not all addresses have been assigned to classes. In particular the network "127.0.0.0" and the loopback address "127.0.0.1" are used to specify the host itself allowing a computer to communicate with itself without packets being sent on any network [Comer 2000].

2.8.3 Internet Protocol Fragmentation

Each of the different network topologies have different Maximum Transmit Unit (MTU) sizes, which represent the maximum amount of data that can be passed in a single frame. On Ethernet networks this is typically 1500 bytes while 16MB/s Token Ring has a default MTU size of 17,914 bytes. Whenever a datagram has to be sent across a network which has a smaller MTU than the size of the datagram itself, fragmentation must occur. Usually on local networks fragmentation is not required. In TCP for example, consideration is given to MTU size of the lower layer. TCP allows a Maximum Segment Size to be specified as an option. The segment size is typically set the same as the MTU of underlying network eliminating fragmentation of TCP segments on a local network. UDP applications however, such as Network File Service (NFS) often send more data than can fit within the local network's MTU. Internet Control Message Protocol (ICMP) messages such as 'ping' can also require fragmentation if the user specifies a large amount of data [Hall 2000].

The fragmentation process involves the use of three fields in the IP header: Identification, Flags and Fragment Offset. Each datagram has a different identification number. If the 'Don't Fragment' flag is set the datagram cannot be fragmented and must be discarded. An ICMP "Destination Unreachable Message, fragmentation needed and DF set" message is sent back to the originating host to indicate that the datagram cannot be sent [Postel3 1981]. When an IP module receives part of a fragmented datagram it allocates a storage buffer. The data is copied into the storage buffer at location (fragment offset in the IP header * 8). The data is copied in a similar way for each additional segment received of the datagram with the same identification number. The datagram is complete and passed to the upper layer when all fragments from zero are received including one with the Last Fragment flag set. If two fragments with the same identification and fragment offset are received the newer one received is always used. If all of the datagram isn't received within a finite time the partially reassembled datagram is discarded and an ICMP "Time Exceeded

Message, fragment reassembly time exceeded" message is sent back to the originating host to indicate that the datagram cannot be reassembled [Postel2 1981].

2.8.4 Static IP Routing

An *internetwork* is a collection of individual networks, connected by intermediate networking devices, that functions as a single large network. The networks can have different topologies such as Ethernet, FDDI and Token Ring as shown in figure 2.13. Routing is the movement of information from its source to destination across an internetwork. The information will travel through at least one intermediate node. Routing occurs at layer three in the OSI.

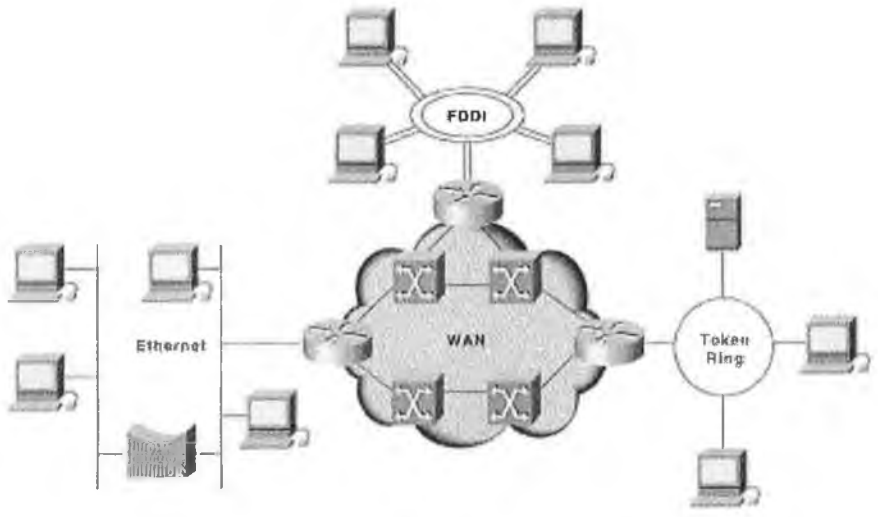

Figure 2.13 An inter-network [Cisco *et al* 2000].

Metrics are used to determine the best path for packets to travel. Typical metrics include Path length, Reliability, Delay, Bandwidth, Load and Communication cost. The routing table on a Windows 2000 PC is shown in figure 2.14.

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	10.1.1.1	10.1.1.3	1
10.1.0.0	255.255.0.0	10.1.1.3	10.1.1.3	1
10.1.1.3	255.255.255.255	127.0.0.1	127.0.0.1	1
10.255.255.255	255.255.255.255	10.1.1.3	10.1.1.3	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
224.0.0.0	224.0.0.0	10.1.1.3	10.1.1.3	1
255.255.255.255	255.255.255.255	10.1.1.3	10.1.1.3	1

Figure 2.14 A routing table from a Windows 2000 workstation.

The *Network Destination* identifies a network N.

The *Netmask* specifies the size of the network.

The *Gateway* is the next hop for routing a datagram.

The *Interface* identifies the Network Interface Card to send the datagram on.

The *Metric* indicates the cost of sending the datagram.

The algorithm used to route datagrams is given below. The network prefix of a datagram is computed by logically ANDing the destination address with the net mask.

Extract destination IP address, D, from the datagram and compute the network prefix, N.

If N matches any directly connected network address

Deliver datagram to destination D over that network.

Else if the table contains a host specific route for D

Send datagram to next hop specified in the table.

Else if the table contains a route for network N

Send datagram to next hop specified in table.

Else if the table contains a default route

Send datagram to the default router specified (0.0.0.0) in the table.

Else

Declare a routing error.

2.8.5 Dynamic IP Routing

Static routing, which was discussed in section 2.8.4 generates the routing table from the interface configuration. Routes can also be manually added. This is typically used in a desktop PC on an office LAN where there is one router and there is no reason for the routing table to change. However in networks with multiple routers, a dynamic form of routing is required to recover from the problem of one of the routers crashing. Two common dynamic routing protocols that are used are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). Both are Interior Gateway Protocols (IGP) which means routing information is only distributed between routers in a single autonomous system.

Routing Information Protocol (RIP) uses a distance vector algorithm. The hop count to a destination network is used as the metric for a route. Each router periodically sends a copy of its routing table. Figure 2.15 shows four networks: W, X, Y and Z. The routing table for each uses the hop count from source to destination as the metric, e.g. for R1 to get to network Z, it has to go through R2 and R3 (2 hops). The routing tables are generated by periodic UDP broadcasts. When a router is added or fails, it takes a while for its routes to propagate through the system. E.g. in figure 2.16 when a route changes on the network adjacent to R3, each router from R3 to R1 must re-compute its routing table from the periodic broadcasts.

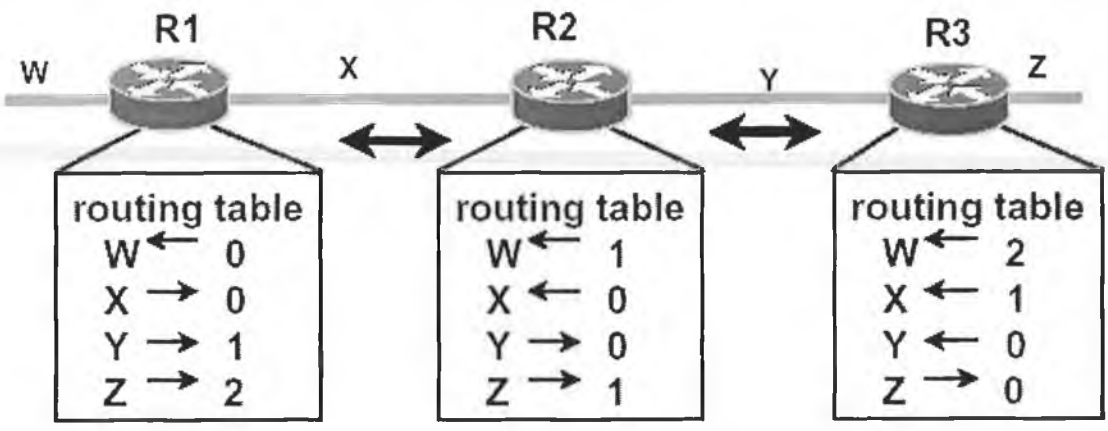

Figure 2.15 Routing Information Protocol network [Hedrick 1988].

Hence there is a slow convergence from when a route fails to when the entire system will be aware of the failure. RIP has safeguards to ensure the system cannot become unstable.

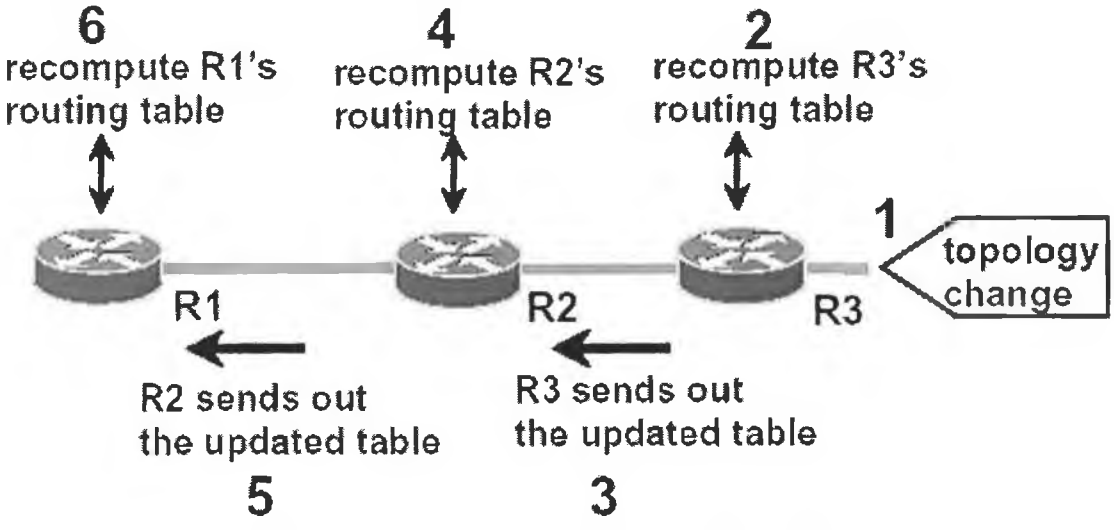

Figure 2.16 Routing Information Protocol network topology change [Hedrick 1988].

For example in figure 2.17, all routes have a metric of 1. Take the case where the route from B to D fails. B will detect this as it doesn't hear anything from D and will update its route table accordingly. However C still thinks it can route to D through B and will advertise this (it hasn't got the broadcast from B yet). Hence B will receive a route to D through itself which is invalid. B will then broadcast this route back to C and so on. The result is both routes will keep updating their routing table with increasing metrics, this is known as "counting to infinity" and will cause the system to fail. The solution is to put a limit on the metric: 15, a metric of 16 mean a route is unreachable. B would then send a route to D with a metric of 16 (infinity) which indicates to C that it can no longer route to D. Furthermore no node should send a route for a network to a router that is adjacent to that network (split-horizon). Triggered updates in which a change in a route metric causes the routing table, to be immediately broadcast, allow a quicker convergence of routing tables in the system [Hedrick 1988].

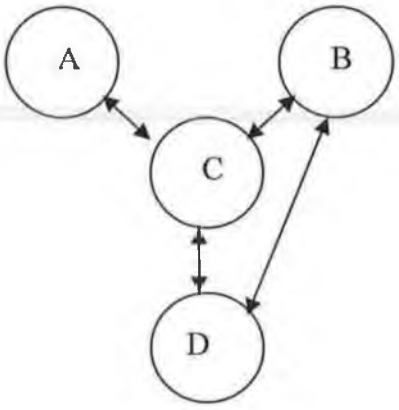

Figure 2.17 Routing Information Protocol counting to infinity [Hedrick 1988].

There are two versions of RIP; RIP1 and RIP2. Both versions are UDP based and typically use port 520. An RIP1 response broadcast is shown in figure 2.18.

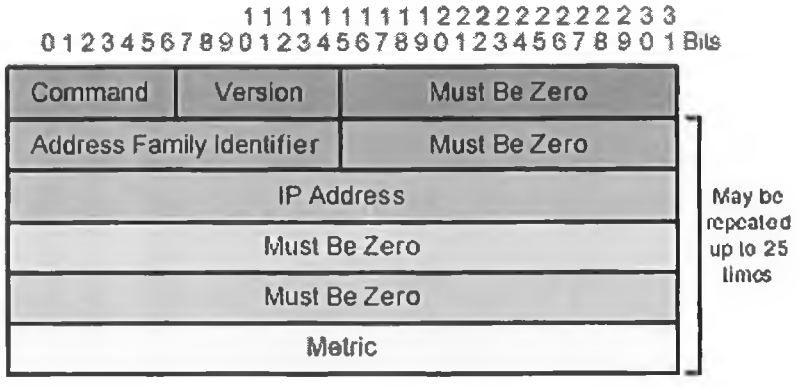

Figure 2.18 RIP1 frame format [Malkin 1998].

Only the IP address and metric for each route are transmitted. The subnet mask must be derived from the IP address class and the next router hop is the broadcasting router. An RIP2 response is shown in figure 2.19. It allows the subnet mask to be specified as well as the next hop router to send the packet. This allows routers to broadcast routes for other routers (which don't support RIP) on the same network [Malkin 1998].

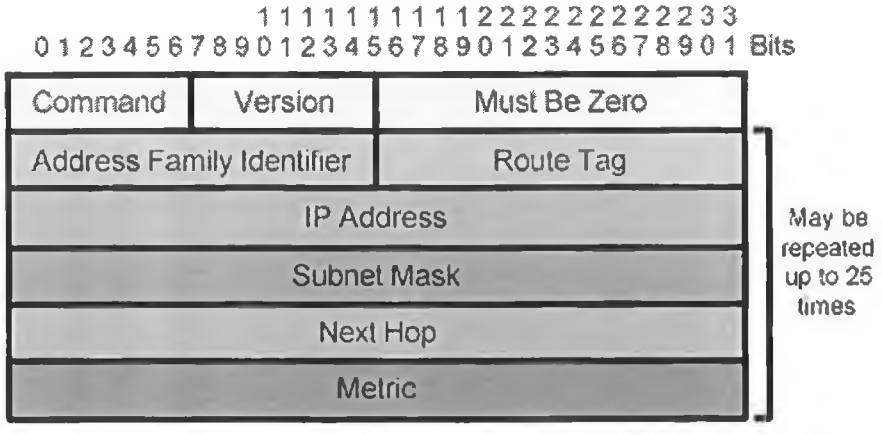

Figure 2.19 RIP2 frame format [Malkin 1998].

RIP1 and RIP2 are simple routing protocols to implement and are effective for small network topologies with less than 16 hops between nodes. Given its simplicity it would be well suited for routing between RTUs using radios and Ethernet.

"*Open Shortest Path First (OSPF) is the most widely used internal gateway routing protocol on the Internet*" [Schneider 2001]. OSPF is a link state routing protocol developed by the IETF. OSPF routes packets solely on the destination IP address. Unlike RIP it quickly detects router failures: it has a fast convergence. Each router in the autonomous system has the exact same routing information in its database and uses the same routing algorithm. OSPF uses the cost of a route as a metric. The cost can vary from delay to economic cost. If several routes exist with the same cost, the traffic is distributed evenly between them. The router builds up a database of all routes to all destinations. It then generates a tree of the shortest paths to each destination. This tree is used as the routing table. OSPF can also allow only authenticated routers to participate in the system. OSPF is a much more complicated dynamic routing protocol than RIP. Like RIP, OSPF sends its routing table regularly or on change to neighboring routers. The information is encapsulated in a raw IP packet (unlike RIP which uses UDP) [Moy 1998].

2.8.6 Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) allows IP devices to exchange information about networks problems. IP is an unreliable protocol, datagrams can gets lost for various reasons such as over loaded routers, TTL expiry, re-fragmentation time exceeded or unreliable communications links. However problems can arise where all datagrams get lost. This requires that some action be taken on the sender's side. ICMP is used to notify the sender of these problems so they can take appropriate action to rectify the problem. The header for ICMP messages is given in figure 2.20. There may typically be more than three fields. A summary of messages is given below for the Type field:

Figure 2.20 ICMP header [Postel3 1981].

Type	Message
0	Echo Reply, see Echo below.
3	Destination Unreachable, this message is typically sent from routers back to the source of a datagram to indicate that either a network, host is unreachable or a protocol or port on the destination host is not bound. It

is also sent when a datagram must be fragmented but the "*Don't fragment*" flag is set.

4 Source Quench, the source quench is sent to a host which it is sending two much information for the destination to process. If the sending host does not slow down, then packets will be lost.

5 Redirect, used to indicate to the sender of that a datagram a shorter route to the destination exists.

8 Echo, the most common user ICMP application is Ping. Ping is a utility available on most operating systems. It allows a message (Echo) to be sent to another host for the purpose of seeing if the host is accessible over the network or the Internet. The host then sends the message back (Echo Reply). Although this may seem like a trivial application it can be invaluable in tracking network faults.

11 Time Exceeded, this message is send back to the source of a fragmented datagram when all of its segments haven't being received in a finite time.

12 Parameter Problem, sent back to the source of a datagram because it options are invalid. Typically indicates the datagram is being discarded.

13 Timestamp, this message along with Timestamp Reply is used to determine the latency that a network is experiencing. The time required on the destination host to process the message and the time taken for it to propagate through the network can be found [Hall 2000].

14 Timestamp Reply, see Timestamp above

15 Information Request, this message is a way for a host to find out the number of the network it is on.

16 Information Reply, see Information Request above [Postel3 1981].

2.8.7 Address Resolution Protocol

As described in the previous sections, each computer on a TCP/IP network has a 4 byte IP address. However computers which have network interface cards such as Ethernet also have a 6 byte MAC address. As the MAC address is manufacturer assigned and an IP address is typically administrator assigned there is no logical mapping between the two. Therefore a method of converting protocol addresses (e.g., IP addresses) to LAN (e.g., Ethernet MAC addresses) is required. Address Resolution Protocol (ARP) is a protocol which is used for that purpose. The protocol uses two messages; ARP Request and ARP Reply to resolve IP address on an LAN into MAC addresses which identify a network

interface card. ARP can be used in network protocols other than IP for resolving protocol addresses into LAN addresses. The format of an ARP message is shown in figure 2.21. When a host on a LAN needs to send a frame to another host (on the same LAN), it must know the other hosts MAC address. It first sends out a broadcast Ethernet frame with ARP request message in the data field as in Figure 2.21. It then waits a finite amount of time for an ARP reply. When it receives the reply it checks the Sender Protocol Address field. If that matches the IP address sent out in the ARP request, then the Sender Hardware Address in the reply is used to send the frame to. Typically a computer keeps a cache of IP to MAC translations or an ARP table.

Bytes in field	2	2	1	1	2	n	m	n	m
	Hardware Address Space	Protocol Address Space	Hardware Address Length (n)	Protocol Address Length (m)	Opcode	Sender Hardware Address	Sender Protocol Address	Target Hardware Address	Target Protocol Address
Bytes in field	2	2	1	1	2	6	4	6	4
ARP Request	0x0001	0x0800	6	4	0x0001	Sender Hardware Address	Sender Protocol Address	All 1's	Target Protocol Address
Bytes in field	2	2	1	1	2	6	4	6	4
ARP Reply	0x0001	0x0800	6	4	0x0002	Sender Hardware Address	Sender Protocol Address	Target Hardware Address	Target Protocol Address

Figure 2.21 ARP Request/Reply [Plummer 1982].

Each time a request or reply is received, if the host is currently communicating with the source node, the IP-MAC address pair is added or updated in the ARP table. After an amount of time without any communications to the node, the ARP entry is removed from the cache. If no ARP reply is received when a request is sent out the request is send again for a fixed number of retries. If after the retries no reply is received, sending of the frame has failed [Plummer 1982].

2.8.8 Future of IP

The Internet has grown from being used in just desktop computers, server and routers to devices such as mobile phones, PDAs and digital cameras. However such expansion will cause a shortage of IP addresses in a few years given IPv4 has a 32-bit IP address. Techniques such as Network Address Translation (NAT) which allow a single IP address to be shared by multiple computers will prolong the life of IPV4; however a newer standard will be required which supports many more hosts. A newer version: Internet Protocol Version 6 (IPV6) has been developed to resolve the shortage of IP addresses.

IPV6 uses a 128-bit IP address. Its format differs from the dotted notation of IPV4 (a.b.c.d). Instead hexadecimal notation will be used with colons separating each 16-bit digit, e.g. FEDC:BA98:7654:3210:FEDC:BA98:7654:3210 [Murray 2001]. IPV6 makes the following additions to IPV4:

- 128-bit IP address drastically increases the number of nodes that can be used to the point address shortages shouldn't be a problem in the future.
- Header format simplified.
- More support for options and extensions.
- Packets can be "flow labeled"; this allows certain streams to be treated specially (such as real time data).
- Optional support for authentication and data privacy.
- Improved efficiency, reassembly by intermediate routers is not allowed. Also there is no checksum in the header, upper and lower layer checksums are sufficient to detect errors [Deering *et al* 1998].

The transition to IPV6 will require updating of network software and hardware components. It will be at least 2005 before the standard will be widely adapted [Hunt 1998]. Systems will initially use dual stacks which support both IP versions. IPV6 devices will be backward compatible with IPV4 and IPV4 systems will be updated to also support IPV6. In time IPV4 will be phased out [Murray 2001]. Given IPV6 is such a new standard which has yet to be deployed to most desktops are servers, it will be many years before it can be used in a SCADA system.

2.9 The Transport Layer

The transport layer is responsible for providing reliable and cost-effective data transport from source machine to destination machine. It provides these services to the session, presentation and application layers in the OSI. The two most common transport layer protocols are User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) and these are discussed in the following section [Tanenbaum 2003].

2.9.1 User Datagram Protocol

User Datagram Protocol (UDP) is a layer 4 protocol and provides a low overhead connectionless transport for applications that don't need or can't use connection-orientated services offered by TCP. It is the simplest protocol in the TCP/IP suite. UDP is most often used in applications which make heavy use of broadcast and multicast datagrams.

Most Internet applications use TCP as reliability and flow control are required so that data doesn't get corrupted during transport. UDP is useful in applications which send regular messages such as updates. Even if not all of the messages need reach their destination, as they are repeated regularly it won't cause the application to fail. UDP is an unreliable datagram transport protocol. Unlike TCP where it takes responsibility for the successful delivery of data, in UDP it is up to the application to ensure that the message reaches its destination. The UDP header format is given in figure 2.22.

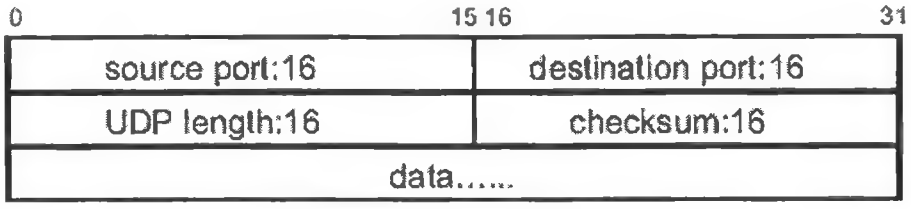

Figure 2.22 UDP header [Postel 1980].

The source and destination ports are used for multiplexing. This allows multiple applications to use UDP. Applications are assigned port numbers, e.g. Domain Name System (DNS) uses port 53 [Hall 2000]. The length field is the total length of the UDP datagram including header and data. The checksum is used to verify the integrity of the data. The checksum is the 16-bit ones complement of the ones complement sum of a pseudo header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets. The pseudo header is made up of fields from the IP header as in figure 2.23. When an application wants to use UDP it binds a socket to a port or uses an operating system assigned port. This is known as the source port. The port that the application sends datagrams to is known as the destination port.

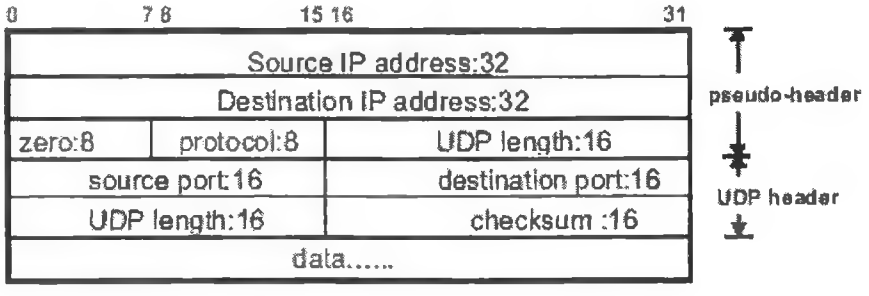

Figure 2.23 UDP pseudo header and UDP header [Postel 1980].

The de-multiplexer as shown is figure 2.24 uses the destination port on incoming datagrams to send the data to the appropriate application or socket. The IP protocol number for UDP is 17. UDP allows hosts to join multicast groups. A host can send a

datagram to a multicast group and all members of that group will receive the packet. A host doesn't have to be a member of the group to send a multicast datagram [Postel 1980].

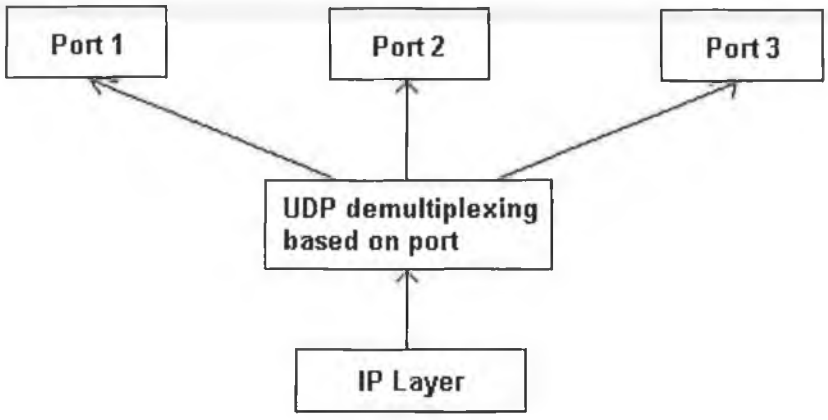

Figure 2.24 UDP port multiplexing.

2.9.2 Transmission Control Protocol

Transmission Control Protocol (TCP) is a reliable, connection-oriented and full duplex protocol. TCP is a layer 4 protocol and fits in the layer above IP (3) in the OSI architecture. TCP has the following characteristics:

- Connection based, a client must first establish a connection with a server before transferal of data.
- Basic Data Transfer, a continuous stream of bytes can be transferred. A "push" function allows data to be promptly forwarded to the receiver through the network.
- Reliability, ability to recover from lost, duplicate, damaged or datagrams received out of order.
- Flow Control, allows receiver to control the rate at which the sender sends data using a "window"
- Multiplexing, similar to that in UDP, ports are used to allow multiple processes within a single host to use TCP communication facilities simultaneously.
- Precedence and Security, allows the user to specify security and precedence of their communication [Postel4 1981].

TCP's high reliability makes it suitable for the transferring of large files. It is therefore used in Web and File Transfer Protocol (FTP) servers. The header for a TCP segment is given in figure 2.25 and a brief description of each field is given as follows:

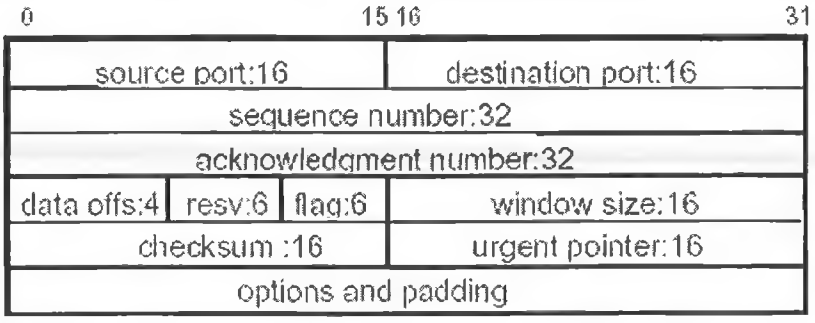

Figure 2.25 TCP header [Postel4 1981].

- Source port: calling port.
- Destination port: called port.
- Sequence number: the number of the first octet of data in the segment.
- Acknowledgment number: the next expected TCP octet.
- Data offset: the number of 32-bit words in the header.
- Reserved: always 0.
- Flags: control functions, e.g. setup and termination of a session.
- Window size: the number of octets the sender is can accept.
- Checksum: integrity check.
- Urgent pointer: indicates the sequence number of the end of the urgent data.
- Option and padding: options such as maximum TCP segment size can be negotiated.
- Data: upper layer protocol data

Like UDP, multiplexing is used to allow multiple processes to make use of the TCP as shown in figure 2.26. A socket is used to identify the connection at application level. A socket doesn't necessarily mean that there is a connection; it simply can be associated with a connection. Typically a socket is created first using the socket() function. When a connection is established it is then associated with the socket using connect() or accept().

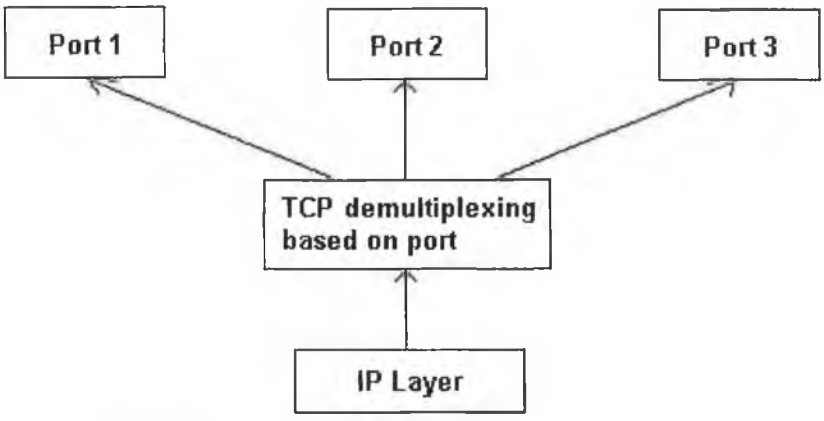

Figure 2.26 TCP port multiplexing.

There are two types of TCP socket; passive and active. Passive sockets are used in servers. They always remain in the LISTEN state, a copy is created for each incoming connections. Active sockets are used to connect to a server (which is in the LISTEN state). A TCP connection goes through several states during its lifetime as shown in figure 2.27. A connection starts in the closed state (no connection) and through a handshake goes into the established state. When the connection is terminated it goes back to the closed state.

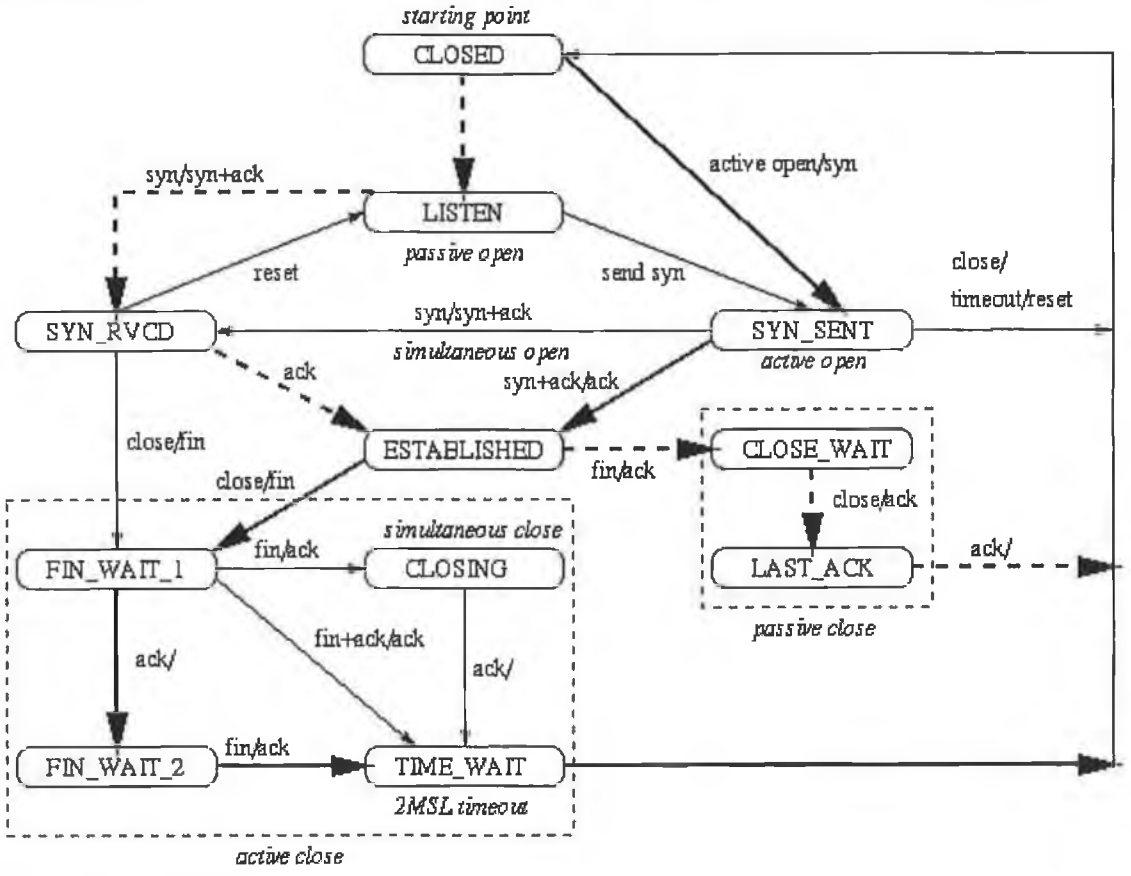

Figure 2.27 TCP state machine [Postel4 1981].

A description of each of the states is given below. A Transmission Control Block (TCB) is used to hold the state information along with other TCP variables.

- CLOSED, this state indicates that a connection doesn't exist and doesn't have a TCB associated with it.
- LISTEN, in this state the TCB is waiting for an incoming connection. To get a socket in to this state it must be first bound using bind() and then a listen() must be done. When an incoming segment is received in this state, a new TCB is allocated with the foreign/local IP address and TCP port. This TCB identifies the new connection from then onwards.
- SYN-SENT, a connection in this state has just sent a SYN segment and is waiting for a response. The connect() function is used to enter this state using sockets.

- SYN-RECEIVED, represents a connection having received a connection request (SYN) and having sent its own connection request along with an acknowledge (ACK) for the peers request. In this state it is waiting for the peer's ACK response to enter the established state.
- ESTABLISHED, this is normal state for the transferal of data and represents an open connection. When this state is entered either the sockets connect() function or accept() function succeeds, the latter in the case of a server.
- FIN-WAIT-1, represents a connection waiting for a terminate request from the foreign host, or an acknowledgement of a termination request already sent by the local host. The sockets function shutdown() or closesocket() are used to start the termination sequence.
- FIN-WAIT-2, represents a connection waiting for a terminate request from the foreign host.
- CLOSE-WAIT, in this state a connection has been closed by the foreign host and is waiting for the local user to also close the connection.
- CLOSING, represents a connection, waiting for an acknowledge of a terminate request already sent.
- LAST-ACK, in this state the foreign host has sent a terminate request and the local host has acknowledged it, the local host has also sent a terminate request and is waiting for the acknowledgement of it before it closes the connection.
- TIME-WAIT, this state is used to ensure that the foreign host has received the acknowledgement of its terminate request. When a timer initiated with twice the maximum segment lifetime (2MSL) expires, the TCB is deleted and hence the connection is closed.

TCP uses sequence numbers to identify data. Every byte of data in a segment occupies one sequence number. Acknowledgement numbers are used to indicate the reception of data in the correct order. A segment with a SYN or FIN flag takes up one sequence number. Every segment transmitted that contains sequence space must be acknowledged by the foreign host. If no acknowledgment is received, the segment is re-transmitted. The acknowledgment number is always the next sequence number the sending host is expecting to receive from the peer. Several segments can be sent (up to the window size) and only one acknowledgement needs be received as long as it covers all the sequence space sent in the segments.

To establish a connection, a segment is sent to the foreign host with the SYN flag set; an initial sequence number is chosen. The foreign host will then acknowledge the SYN (SYN takes up one sequence number) and send back its own SYN segment along with the initial sequence number it has chosen. Typically the ACK and SYN will be sent together (piggybacked). Finally when the local host receives the ACK and SYN segment it will send and acknowledge for the SYN and enter the established state. The purpose of the three-way handshake is to synchronize sequence numbers on both sides. Figure 2.28 illustrates the three-way handshake.

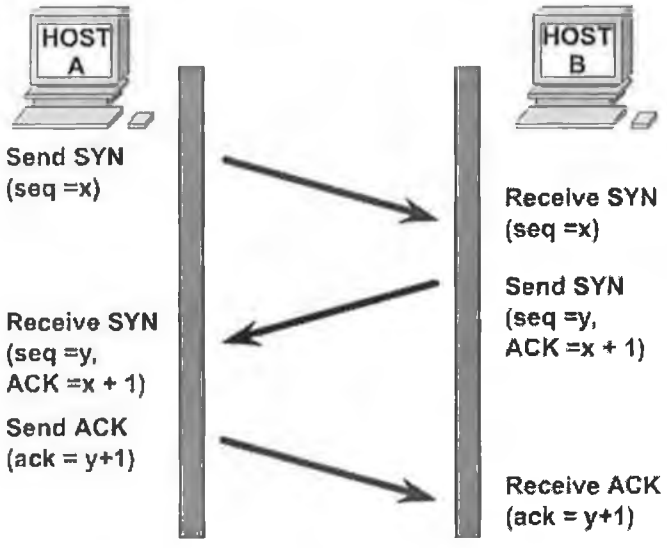

Figure 2.28 TCP connection sequence [Cisco1 2000].

Every byte of data sent has a sequence number (as it's only 32-bits, sequence numbers are not necessarily unique). Typically a connection is in the established state when sending data, however TCP does allow data to be queued during the initial three-way handshake when establishing a connection. Every segment with data must be acknowledged by the peer. TCP allow for cumulative acknowledgment; only one acknowledgment is sent for multiple segments. If a segment is not acknowledged a retransmission mechanism is used to retransmit the segment several times until it is acknowledged, or in the case of when no acknowledgment is received the connection is terminated. A window is used for flow control. The window represents the amount of buffer size available to the sender. It allows a host with limited memory or resources to control the amount of data received by keeping the window size small. The sockets functions send() and receive() are used for the transfer of data over TCP at application layer level. TCP also allows for "Urgent data". Using the urgent pointer with the URG flag set, the receiver can be notified of the presence of urgent data in the stream. How the application deals with such data is not specified by TCP. Figure 2.29 illustrates the sending of data.

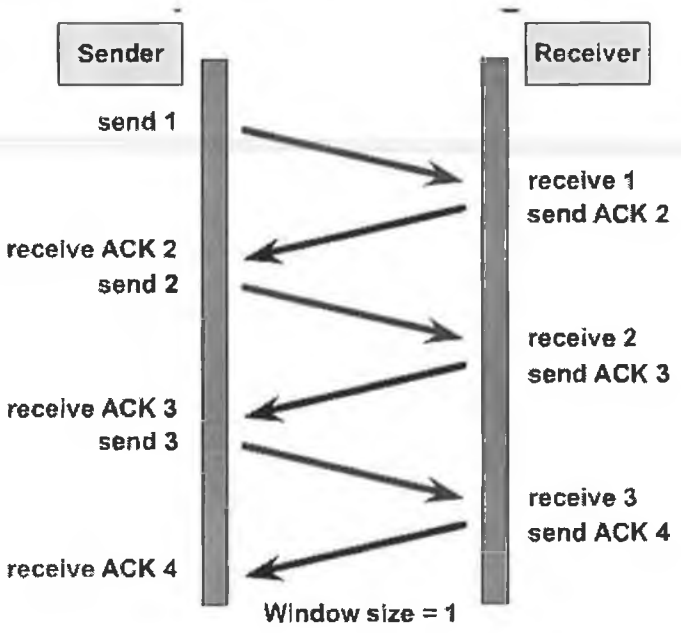

Figure 2.29 TCP basic send sequence [Cisco1 2000].

The typical close is similar to the connection sequence except FIN segments are used. Like an SYN, a FIN segment also takes up one sequence number. Closing a connection typically is done using the sockets function shutdown() or closesocket().

The Client/Server model on which TCP connections are based is a computer architecture where client processes request service from server processes. Services can be broadly interpreted to mean data, processing or the combination of the two. A file server is a simple example of a server. The client requests a particular file from the server, the server then transmits back to the file to the client. Figure 2.30 shows a diagram of the client/server architecture. The client/server architecture has the following distinct components which are usually implemented on either the client or server machine or both:

- User interaction/presentation component, usually a Graphical User Interface (GUI). A web browser is a typical example of a client side user interface.
- Application component, implements the requirements defined by the client/server application. In the case of a web server this is the processing of a request for a web page.
- Database management, performs the data manipulation and management required by the application such as retrieval and transferring a file [Pressman 1997].

TCP is the most commonly used client/server protocol on the Internet. A server application normally listens at a well-known address or port for service requests. The server process remains dormant until service is requested by a client's connection to the server's address.

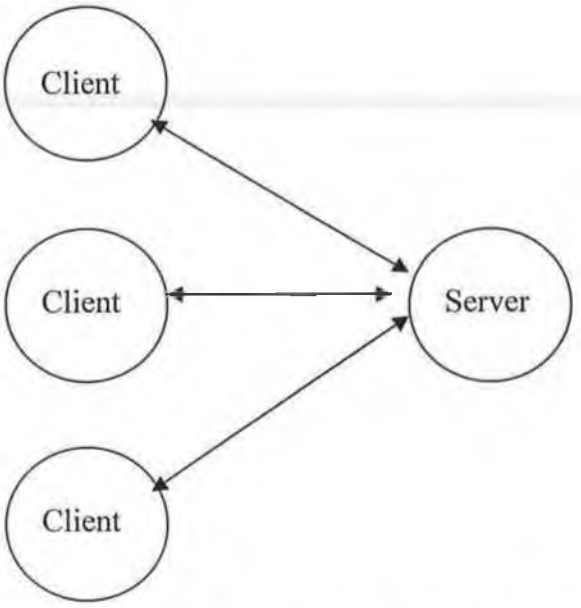

Figure 2.30 Client/Server model [Pressman 1997].

When a client requests service, the server process wakes up and provides whatever service the client requests. Figure 2.31 shows a flow chart for a typical server process under Berkley or Winsock.

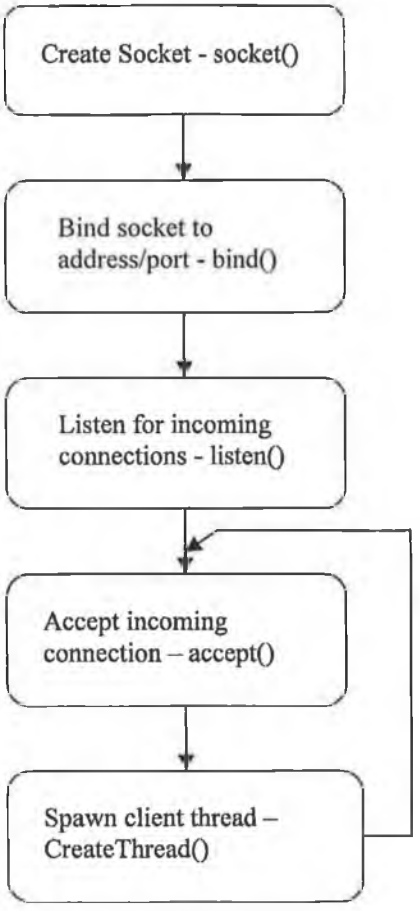

Figure 2.31 Multithreaded server using Win32 sockets [Hall *et al* 1993].

Firstly a socket is created using the socket() function. It is then bound to local interface (IP address) and TCP port. An interface address of "0.0.0.0" allows it to listen on all network interface cards on the system (multi-homed). The port is the TCP local port, e.g. Web Servers typically use port 80. The process then listens and accepts incoming connections using listen() and then accept(). There are two types of server; single-threaded and multi-threaded. In a single threaded server, there is only one thread – the main process thread. Each incoming connection is managed by the same thread. This involves the use of the select() function on a list of sockets to know which are ready to send or receive data [Hall *et al* 1993]. In a multi-threaded server each client connection is received by the main server thread and an instance of a client thread is created (in Win32 the CreateThread() or beginthread() function is used [Microsoft 2001]). The client thread then communicates directly with the client and processes all of its requests. When the client is finished it then closes the connection [Deitel *et al* 1999]. The resulting server is more responsive and efficient as each client thread can suspend on a socket. When data is received the thread responds immediately (assuming there no higher priority threads waiting), processes the client request and sends back a response. This is more efficient and responsive than individually polling a list of sockets at a regular interval.

2.10 The Application Layer

Application layer protocols are the most visible to the end user. The most common applications used on the Internet are the Web, Email and FTP. *Hyper Text Transfer Protocol* (HTTP) is the protocol used throughout the World Wide Web for the transfer of files such as Hyper Text Markup Language (HTML). When a user opens a web page, the browser connects to the HTTP server (web) on TCP port 80. The browser then sends a request to the server which specifies what operation to perform. The server then sends a response back to the browser. In the original HTTP1.0 specification, the connection is then closed. However the newer HTTP1.1 implementation allows persistent connections in which the browser can retrieve several files before closing the connection. Every request and response contains a header specifying the data in the message. In the case of the request message, the most important fields in the header are the method to be performed and the Uniform Resource Locator (URL – file) to retrieve (if any). The most common methods used are GET and POST. The GET method is used to download a file from the server. The POST method is used to send information such as input fields on a web page to a server [Fielding *et al* 1999].

File Transfer Protocol (FTP) is a protocol for transferring files from one computer to another computer over a network connection. A client computer uses an FTP client software program to request a file from another computer (server). FTP allows the reliable transfers of files using TCP and is independent of underlying file system on either server or client. FTP uses two connections, control and data. The control connection uses port 21 and the data connection uses port 20. The control connection is used to issue commands such as STOR (store file), MKD (make directory), etc. and is ASCII based. The data connection is used any time a directory listing or a file needs to be transferred. Figure 2.32 shows the FTP model for the transfer of information.

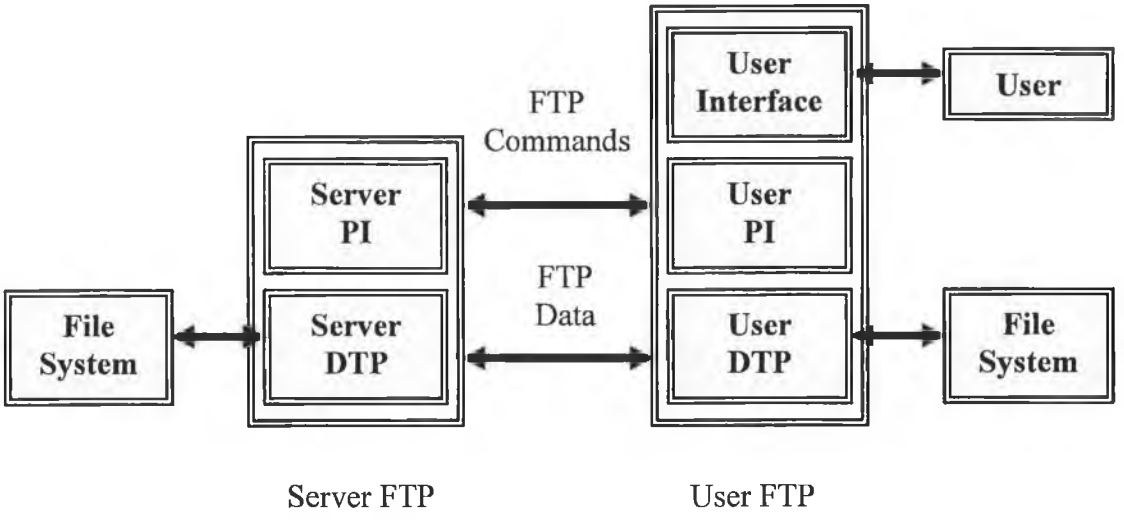

Figure 2.32 FTP model [Postel *et al* 1985].

The data connection is only made when needed, e.g. before the LIST (retrieve directory listing) command is issued over the control connection, the client makes a data connection to the server, using the PORT command it then indicates which data connection to retrieve the directory listing on. After the LIST command has been issued and the directory listing retrieved, the data connection is closed. FTP allows for the use of ASCII or binary mode data transfers. ASCII transfers are used to retrieve directory listings which are given in a formatted notation while binary transfers are used for the transfer of files. FTP allows for authentication of users and can support anonymous connections (no authentication). All common file and directory operations found in most operating systems such as make, delete, retrieve, store and rename can be implemented as well as path iteration functions [Postel *et al* 1985].

Telnet is a TCP extension to the commonly used terminal application. Terminal allows the executing of commands on a host over a serial communications port. A terminal program is used to type commands on and display the responses from the host. As the user type commands, each character is sent to the terminal server in real time over the serial port.

Finally when they hit return, the terminal server performs the requested operation and sends the response back over the serial port. The commands that are supported depend on the system, e.g. a PC based terminal server would allow DOS type commands to be executed. Telnet uses TCP port 23 for the transfer of data as apposed to a serial port [Postel *et al* 1983].

To send and receive *Email* on the Internet requires the use of two applications layer protocols. Simple Mail Transfer Protocol (SMTP) is used to transfer email from a client such as Microsoft Outlook to a server. This SMTP server then tries to deliver the email to a Post Office Protocol Version 3 (POP3) server associated with the recipient. When the recipient checks their email through a mail client, the messages are retrieved from the POP3 server. Figure 2.33 shows the typical scenario for mail delivery from sender to receiver.

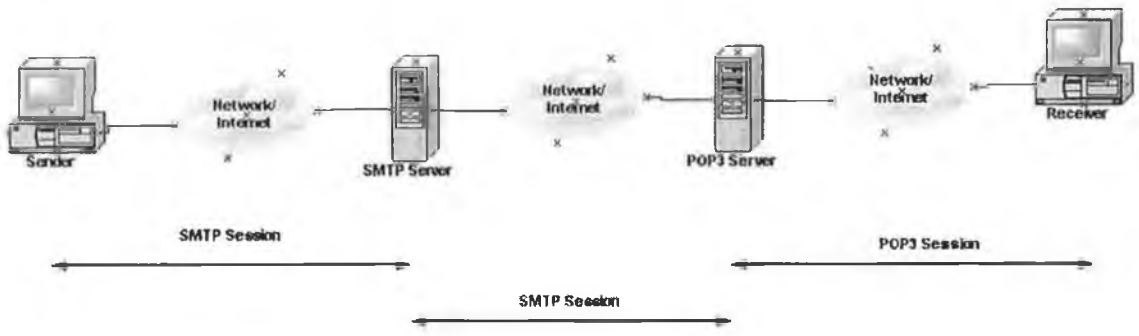

Figure 2.33 Email delivery [Tanenbaum 2003].

When the sender wants to send an email, using an email client the message is delivered to an SMTP server. To deliver the email the SMTP server uses the Domain Name System (DNS). DNS is used to resolve host names such as www.host.com to an IP address, e.g. "10.1.1.1". It can also be used to specify the IP address to where email for that domain should be send to. This is known as the MX record. Hence the SMTP server looks up the MX record for recipients' domain, e.g. if the recipients email address is joe.bloggs@testdomain.com, using DNS the 'MX record' (IP address) will be looked up for testdomain.com. The email will then be delivered to the SMTP server at that IP address. That server will then deliver to email to the POP3 server associated with the recipient. To check for new messages the recipient connects into the POP3 server using a client such as Microsoft Outlook and can then retrieve their email. SMTP normally uses TCP port 25 and POP3 normally uses port 110 although this is usually configurable. Both protocols are ASCII based and use commands for the transfer and delivery of messages [Tanenbaum 2003].

2.11 Embedded TCP/IP

Most Internet users will associate the Internet and hence TCP/IP as being used only on desktop PCs and routers. However TCP/IP has found its way in to many portable embedded devices. The implementation of TCP/IP in possibly low speed, low resource systems do pose some problems. A good TCP/IP stack should have the following features:

- Use pre-allocated buffers rather than from the heap using malloc.
- Timers for connection management such as in TCP should be based on the Real Time Operating System (RTOS) if present rather than a separate implementation.
- If an RTOS is used it shouldn't affect interrupt latency.
- All buffering mechanisms should have semaphore protection to allow multithreaded operation.
- Data copying should be kept to a minimum. Copying large buffers can add large time overhead and can increase the response time to connected clients.
- Link layer multiplexing, an interface should be defined between the TCP/IP stack and data link layer interfaces such as Ethernet or Point-to-Point Protocol. This will allow drivers for new types of interfaces to be easily added in future additions.
- The CPU time the stack required should be dependant on the application. Some applications such as streaming of video require more CPU intensive operations.

A designer has three options when deciding to put a TCP/IP implementation in an embedded system. The easiest is to use a hardware stack; this is the less flexible and adds extra manufacturing cost. The second option is to obtain a free software stack or to implement one. However there is limited technical support for this option and it may require developing a simple RTOS. Thirdly a stack can be purchased from a third party vendor. This may also have limited technical support. In either case it is up to the developer to handle porting to their particular interfaces [Herbert 2003].

Most of the Internet implementation is done using the 'C' language, in fact most operating systems are written in that language. The 'C' language is well suited to coding Internet protocols as it allows low level access to memory and I/O. Of the entire Internet protocols, TCP is the most complicated to implement. Unlike UDP, TCP is a connection based protocol and requires state information. The state can depend on three different events; user events such as the user opening a connection, network events such as segments being received from the network and time events such as timeout waiting for an ACK [Crowcroft *et al* 2002].

The use of Ethernet for real time control applications does pose one major problem, the time required to send a frame is not deterministic. This is due to the fact that it uses CSMA/CD. If a collision occurs when a frame is being sent the transmission must be retried. There is also the possibility that the sender may have to wait for a certain time before the network becomes idle. This sort of variable delay means Ethernet is not suitable to a hard real-time system. However Ethernet is gaining popularity for control applications. Switches along with faster Ethernet speeds (up to 1Gbps) have almost eliminated the problem [Flammini *et al* 2002].

A comparison was done between Profibus, CANbis2.0 and Ethernet IEEE802.3 by [Flammini1 *et al* 2002]. The research was done to see if a system could be implemented using Ethernet that would have the same performance to that of the other two fields buses. The system has one anolog input, along with two Light Emitting Diodes (LEDs). The analog input can be polled and the LEDs values can be set over the Internet through the Ethernet interface. The system uses a simple implementation of the protocols: IP, UDP, ICMP and ARP. This is the minimum amount of protocols required for a host to operate on a network. UDP was chosen as the application layer protocol to be used to poll the device. The reason being that it has low overhead; it has an eight byte header and doesn't have any flow control or error recovery like TCP. IP fragmentation is not supported either. This places a limitation on the size of datagrams that can be handled. However as the local network is Ethernet which has a frame data size which is more than enough for the UDP request and response. If the device is to be communicating over the Internet then the router on the local network must reassemble the fragmented datagram before forwarding it on the sensor. The system is based around an 8-bit 68HC11 microcontroller and CS8900A Ethernet controller with 10Base-T interface. In tests the sensor proved to be comparable in regard to cost and data throughput. Although the system is comparable to other field bus technologies and is easy to implement, it doesn't support TCP which over a WAN would result in data loss. Furthermore the option to force a router to reassemble datagrams may not always be possible.

Another system by [Flammini *et al* 2002] allows users to connect in over the Internet using a web browser and view the status of a humidity sensor. It was based on a PIC18C452 @ 40MHz and only required approximately 16kbytes of ROM and 1kbyte of RAM. The research proposed three possible means of providing multiple connections using TCP. The first being the normal mechanism of setting up the connection, exchanging the data in both directions (HTTP Get request/response), and finally terminating it when the user closes the browser. The second automatically terminates the connection with the response (FIN sent

with HTTP Get response). This allowed for "one shot" connection, each time the web pages is to be refreshed the connection must be remade. The third mechanism is a "stateless" system where the sensor holds no information about the state of the TCP connection. Instead it figures it out from the incoming TCP header field. However this system was discarded as it wasn't reliable enough for time critical applications. The system that was implemented was a mixture of the continuous and one shot system. One high priority user from the LAN could connect as a continuous user. Two other one-shot users could connect over the Internet to view the sensor's web page. Although the system has limited RAM (1536 bytes), it supports HTTP Get requests which were fragmented up to a length of 65535 byte. This is done by using a 128 byte de-fragmentation buffer. As all the important information in a HTTP Get request is contained in the first 68 bytes (such as HTTP command, URL and version) the rest of the fragmented datagram can be ignored as its fields inserted by the browser. Checksum calculations still have to be done to ensure data integrity and hence and are done as the segment is read. This system does allow for multiple connections, however only a Web server is supported. Not all applications are like HTTP in the sense that all the information that's needed is in the first 68 bytes or any arbitrary amount for that matter. Hence for microcontrollers with RAM sizes smaller to that of the Ethernet Maximum Receive Unit (MRU) and when IP reassembly is required, this mechanism is not sufficient.

To address the problem making TCP/IP/Ethernet based protocols more real-time research was done by [Park *et al* 1997]. The system developed was called LAN-TCP and is implemented on a MC68EN360 CPU with 512kbytes of RAM and a 10Base-T interface. LAN-TCP allocates a certain amount of bandwidth to each node by controlling the packet transmission interval. LAN-TCP maintains compatibility with existing TCP. Rather than sending a window size in segments which reflects the free buffer space, it sends a pseudo window which has a maximum value of the Maximum Segment Size (MSS) of Ethernet. This prevents the standard TCP node from transmitting bursty data and eliminates the requirement of IP fragmentation and reassembly which can be resource intensive for an embedded system. For transmission, the transmission interval is set to the minimum interval of the system clock. The normal TCP sliding window algorithm allows a sender to send several segments up to the maximum receiver window size without an acknowledgement. The acknowledgement is sent when the acknowledgement timer expires (typically 200 ms after last acknowledgement). LAN-TCP uses a Periodic Transmission Mode (PTM) timer to extend the acknowledgement timer period for a very long duration to avoid a series of ACK segments. The ACK packet isn't used for positive

acknowledgement but instead for window size notification and negative acknowledgements of lost packets. When ever the window from the normal TCP node is smaller than the MSS then the PSH flag is set in the next segment which will cause the normal TCP node to send back an ACK segment which can be used to update the LAN-TCP's value of the remote window. An example given shows a transmit buffer of 8192 bytes and a data length of 50 bytes, an acknowledgement is only sent every 150 packets. Analysis showed that PTM can reduce network traffic to below 50% because of the reduced amount of acknowledgements. Furthermore the number of collisions is reduced by 10%. Unlike other embedded TCP/IP stacks discussed, LAN-TCP does support multiple processes. However it does require knowing which process is sending a packet and how much bandwidth is reserved for it. A TCP implementation like this is only necessary for real-time applications and wouldn't be required in a typical embedded system that just wanted network connectivity.

TCP/IP Lean by [Bentham 2000] is an implementation of a Web server which supports dynamic content on an embedded device. The implementation is written in "C" and Ethernet or Serial Line IP (SLIP) interfaces can be used. The implementation is a much reduced version of a typical TCP/IP stack found in a desktop PC. It has limited support for IP fragmentation; only datagrams of one or two packets are supported. In the case of UDP, IP fragmentation can be ignored by simply making the maximum UDP datagram size set at the Maximum Transmit Unit (MTU) size of the network less the IP and UDP header sizes. Hence this requires knowledge of the fact at the application layer. For TCP, IP fragmentation can be ignored by setting the maximum transmit and receive segment sizes set at the Maximum Transmit Unit (MTU) size of the network less the IP and TCP header sizes. If these two considerations are taken care of, the only problem occurs when the network contains a link with a smaller MTU. In that case data throughput on the faster links may suffer since they are constrained to the size of the link with the smaller MTU.

An implementation is given for a PIC16C76 microcontroller with 8kbyte ROM and 368 bytes RAM connected to a PC using a SLIP connection. A simple Web server with dynamic contents is also detailed. There is a limit of 966 bytes on the amount of data in a TCP segment. This limitation is due to the fact that the state of a TCP connection is not stored locally on the microcontroller. It is instead figured out from examining the incoming segments. When a client (from a Web browser) connects to the server, the SYN handshaking sequence is done as normally, however the server doesn't hold any information about the connection. When the Hyper Text Transfer Protocol (HTTP) request is received, the GET/POST URL is processed along with any fields in the header. The

response is generated and is sent with the ACK for the request. A FIN is also sent in the same segment to ensure the client cannot request any more data on the connection. No retransmission timer is required since failure of the server to acknowledge the request from the browser will cause the browser's TCP to retransmit the request. Since the HTTP data and FIN are sent with the acknowledgement for the request this ensures if the response from the server gets lost, the client browser will automatically cause the retransmission of it by sending another HTTP request. Although this TCP/IP stack is small and can be implemented on a small microcontroller it may be too trivial for a system which requires multiple servers such as Web, FTP, and Telnet.

2.12 TCP/IP Security

TCP/IP is susceptible to many types of attack from hackers. Some are as simple as password guessing while others can be more serious attacks such as denial of service. Some of the most common attacks on TCP/IP are as follows:

- SYN flooding, this inhibits genuine users from accessing a server; it is a denial of service attack. A computer on usually a remote network continuously sends SYN segments. Each SYN segment causes resources to be allocated on the server. The server sends back an ACK, however the hacking computer usually sends a false IP address as the source. The result is all the server resources are tied up with partially made connections which take a certain time before they are freed. Hence no other user can use the service. One way around this is for Internet Service Providers (ISPs) to filter out packets from users whose IP addresses don't match what they were allocated. Other methods include thorough checking of TCP/IP header fields to ensure they are valid for the state of the connection.
- IP/Password sniffing, with a simple protocol analyzer, plain text passwords can be viewed on a network. In fact any data which is unencrypted can be seen. Hence passwords should always be encrypted.
- IP spoofing, this takes advantage of a "trust relationship" of a host. This hack is only possible if a server allows hosts to access it depending on their IP address. A malicious host sends datagrams with the source IP address with that of the trusted host; it pretends that it is the trusted host. In order for this hack to work the malicious host must be on the same network as the trusted host. Furthermore the real trusted host must be brought offline using something like the denial of service attack like the SYN attack. The reason for this being the real trusted host, would

respond with RST segments as it thinks the responses sent by the server (due to the malicious users segments) are invalid. It is also possible to "hijack" a TCP session. By sending a segment with a valid sequence number it can take over the connection as all segments sent by the spoofed host become invalid.

- RST and FIN attacks; these attacks use a protocol analyzer to learn sequence number of a connection between two hosts on a network. The malicious host can then sent a segment with a valid sequence number and either the RST or the FIN flag set to either host with the source address and ports the same as the host to be spoofed. This will cause the connection to be closed (denial of service attack).
- Ping O' death, this attack uses the Ping program which is used to test if a host is reachable. The maximum datagram can be sent is 65535 (limited by IP Fragment Offset field) – IP header (20) – Ping ICMP header (8) = 65507. This usually requires IP fragmentation and reassembly of the datagram. Some operating systems such as Windows 95/NT and Linux allow a user to specify a larger size than 65507. If the host that is pinged does not thoroughly check all the fields in the fragmented datagram, internal buffers can get overwritten causing the system to crash [Harrisa *et al* 1999].

A short expedition in London by employees of I-SEC, a security company, recently found that many companies are not using any form of security in their 802.11 wireless LANs. Wireless LANs by default have security disabled, many companies are complacent and don't go the bother of setting it up. All a hacker has to do is to listen for beacon messages which indicate a Wireless Access Point (WAP). When a WAP has been found they can interrogate its SSID (network identification number). The hacker can then access any resources on the wireless LAN which are not protected. In 30 minutes, 60 unsecured wireless networks were found. The expedition only used a laptop, 802.11 wireless LAN card and a directional antennae. The solution is to use encryption keys, IPsec and don't use TCP/IP for file and printer sharing and enable passwords on WAPs [NetSecurity 2002]. This highlights the security issues of using such an open system as TCP/IP over any form of radio. A SCADA system using and unprotected wireless devices could be a major safety risk.

TCP/IP is commonly used in manufacturing automation devices such as Programmable Logic Controllers (PLCs), similar to an RTU but except outputs are logic functions of the inputs. Some of the main threats to security and which may affect a TCP/IP SCADA system are:

- Denial of service, this is caused by incorrect configuration of automation devices. Two devices having the same IP address it a typical example. It could be caused accidentally by a person installing a device without checking to see if its address is already in use.
- Unauthorized access, users obtain access to resources or information they shouldn't such as trade secrets.
- Interception or alteration of traffic, this attack is less likely than the others described. The attack uses ICMP redirect messages. This causes hosts to change the router to which it sends datagrams. If there are two hosts having a conversation, it is possible to change one of them to using a potentially malicious router which may alter the conversation.

Some of the solutions to the above problems are:

- Network segmentation using simple routing. By splitting the networks into subnets connected using a router, internal traffic from segment A will not be seen on segment B. For example all office computers could be on one segment A, all automation devices on segment B and all servers on segment C etc. This stops malicious user who have access to one segment from viewing traffic on another segment. The only risk is when traffic has to travel from one segment to another.
- Segmentation with router access control. This is similar to the previous solution except the router is configured to only allow certain hosts on segment A to access certain hosts on segment B depending on source and destination IP address. E.g. only a PLC programmer's computer on an office network may access a PLC on an automation network which is on a different segment. This however may cause increased maintenance and inconvenience.
- Packet filtering, this takes the previous solution a step further and only allows access based on UDP/TCP port numbers. It is also configured on the router joining the segments. This can allow some users access to certain services on a device and other users other services or none at all, e.g. user A can only access a Web server (usually TCP port 80), user B can only access a FTP server (usually TCP port 21).
- Firewalls, this can be used to block access of external hosts to an internal network. Typically firewalls are used to block malicious hosts from the Internet to access internal networks. However in manufacturing automation they can be used as the interface between the manufacturing facility and the enterprise WAN [DePriest 1997].

Given the various attack methods that can be perpetrated on TCP/IP, a mechanism was developed by The Internet Engineering Task Force (IETF). The standard is known as IP Security (IPsec). IPsec covers IPV4 and the newer IPV6. It ensures data origin authentication, data integrity, replay detection, data confidentiality, limited traffic confidentiality and access control. IPsec has the following components:

- IP authentication header (IP AH), this is an addition to the original IP header. It provides data origin authentication, data integrity and replay detection. It consists of a random value which along with the destination IP address is used to identify the Security Association of the datagram, a sequence number to protect against replay attacks and authentication data which verifies the integrity of the protected fields.
- IP encapsulating security protocol (ESP), this is used either with or without the IP authentication header to provide payload data confidentiality.
- Security Association (SA), this represents an agreement between two hosts on security services to be applied to datagrams between them.
- Authentication and encryption algorithms [Molva 1999].

IPsec provides security of upper layer protocols that use IP such as TCP and UDP. Encryption of data involves computation processing which adds a large overhead to an embedded system with limited resources and CPU speed.

Secure Sockets Layer (SSL) is another TCP/IP security technology developed by the IETF. It allows client/server (TCP) connections to communicate securely without eavesdropping, tampering or forgery. Its goals are:

- Cryptographic security, allows the establishing of a secure connection between two parties.
- Interoperability, allows independent programmers to be able to encrypt data without knowledge of any other users keys.
- Extensibility, allows new encryption methods to be easily incorporated.
- Relative efficiency ensures that CPU utilization is kept to a minimum.

SSL uses various public/private keys encryption algorithms such as MD5 and Data Encryption Standard (DES) [Freier *et al* 1996].

SCADA systems can be an easy target for hackers and terrorists. Using TCP/IP, cyber assaults on a SCADA system can be carried out thousands of miles away. These attacks can be launched on any of the utilities industries from gas to water. American National Security Agency (NSA) officials ran a simulated attack on the SCADA systems controlling

the US power grid and military systems. They believe they can shut down the electric power grid in a number of US cities within days and disrupt military operations worldwide. The attack could be done using tools available on the Internet. The vulnerabilities of many SCADA systems are not recognized. Very few such systems are protected even by passwords – the weakest for of security. In many systems, all that is required is for the hacker to have knowledge of the protocols and equipment used by an organization [Smith 1998]. Although security has been increased since September 11th, using open protocols make it easy for a hacker to take control of a system. Given the post September 11th environment, it is crucial that the utilities industry ensure the SCADA systems controlling vital resources such as gas, electricity and water are well protected.

Chapter 3

Review of SCADA

3.1 Introduction

This chapter gives a background to Supervisory Control and Data Acquisition (SCADA). The main terms will be explained as well as some examples of typical applications. The typical elements of a SCADA system will be identified. These range from hardware like an RTU to communications protocols such as Modbus. The current uses of TCP/IP in SCADA will also be discussed.

SCADA systems are used to monitor and control plant and equipment in industries such as telecommunications, water, waste control, energy, oil and gas refining, manufacturing and transportation [Webopedia1 2001]. The main components of a SCADA system are *Remote Terminal Units* (RTUs), processing units and human-machine interfaces (HMIs). An RTU is a device installed at a remote location that collects data, codes it into a format suitable for communications and sends it back to a master station. It also collects information from the master device and implements processes that are directed by the master. RTUs contain input channels for sensing and metering and output channels for control, indication or alarms and a communications port [Webopedia 2001]. A HMI is used to display the status of the RTUs to the user through a Graphical User Interface (GUI) while data processing units are responsible for converting the format of the information so it can be distributed over a LAN and stored on a database. The HMI and data processing units are located in a control room and can be networked. The RTUs are situated in remote locations where plant devices are to be monitored. A communication link such as a leased line or radio links the RTUs and data processing units. Figure 3.1 shows a typical SCADA system.

The communication protocols used between the RTUs and the data processing units are often proprietary which makes future expansion difficult [Ong 2000].

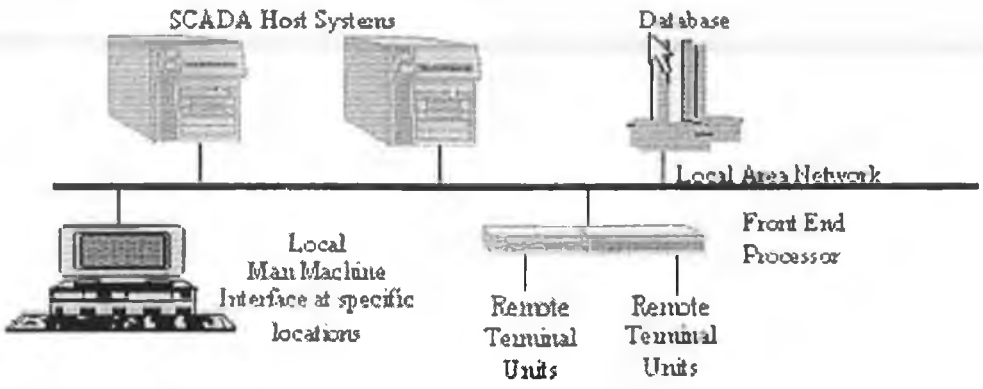

Figure 3.1 A typical SCADA system [Ong 2000].

"A SCADA system allows an operator in a location central to a widely distributed process, such as oil or gas field, a pipeline system, or a hydroelectric generating complex, to make set-point changes on distant process controllers, open or close valves or switches, monitor alarms, and gather measurement information." This distributed process can span thousands of kilometers. SCADA eliminates the need for an operator to permanently stay or visit remote locations in the normal operation of that remote facility. Hence it reduces the costs in having an operator make regular visits, especially if the locations are remote and requires special transport such as a helicopter. Some common examples of SCADA systems are:

- A group of hydro-electric generators in a remote location. In response to demands on the power grid, from a control room they can be turned on of off by opening or closing a valve to the turbine.
- Oil, water and gas production facilities such as well, fluid measurement, and pumps spread over a large geographical area can be monitored and controlled.
- Electric transmission systems covering thousands of square meters can be turned on or off in response to load changes on the lines.

These are just a few examples; SCADA can be installed in many other types of system. The typical signals monitored from remote locations include analog and digital values, alarms and totaled meter values. The signals that can be sent from the master station are usually analog (such as the value for a controller set-point) or digital (such as turn a valve on or off). These signals are connected into an RTU which is connected to the master station using a communications link. The most common types of communications link are serial cable, land line, leased line or radio. A serial cable is a pair of copper wires and can be used over very short distances, e.g. an RS232 cable can be used for distances less than

50 feet. If a modem is used however the distance is greatly increased. Land lines use a modem connected to the telephone network and involves the RTU dialing up the master station on alarm, or the master station periodically dialing into the RTU to poll its values. Leased line is similar to land line except it is a permanent connection over the telephone network to the master station. Finally radio allows remote control and monitoring in situations where cables aren't feasible and can be more cost effective than using a telephone network (don't have to pay line rental or make calls). In all cases the amount of data transferred in very small, for a basic RTU, 300 bits (approximately 30 bytes) per second in sufficient [Boyer 2002].

A *Programmable Logic Controller* (PLC) is another SCADA device similar to an RTU. "A Programmable Logic Controller is a small computer running a program. The program reads the inputs of the logic controller, calculates a custom logic function, and then produces the outputs" They are typically programmed using a language called ladder logic and are used in industrial automation [Wikipedia 2002].

3.2 SCADA Protocols

SCADA protocols can be divided into two categories: public and proprietary. The following sections review a proprietary protocol and several public protocols.

3.2.1 Modbus

Modbus is an application layer protocol (layer 7 of the OSI) that provides master/slave communication between devices connected on a bus or a network. It is the SCADA industries de facto standard since 1979. Modbus is a request/reply protocol and can be used over different types of media such as RS232, RS485, High-level Data Link Control (HDLC) and Ethernet. RS232 is a serial communications physical layer interface found in most PCs (COM port). RS485 is a multi-drop serial interface; several slaves (such as RTUs) can be connected to a master on the same line. Figure 3.2 shows the Modbus communications stack. Modbus uses a Protocol Data Unit (PDU) to encapsulate the request type (function code) and the data as shown in figure 3.3. This PDU is independent of the lower data link layer. The function code (1 byte) is the message type such as read coil, write register. The PDU data is the information read or written to the device. The PDU is encapsulated in an Application Data Unit (ADU). The ADU is specific to the underlying communications layer (figure 3.3 shows the ADU for RS232/RS485). The Additional address is a one byte slave id and the error check is a checksum to verify the

integrity of the frame. Modbus uses big-endian byte ordering. A different ADU is be used for RS232/RS485, HDLC or TCP/IP.

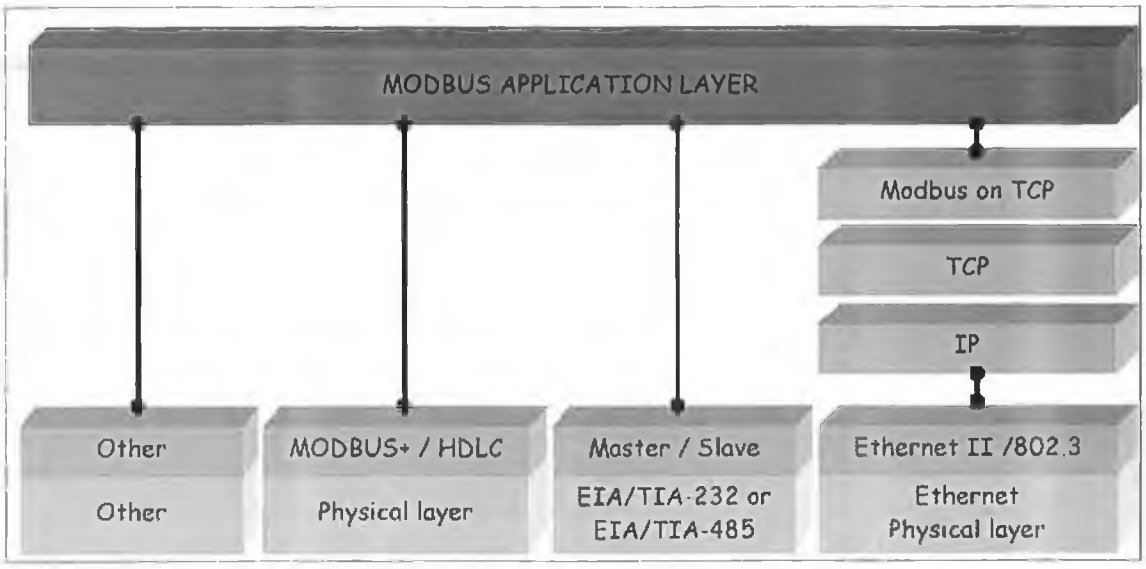

Figure 3.2 Modbus protocol stack [Modbus 2002].

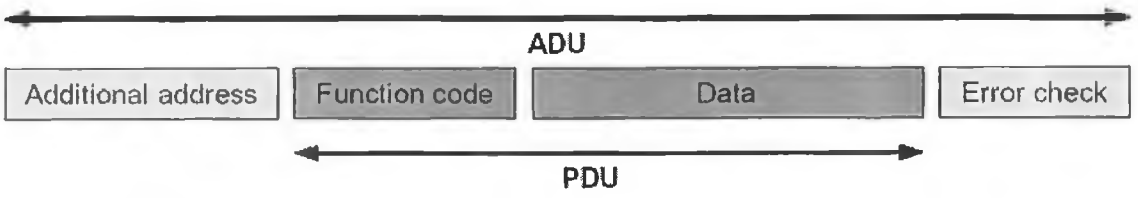

Figure 3.3 Modbus Application Data Unit [Modbus 2002].

Modbus defines the following PDU's:

- Modbus Request, this a request sent to a slave device to read or write values.
- Modbus Response, this is either the data polled sent back from the device or an indication that the values written to the device was successful.
- Modbus Exception Response indicates that the request sent to the device failed [Modbus 2002].

Modbus over serial links supports two transmission modes: ASCII or RTU mode. ASCII encodes each byte as two ASCI characters (hex); it makes the framing more transparent and allows messages to be sent over communications devices that only support 7-bit communications. Modbus supports several messages (function codes), of which the most commonly used are (function code number given first):

- 01 Read coils status – digital outputs.
- 02 Read inputs status – digital inputs.
- 03 Read holding registers – analog outputs.
- 04 Read input registers – analog inputs.

- 05 Force single coils – digital output.
- 06 Preset single register – analog output.
- 15 Force multiple coils – digital outputs.
- 16 Preset multiple registers – analog outputs [Modicon 1996].

Modbus over TCP/IP uses a different ADU to RS232. TCP port 502 is the standard for ADU communications. Modbus TCP servers can be multithreaded which gives the best performance. The Modbus TCP ADU is shown in figure 3.4.

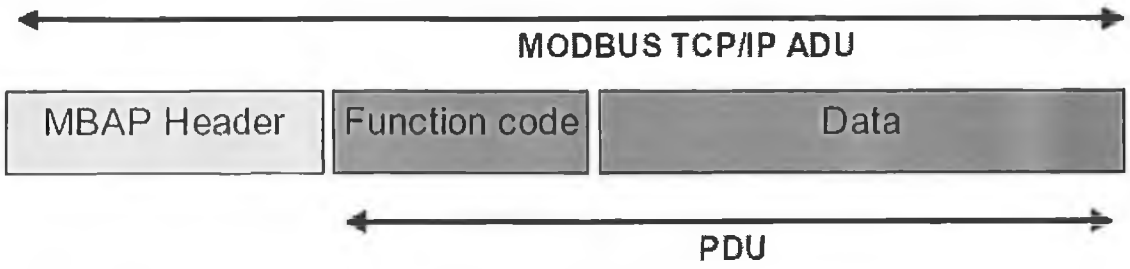

Figure 3.4 Modbus TCP Application Data Unit [Modbus1 2002].

No checksum is used as the underlying TCP/IP protocol already has several error checking mechanisms. The Modbus Application Protocol (MBAP) header contains four fields:

- Transaction identifier [2 bytes] used by client to identify responses for requests it has sent to a server.
- Protocol identifier [2 bytes], always 0 for Modbus.
- Length [2 bytes] of unit identifier + PDU.
- Unit identifier [1 byte], same as the additional address (slave id) [Modbus1 2002].

3.2.2 DATAC 922 RTU Protocol

Datac Control International Ltd., Dublin developed a proprietary protocol for a radio based SCADA system. The protocol occupies four layers of the OSI, physical, data link, network and application. The system consists of numerous RTUs placed in remote locations. One RTU has a Public Switched Telephone Network (PSTN) and a backup GSM modem link to a master station. Distant RTUs can be polled from the master station by using intermediate RTUs as routers. The radio uses a 400 – 500 MHz 4800 baud link. Each RTU has a static routing table which must be programmed during configuration. An example routing table for the master station (RTU 1) is given in table 3.1.

Des. RTU	2	3	4	5	6
Main path	2	2	2	5	5
Backup path	5	5	5	2	2
Retry – Main	2	2	2	2	2

Retry – Backup	1	1	1	1	1
Time out in sec	1	1	1	1	1

Table 3.1 Example master station routing table for Datac 922 RTU radio protocol [Datac 2001].

The Des RTU gives the destination of the route (column). The route to each RTU has a main and backup path. Similarly there are retries for each path as well as a timeout. In the case where the main path is the same as the destination, then no routing is done as the destination is the next node to the master. When the destination and main paths are different, the message is routed through the main path (intermediate node). Each RTU is not aware of the any other RTUs routing table. Each RTU has a similar routing table which is generated depending on topology. The format of the 922 RTU radio protocol frame is given in figure 3.5 and a description of each field is given below:

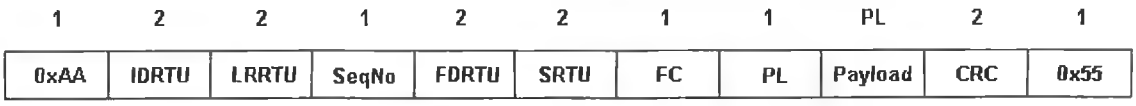

Figure 3.5 Datac 922 RTU radio protocol frame format [Datac 2001].

- The start (0xAA) and end (0x55) of frame are used for synchronization.
- IDRTU, Intermediate Destination RTU ID. This is the id of the next RTU that the message is to be routed through.
- LRRTU, this is the id of the last RTU that the message was routed through.
- SeqNo, sequence number. Used in matching requests and replies.
- FDRTU, final destination RTU.
- SRTU, source RTU.
- FC, function code. Indicates the function to be performed as a result of the message.
- PL, payload length, this is the length of the data in the frame.
- Payload, this can be the values read on a poll or can be the values to be updated.
- CRC, cyclic redundancy check for verifying the integrity of the message. If the CRC is invalid, the message is dropped.

When a message is to be sent from node A to node B, an acknowledgment is always sent from B to A to indicate it was successfully received. This is also the case if the message is being routed. E.g. if a message is to be sent from A to C through B, B sends an acknowledgment to A, and C sends an acknowledgment to B. The acknowledgment mechanism ensures a message reaches its intermediate node. A "message received" is sent

from the final destination RTU back to the source RTU when a message has being successfully routed and has reached its destination. Similarly a "not relayed" message is sent back to a source node if routing failed on one of the intermediate nodes. In the case of a receiving a "not relayed" message, the source node will retry sending the message a maximum of two more times until it gets a "message received" frame. Examples of the routing process are given in figures 3.6 to 3.9 and the routing tables for the six RTUs is given in figures 3.2 to 3.7 [Datac 2001].

Des. RTU	1	3	4	5	6
Main path	1	3	3	5	5
Backup path	5	5	5	3	3
Retry – Main	2	2	2	2	2
Retry – Backup	1	1	1	1	1
Time out in sec	1	1	1	1	1

Table 3.2 Routing table for RTU1 [Datac 2001].

Des. RTU	2	3	4	5	6
Main path	2	2	2	5	5
Backup path	5	5	5	2	2
Retry – Main	2	2	2	2	2
Retry – Backup	1	1	1	1	1
Time out in sec	1	1	1	1	1

Table 3.3 Routing table for RTU2 [Datac 2001].

Des. RTU	1	3	4	5	6
Main path	2	2	4	5	6
Backup path	5	5	5	2	5
Retry – Main	2	2	2	2	2
Retry – Backup	1	1	1	1	1
Time out in sec	1	1	1	1	1

Table 3.4 Routing table for RTU3 [Datac 2001].

Des. RTU	1	2	3	5	6
Main path	3	3	3	5	5
Backup path	5	5	5	3	3
Retry – Main	2	2	2	2	2
Retry – Backup	1	1	1	1	1
Time out in sec	1	1	1	1	1

Table 3.5 Routing table for RTU4 [Datac 2001].

Des. RTU	1	2	3	4	6
Main path	1	2	3	4	6
Backup path	2	3	2	3	3
Retry – Main	2	2	2	2	2
Retry – Backup	1	1	1	1	1
Time out in sec	1	1	1	1	1

Table 3.6 Routing table for RTU5 [Datac 2001].

Des. RTU	1	2	3	4	5
Main path	5	3	3	3	5
Backup path	3	5	5	5	3
Retry – Main	2	2	2	2	2
Retry – Backup	1	1	1	1	1
Time out in sec	1	1	1	1	1

Table 3.7 Routing table for RTU6 [Datac 2001].

Case: 1

- Assume a packet has to be sent to RTU4 from RTU1. The packet is sent to RTU2
- RTU2 - RTU3 RF link has failed; then the packet is sent to RTU5 which is in the Backup Path.
- Since RTU4 is in the RF range of RTU5 the packet is directly put to RTU4.
- Since the RTU5 - RTU4 RF link has failed, RTU5 sends a NOT RELAYED message to RTU1 (Back up path is not tried by RTU5 so as to prevent the packet from being in air infinitely).

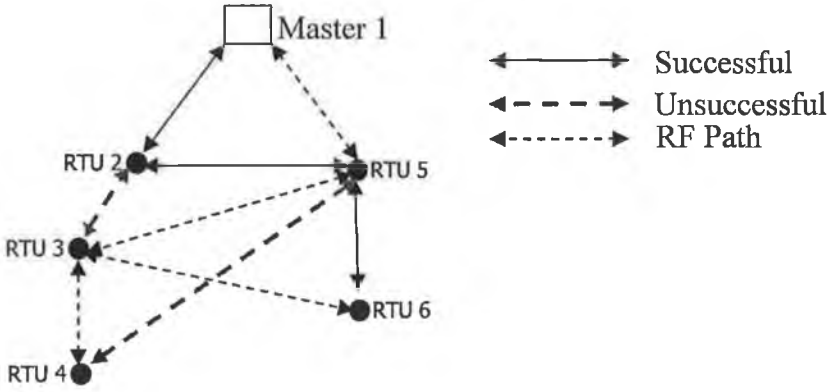

Figure 3.6 Sending a packet to RTU4 from RTU1 [Datac 2001].

Case: 2

- Assume a packet has to be sent to RTU6 from RTU1. The packet is sent to RTU5.
- RTU1 - RTU5 RF link has failed; then the packet is sent to RTU2 which is in the Backup Path.
- The Packet is sent to RTU5 by RTU2 which is in the main path to RTU6.
- Since the RTU2 - RTU5 RF link has failed; the packet is sent to RTU3 which is in the backup path to RTU6
- The packet is then sent to RTU6 which is the FDRTU.
- A received message is sent to RTU1 from RTU6

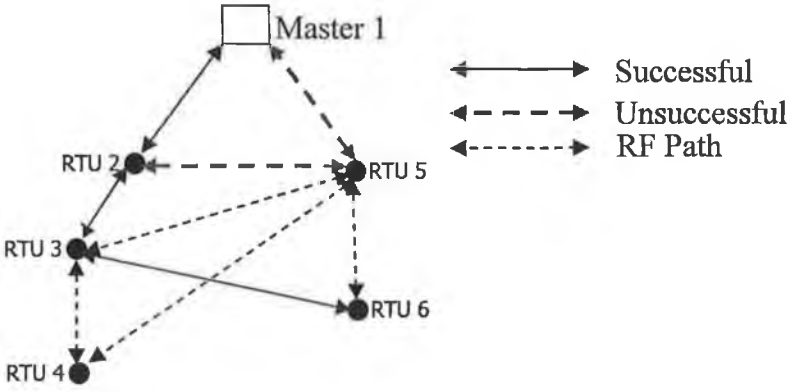

Figure 3.7 Sending a packet to RTU6 from RTU1 [Datac 2001].

Case: 3

- Assume a different packet has to be sent to RTU1 from RTU6. The packet is sent to RTU5.
- RTU5 - RTU1 RF link has failed; then the packet is sent to RTU2 which is in the Backup Path.
- The Packet is sent to RTU1 by RTU2 which is in the main path.
- A received message is sent to RTU6 from RTU1

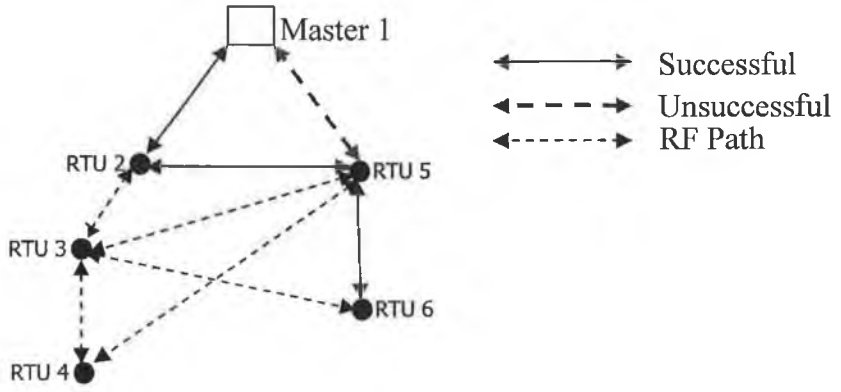

Figure 3.8 Sending a packet to RTU1 from RTU6 [Datac 2001].

Case: 4

- Assume a packet has to be sent to RTU2 from RTU1.
- Since the RTU1 - RTU2 RF link has failed; the packet is sent to RTU5.
- The Packet is sent to RTU2 by RTU5 which is in the main path.
- A received message is sent to RTU1 from RTU2

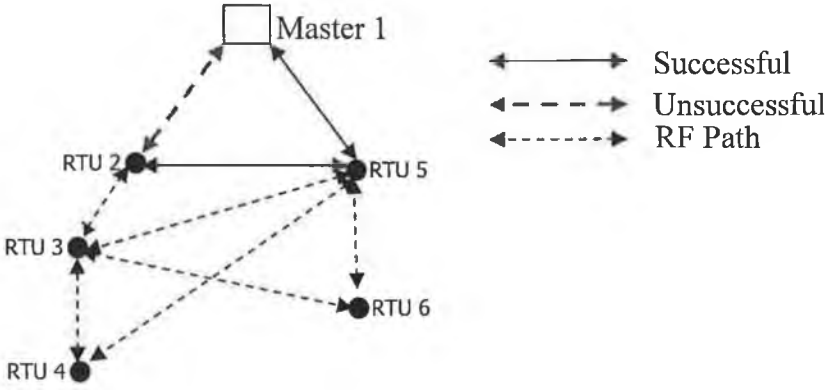

Figure 3.9 Sending a packet to RTU2 from RTU1 [Datac 2001].

3.2.3 Distributed Network Protocol version 3

Distributed Network Protocol version 3 (DNP3) is a protocol widely used in the electricity industry. DNP3 (DNP3) is an open and public protocol and was developed by Harris, Distributed Automation Products. It occupies the physical, data-link, network and application layers of the OSI. In 1993 DNP3 ownership and responsibility for the defining future version of the protocol was handed over to the DNP3 Users Group. They are a

group of utilities and vendors who use the protocol. DNP3 has the following robust and flexible features:

- Output options.
- Secure configuration/file transfers.
- Addressing for over 65,000 devices on a single link.
- Time synchronization and time-stamped events.
- Broadcast messages.
- Data link and application layer confirmation [dnp.org 2003].

DNP3 supports all the standard analog and digital I/O as found in most RTUs. It uses the term 'static' which represents the current values of an input or output. The current value of one or more inputs or outputs can be polled simultaneously. As well as polling DNP3 also supports events. Events are messages that are sent to the master station when an alarm occurs such as an input going outside its predefined limits. These can be setup over the protocol. DNP3 allows flexible topologies such as one-on-one (one master and one slave), multi-drop (like RS485 one master and multiple slaves) and hierarchal (an RTU can be master to an RTU lower down in the hierarchy and can be a slave to an RTU or master station upwards in the hierarchy). The frame format for a DNP3 message is shown in figure 3.10.

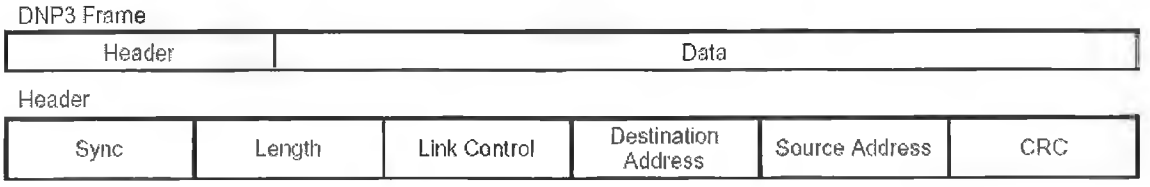

Figure 3.10 Distributed Network Protocol version 3 (DNP3) header and data frame [Curtis 2000].

The sync (2 bytes) is used by the receiver to determine where the frame begins. The length (1 byte) specifies the remaining bytes in the frame excluding the CRC bytes in the payload. The link control (1 byte) is used by the transmitter and receiver to coordinate activities. The destination (2 bytes) and source address (2 bytes) identifies the target device and the sending RTU respectively. Every DNP device has a unique address. DNP3 supports 65520 individual addresses as well as broadcasts. The CRC (2 bytes) is used for verifying data integrity. A CRC is sent for every 16 bytes of data in the payload. This provides a high level of assurance of data integrity. Acknowledgments for frames are sent along and optional retransmissions are also supported. The maximum frame is 292 bytes (250 bytes payload without CRCs). For larger frames the transport layer (which is incorporated into the application layer) uses one byte in the data link payload for fragmentation and

reassembly. Fragmentation and reassembly can also occur in the application layer. The breaking down of messages in these two layers is similar to that used in TCP/IP where the IP layer can fragment and reassemble datagrams. Similarly TCP allows data to be segmented in to smaller IP datagrams. DNP3 uses variation and objects numbers to identify the type of data being transmitted. Variation numbers indicate the type of I/O such as 32-bit analog input or 64-bit floating point value. Object numbers can be used to indicate whether the data is static (polled current value) or an event due to an alarm. Like Modbus, DNP3 has been implemented over TCP/IP. This allows the control of devices over a wide area. The main advantages of DNP3 are:

- Interoperability between multi-vendor devices.
- Reduced costs and faster delivery of product.
- Less testing, maintenance and training.
- Improved documentation and independent conformance testing.
- Easy system expansion [Curtis 2000].

3.2.4 Manufacturing Automation Protocol and Technical Office Protocol

Manufacturing Automation Protocol (MAP) was developed by General Motors and uses the IEEE 802.4 token ring passing bus adopted by many manufacturing companies in the early nineties. It allows real-time deterministic delivery of packets. Under the token ring passing bus which normally uses coaxial cable as the transmission medium, a station can transmit its packets as long as it holds the network token. There are mechanisms in every station to correct errors such as multiple tokens, lost tokens, failed stations, addition of new stations and duplication station addresses. During normal operation, the token is passed from the highest addressed station in the network to the next highest addressed station in descending order. After the token reaches the lowest addressed station, it is passed to the highest address station. The circular passing of the token, forms a logical ring and hence the name "token ring" [Hwang *et al* 1992].

MAP is based on the OSI model of layered protocols. Some of the most common application layer protocols are:

- File Transfer Access Method (FTAM), used for file transfer.
- Manufacturing Message System (MMS) for transferring messages to robotic equipment.
- Association Control for Service Elements (ACSE), program to program communications.

Technical Office Protocol (TOP) was designed for the office environment and can use IEEE 802.3 Ethernet, Token Bus (IEEE 802.4) and Token Ring (IEEE 802.5). Like MAP it is based on the OSI model. Typically TOP is used for design in the office using Computer Aided Design (CAD) and MAP is used in the manufacturing floor using Computer Aided Manufacturing (CAM). Like MAP, TOP also contains FTAM and ACSE, but MMS is not included. Some other common TOP applications are:

- Message Handling Service (MHS), used for electronic mail transfer of all types of documents.
- International Organization for Standardization (ISO) Virtual terminal allows any machine to log on to another.
- ISO Office Document Architecture and Office Document Interchange Format permits formatted documents to be transferred regardless of the package used to create them.
- ISO Computer Graphics Metafile (CGM) permits the exchanges of geometrical graphics.
- American National Standards Institute (ANSI) Initial Graphics Exchange Standard (IGES), used to exchange design information among CAD/CAM workstations [Jack 2001].

Due to improvements in speed and performance, Ethernet is now becoming common place in the factory floor. Using switched networks eliminates some of the traditional problems of using Ethernet for real-time applications since critical data can be given priority (determinism). Ethernet and TCP/IP are now replacing the MAP/TOP protocols since users were not able to reach agreements on the MAP/TOP specifications. In addition MAP/TOP failed to catch on with industrial users because it was too complex. Ethernet also offers better scalability than special-purpose MAP/TOP industrial networks and has lower maintenance costs [Marsan 2000]. Although MAP/TOP has some similar applications to TCP/IP such as file transfer and electronic mail, they are still totally incompatible. There would be little value in supporting MAP/TOP in an Internet compatible RTU. The MAP/TOP protocols which are not Internet compatible are being replaced by Ethernet and TCP/IP, the protocols which form the foundation of the Internet.

3.2.5 LS900, HART and IEC 60870

LS900 is a proprietary SCADA protocol developed by Datac Control International Ltd, Dublin and allows the control and monitoring of RTUs. Since it is proprietary protocol

and the specification is not available, it will not be discussed in this thesis. Highway Addressable Remote Transducer (HART) uses the traditional 4-20mA measurement or control signal with a superimposed bi-directional digital communications current signal that provides additional information about the field device. HART devices are usually flow, pressure and temperature sensors which do not support TCP/IP. For this reason, HART will not be reviewed in this thesis. IEC 60870 is another SCADA protocol with similarities to DNP, however since there is no TCP/IP version, it also is not reviewed in this thesis.

3.3 Remote Terminal Units

Datac Control International Ltd., a SCADA company based in Dublin has three RTU products: the Micro RTU, the 922 RTU and the 932 RTU. Table 3.8 lists the features of each RTU. None of the RTUs has any support for TCP/IP or network connectivity such as Ethernet. Neither has a file system or mass storage device. These RTUs have similar features to those sold by most SCADA companies. Any of the RTUs can be used in remote locations using a radio, PSTN or GSM link. The Micro RTU would be unsuitable for TCP/IP use as it has very limited CPU, RAM and ROM resources.

	Micro RTU	922 RTU	932 RTU
CPU	8051	Hitachi 32-bit H8S 2322	Mororola 32-bit 68000
CPU Speed	20MHz	20MHz	20MHz
RAM	1 kbyte	2MB	4MB
ROM	16 kbyte	2MB	2MB
Operating System	None	Nucleus	OS9
Built in I/O	2 DIPs, 2 CIPs	8 DIPs, 8 DOPs, 8 AIPs	Slot in I/O cards
Distrbuted I/O	None	None	Modbus, HART
RS232 ports	1	6, 4 fully stuffed	2
RS485 ports	None	4 (jumper selectable)	1
GPS	None	Optional	Optional
Protocols	Proprietry	Modbus, 922 RTU Protocol	Modbus, LS900, IEC-870
Radio	UHF Data Modem	Optional	Optional
GSM	Optional	Optional	Optional
PSTN	Built in Modem	Optional	Optional
Liquid Crystal Display	None	None	Optional
LEDs	None	8	None
PID Control	None	None	Optional
Power Supply	9 - 16V DC	4.5 - 28V DC	24 V DC
Logging	None	Yes	None
Alarm Notification	Yes	Yes	Yes
Low Power Sleep Mode	Yes	Yes	None

Table 3.8 Datac RTU product range features [Datac 2003].

The 922 RTU would present the biggest scope for development with TCP/IP and its associated applications as it supports a lower power sleep mode, has no distributed I/O, few master/slave protocols and no network connectivity (such as Ethernet). The 922 RTU has 2MB Flash ROM. The chip is a Thin Small Outline Package (TSOP) and is soldered directly onto the board. A bootloader, application code, configuration data, routing table and data points (database) set are stored on it. The application code is stored as the binary application image. Records or structures are used to store the configuration data, routing table and database.

To configure the 922 RTU, a Java configurator is used. It connects in over a dedicated serial port. A proprietary protocol is used between the RTU and configurator PC. A user interface task runs on the 922 RTU which handles command from the configurator. It is American Standard Code for Information Interchange (ASCII) based and runs at 57600 bps. An ASCII based protocol converts all payload data into hex characters. Hence the byte 5 decimal converts to "05" (two bytes). This makes the protocol more transparent. Various messages types are used to perform different commands such as read or write to the flash, restart RTU etc.

The 922 RTU uses a proprietary protocol which allows a network of RTUs placed over a wide area to communicate over radio. Each 922 RTU effectively becomes a router for RTUs lower down in the chain. A master station at the top of the chain can poll any RTU lower down even if it doesn't have a direct communications link. Instead it routes messages through intermediate RTUs. The protocol also allows an RTU to report an alarm back to the master station. A radio modem is connected to one of the 922's serial ports. "The 922 RTU Protocol" as described in section 3.2.2 is used for communications between the 922 RTU and the master station. The protocol supports a store and forward mechanism. This allows any RTU to route message from RTUs lower down in the chain, up to the master station and vice versa. The values of any I/O or GPS data points of any RTU can be polled from the master station. The protocol also allows alarms to be sent to the master station from any RTU when a data point's value goes outside preset limits. A static routing table is used in each RTU. It must be programmed using a GUI based interface on the configurator during setup and installation and is different for each RTU. RealFlex (HMI software) is used at the master station. A driver has been developed especially for the radio protocol.

The 922 supports the periodic logging of any data point. The logging contains the current value of the data point, tag number and a date and time stamp. The logging data can be