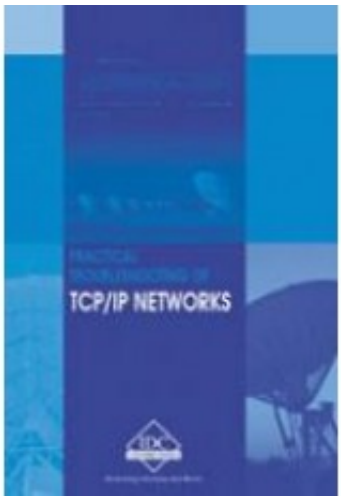
IDC Technologies - Books - 1031 Wellington Street
West Perth
WA 6005

Phone: +61 8 9321 1702 - Email: books@idc-online.com

TT-E - Practical Troubleshooting of TCPIP Networks



Price: $65.95

Ex Tax: $59.95

Short Description

This manual covers the essentials of network management, including fault finding at Ethernet/IP/TCP and application levels.

Description

This manual covers the essentials of network management, including fault finding at Ethernet/IP/TCP and application levels.

Table of Contents

Download Chapter List

Table of Contents

First Chapter

Chapter 1: Introduction

1

Introduction

This chapter deals with the OSI model, its relevance in terms of the technologies dealt with in this manual, and a generic troubleshooting approach.

Learning objectives

After studying this chapter you should be able to:

- Understand the basic concept of the OSI model
- Understand the concept of client-server operation
- Use these concepts in order to formulate a general troubleshooting philosophy for Ethernet/TCP/IP-type networks

1.1 Network types

Communication networks evolved due to the need to exchange and share information amongst a group of machines. During the last century many kinds of communication networks have been developed, such as telephone networks, computer networks and cable TV networks.

With the need for data exchange superseding voice and picture transmission, computer networks have become the most prevalent of all communication networks. Depending on the distances between the computers, computer networks can be further differentiated into:

- **LANs (Local Area Networks)**. These networks interconnect computers and other networked devices located a small distance apart, for instance computers in an office or building. The most popular LAN technology is Ethernet.
- **MANs (Metropolitan Area Networks)**. These networks interconnect computers and other networked devices located at medium distances from each other, for instance around the perimeter of a large city. Technologies used here include Ethernet and FDDI.
- **WANs (Wide Area Networks).**These are interconnected LANs, located at large distances from each other, for instance in different cities or

countries. Interconnection is via a ‘communications cloud’ that encompasses technologies such as SDH, ATM and X.25. LANs connecting to the WAN ‘cloud’ always do so via routers.

1.2 The OSI model

A communication framework that has had a tremendous impact on the design of computer networks is the Open Systems Interconnection (OSI) model of the International Organization for Standardization (ISO). The objective of this model is to provide a framework for the co-ordination of standards development, and allows for existing as well as evolving standards activities to be set within that common framework.

In order to understand the structure of all the technologies discussed in this book, a quick review of the OSI model basics is a necessity.

1.2.1 Open and closed systems

The interconnection of two or more devices with some form of digital communication is the first step towards establishing a network. In addition to the hardware requirements as discussed above, the software problems of communication must also be overcome. Where all the devices on a network are from the same manufacturer, the hardware and software problems are usually easily overcome because all the system components have usually been designed within the same guidelines and specifications.

Proprietary networks that comprise hardware and software from only one vendor are called closed systems. In most cases these systems were developed at a time before standardization, or when it was considered unlikely that equipment from other manufacturers would be included in the network.

In contrast, ‘open’ systems conform to specifications and guidelines that are

‘open’ to all. This allows equipment from any manufacturer that complies with that standard to be used interchangeably on the network. The benefits of open systems include wider availability of equipment, lower prices and easier integration with other components.

1.2.2 The OSI concept

Faced with the proliferation of closed network systems, the ISO defined a ‘Reference Model for Communication between Open Systems’ (ISO 7498) in 1978. This has since become known as the OSI model. The OSI model is essentially a data communications management structure that breaks data communications down into a manageable hierarchy (‘stack’) of seven layers. Each layer has a defined purpose and interfaces with the layers above it and below it.

By laying down functions and services for each layer, some flexibility is allowed so that the system designers can develop protocols for each layer independently of each other. By conforming to the OSI standards, a system is able to communicate with any other compliant system, anywhere in the world.

The OSI model supports a client/server model and since there must be at least two nodes to communicate, each layer also appears to converse with its peer layer at the other end of the communication channel in a virtual (‘logical’) manner. The concept of isolation of the process of each layer, together with standardized interfaces and peer-to-peer virtual communication, are fundamental to the concepts developed in a layered model such as the OSI model. This concept is shown in Figure 1.1.

Figure 1.1
OSI layering concept

The actual functions within each layer are provided by entities (abstract devices such as programs, functions, or protocols) that implement the services for a particular layer on a single machine. A layer may have more than one entity; for example a protocol entity and a management entity. Entities in adjacent layers interact through the common upper and lower boundaries by passing physical information through Service Access Points (SAPs). A SAP could be compared to a predefined ‘postbox’ where one layer would collect data from the previous layer. The relationship between layers, entities, functions and SAPs is shown in Figure 1.2.

Figure 1.2
Relationship between layers, entities, functions and SAPs

In the OSI model, the entity in the next higher layer is referred to as the N+1 entity and the entity in the next lower layer as N–1. The services available to the higher layers are the result of the services provided by all the lower layers.

The functions and capabilities expected at each layer are specified in the model. However, the model does not prescribe how this functionality should be implemented. The focus in the model is on the ‘interconnection’ and on the information that can be passed over this connection. The OSI model does not concern itself with the internal operations of the systems involved.

When the OSI model was being developed, a number of principles were used to determine exactly how many layers this communication model should encompass. These principles are:

- A layer should be created where a different level of abstraction is required
- Each layer should perform a well-defined function
- The function of each layer should be chosen with thought given to defining internationally standardized protocols
- The layer boundaries should be chosen to minimize the information flow

across the boundaries
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy

The use of these principles led to seven layers being defined, each of which has been given a name in accordance with its purpose. Figure 1.3 shows the seven layers.

Figure 1.3
The OSI reference model

The service provided by any layer is expressed in the form of a service primitive with the data to be transferred as a parameter. A service primitive is a fundamental service request made between protocols (Figure 1.4). For example, layer W may sit on top of layer X. If W wishes to invoke a service from X, it may issue a service primitive in the form of *X.Connect.request* to X.

Figure 1.4
Service primitive

Typically, each layer in the transmitting stack, with the exception of the lowest, adds header information, or Protocol Control Information (PCI) – a.k.a. 'header', to the data before passing it across the interface to the next layer. This interface defines which primitive operations and services the lower layer offers to the upper one. The headers are used for peer-to-peer layer communication between the stacks and some layer implementations use the headers to invoke functions and services at the adjacent (N+1 or N-1) layers.

At the transmitting stack, the user application (e.g. the client) invokes the process by passing data, primitive names and control information to the uppermost layer of the protocol stack. The stack then passes the data down through the layers of the stack, adding headers (and possibly trailers), and invoking functions in accordance with the rules of the protocol at each layer.

At each layer, the ‘data’ received at a certain layer (including headers from the layers above it) is referred to as a Service Data Unit or SDU. This is normally prefixed with the first letter of the name of the layer. For example, the Transport layer receives a TSDU from the Session layer. The Transport layer then processes it, adds a header, and creates a Transport Protocol Data Unit or TPDU.

At the receiving site, the opposite occurs with the headers being stripped from the data as it is passed up through the layers of the receiving stack. Generally speaking, layers in the same stack communicate with parameters passed through primitives, and peer layers communicate with the use of the headers across the network.

At this stage it should be quite clear that there is no physical connection or direct communication between the peer layers of the communicating applications. Instead, all physical communication is across the lowest (Physical) layer of the stack. Communication takes place downwards through the protocol stack on the transmitting node and upwards through the receiving stack. Figure 1.5 shows the full architecture of the OSI model, whilst Figure 1.6 shows the effects of the addition of headers to the respective SDUs at each layer. The net effect of this extra information is to reduce the overall bandwidth of the communications channel, since some of the available bandwidth is used to pass control information.

Figure 1.5
Peer layer interaction in the OSI model

Figure 1.6
OSI message passing

1.2.3 OSI layer services

The services provided at each layer of the stack are as follows.

Application layer

The Application layer is the uppermost layer in the OSI reference model and is responsible for giving applications access to the protocol stack. Examples of Application-layer tasks include file transfer, electronic mail (e-mail) services, and network management. In order to accomplish its tasks, the Application layer passes program requests and data to the Presentation layer, which is responsible for encoding the Application layer's data in the appropriate form.

Presentation layer

The Presentation layer is responsible for presenting information in a manner suitable for the applications or users dealing with the information. Functions such as data conversion from EBCDIC to ASCII (or vice versa), the use of special graphics or character sets, data compression or expansion, and data encryption or decryption are carried out at this layer. The Presentation layer provides services for the Application layer above it, and uses the Session layer below it. In practice, the Presentation layer rarely appears in pure form, and it is the least well defined of the OSI layers. Application- or Session-layer programs often encompass some or all of the Presentation layer functions.

Session layer

The Session layer is responsible for synchronizing and sequencing the dialog and packets in a network connection. This layer is also responsible for ensuring that the connection is maintained until the transmission is complete, and that the appropriate security measures are taken during a 'session'. The Session layer is used by the Presentation layer above it, and uses the Transport layer below it.

Transport layer

In the OSI reference model, the Transport layer is responsible for providing data transfer at an agreed-upon level of quality, such as at specified transmission speeds and error rates. To ensure delivery, some Transport layer protocols assign sequence numbers to outgoing packets. The Transport layer at the receiving end checks the packet numbers to make sure all have been delivered and to put the packet contents into the proper sequence for the recipient.

The Transport layer provides services for the Session layer above it, and uses the Network layer below it to find a route between source and destination. The Transport layer is crucial in many ways, because it sits between the upper layers, which are strongly application-dependent, and the lower one, which are network-based.

The layers below the Transport layer are collectively known as the 'subnet' layers. Depending on how well (or not) they perform their functions; the Transport layer has to interfere less (or more) in order to maintain a reliable connection.

Network layer

The Network layer is the third layer from the bottom up, or the uppermost 'subnet layer'. It is responsible for the following tasks:

- Determining addresses or translating from hardware to network addresses. These addresses may be on a local network or they may refer to networks located elsewhere on an internetwork.
- Finding a route between a source and a destination node or between two intermediate devices
- Fragmentation of large packets of data into frames small enough to be transmitted by the underlying Data Link layer (fragmentation). The corresponding Network layer at the receiving node undertakes reassembly of the packet

Data Link layer

The Data Link layer is responsible for creating, transmitting, and receiving data packets. It provides services for the various protocols at the Network layer, and uses the Physical layer to transmit or receive material. The Data Link layer creates packets appropriate for the network architecture being used. Requests and data from the Network layer are part of the data in these packets (or frames, as they are often called at this layer). These frames are passed down to the Physical layer from where they are transmitted to the Physical layer on the destination host via the medium. Network architectures (such as Ethernet and Wi-Fi) typically encompass the Physical layer and the lower half of the Data Link layer.

The IEEE 802 networking working groups have refined the Data Link layer into two sub-layers:

- The Logical Link control (LLC) sub-layer in the upper half, implemented as IEEE 802.2 and shared by several networking technologies such as IEEE 802.3 Ethernet, IEEE 802.5 Token Ring and IEEE 802.11Wi-Fi
- The Media Access Control (MAC) sub-layer in the lower half, included with the Physical layer as part of the networking standards mentioned above

The LLC sub-layer provides an interface for the Network layer protocols, and controls the logical communication with its peer at the receiving side. The MAC sub-layer controls physical access to the medium.

Physical layer

The Physical layer is the lowest layer in the OSI. This layer gets data packets from the Data Link layer above it, and converts the contents of these packets into a series of electrical signals that represent ‘0’ and ‘1’ values in a digital transmission. These signals are sent across a transmission medium to the Physical layer at the receiving end. At the destination, the Physical layer converts the electrical signals into a series of bit values. These values are grouped into packets and passed up to the Data Link layer.

The required mechanical and electrical properties of the transmission medium are defined at this level. These include:

- The type of cable and connectors used. The cable may be coaxial, twisted-pair, or fiber optic. The types of connectors depend on the type of cable
- The pin assignments for the cable and connectors. Pin assignments depend on the type of cable and also on the network architecture being used
- The format for the electrical signals. The encoding scheme used to signal ‘0’ and ‘1’ values in a digital transmission or particular values in an analog transmission depend on the network architecture being used

Medium

The medium (shown below the Physical layer) is not part of the Physical layer. Although dictated by a specific Layer 1 implementation, it is specified in a different document. For example, Ethernet dictates Cat5e cable, but the cable itself is specified in TIA/EIA-568B.

1.3 The Client/Server paradigm

The stack is there for the benefit of two entities to talk to each other. These entities are the ‘client’ and the ‘server’. Although they can be accommodated in

physical devices, they are in fact software entities. The client is the requester of information, and the server is the supplier thereof. Examples of clients are email clients (e.g. Eudora, Outlook Express) and Web clients (e.g. IE6, Firefox). Examples of servers are FTP servers (e.g. Serv-U) or Web servers (e.g. Apache, Xitami).

An example from the Industrial environment is Modbus. Here the client is embedded in the Master (i.e. the controller) and the server is located on the Slave device (e.g. RTU).

Figure 1.7

Client/server communication

1.4 General troubleshooting methodology

Experienced individuals will no doubt develop a ‘gut feel’ regarding everyday problems, but for complex problems a definite methodology will help.

Networks normally experience most problems at the level of the medium, and the least problems at the level of the client and server. See this as a triangle with the medium at its base and the client/server at its apex. Therefore, it is best to start at the base and work your way up.

It is easiest to spot problems at the medium. They are often visible to the naked eye (broken cables, flashing LEDs), and require very little network knowledge or tools. Problems at this level can be dealt with by service technicians or electricians with basic cable testing equipment.

At layers 1 and 2 (Ethernet) things get slightly more complex, but problems here can often be solved by swapping interfaces (NICs) or changing configuration parameters via the Windows control panel.

If the problem persists, it might be at layer 3. At this level a solid knowledge of IP configuration parameters such as IP addresses, subnet masks, default gateways, etc. is required. Troubleshooters also need to be familiar with the DOS utilities and/or have a repertoire of Windows-based troubleshooting software at their disposal. The plot thickens if routers are involved, and a basic knowledge of, say, Cisco router commands and utilities may be required.

Layer 3 problems might also be at the protocol level and for that a working knowledge of Network layer protocols such as IP, ARP and ICMP is required, as well as a suitable protocol analyzer. For most applications a freeware analyzer such as Ethereal may suffice, but some high-speed systems such as Gigabit Ethernet and beyond a dedicated hardware analyzer (at appreciable cost) may be necessary.

At layer 4 it becomes even trickier as TCP/UDP has no configurable parameters (except via the registry) and troubleshooters can only view events via a protocol analyzer. A thorough knowledge of the protocol is also required, as problems at this level involve sockets, connection establishment and connection termination.

At OSI layers 5-7 (collectively known as the TCP ‘Application’ layer) we find protocols such as FTP and HTTP. Apart from observing these protocols in action via a protocol analyzer, they are very much ‘black boxes’.

At the client/server level the situation is even worse, as the internal operation is only known to the vendor. Often problems within the client or server only manifest themselves indirectly, for example as spurious TCP resets, and if the situation points at a suspect client or server it can be taken up with the vendor. Often the problem can be addressed by means of a firmware update.

To summarize: start troubleshooting at the bottom and work upwards.