Networking

Terminology

- **TCP/IP** – Transmission Control Protocol
- **IP** – Internet Protocol
- **UDP** – User Datagram Protocol
- **BIND** – Berkeley Internet Name Domain. Name server software.
- **dhcpd** – Dynamic Host Configuration Protocol Daemon

`TCP/IP` is a set of communications protocols that define how different types of computers talk to each other.

The name `TCP/IP` refers to an entire suite of data communications protocols. The suite gets its name from two of the protocols that belong to it: the Transmission Control Protocol `TCP` and the Internet Protocol `IP`. It is also called the Internet Protocol Suite `IPS`.

Today’s Internet is built by commercial providers. National network providers, called tier-one providers, and regional network providers create the infrastructure. Internet Service Providers `ISPs` provide local access and user services. This network of networks is linked together in the United States at several major interconnection points called Network Access Points `NAPs`.

An `internet` (lowercase “i”) is any collection of separate physical networks, interconnected by a common protocol, to form a single logical network.

The `Internet` (uppercase “I”) is the worldwide collection of interconnected networks, which grew out of the original ARPAnet, that uses `IP` to link the various physical networks into a single logical network.

`Intranets` are `TCP/IP` -based enterprise networks that use Internet techniques and web tools to disseminate internal corporate information.

TCP/IP Important Features

- Open protocol standards, freely available and developed independently from any specific computer hardware or operating system. This makes TCP/IP ideal for uniting different hardware and software components, even when not communicating over the Internet.
- Independence from specific physical network hardware. This allows TCP/IP to integrate many different kinds of networks. TCP/IP can be run over an Ethernet, a DSL connection, a dial-up line, an optical network, and virtually any other kind of physical transmission medium.
- A common addressing scheme that allows any TCP/IP device to uniquely address any other device in the entire network, even if the network is as large as the worldwide Internet.
- Standardized high-level protocols for consistent, widely available user services.

Protocol Standards

`Protocols` are formal rules of behavior.

TCP/IP creates a heterogeneous network with open protocols that are independent of operating system and architectural differences.

The open nature of TCP/IP protocols requires an open standards development process and publicly available standards documents.

Internet standards are developed by the Internet Engineering Task Force (IETF) in open, public meetings.

`RFCs` – The protocols developed in this process are published as 'Requests for Comments' (RFCs).

There are three basic types of `RFCs` : standards `STD` , best current practices `BCP` , and informational `FYI` .

Creating an official Internet standard is a rigorous process. Standards track `RFCs` pass through three maturity levels before becoming standards:

- Proposed Standard
- Draft Standard
- Internet Standard

There are two categories of standards: A Technical Specification `TS` defines a protocol. An Applicability Statement `AS` defines when the protocol is to be used.

There are three requirement levels that define the applicability of a standard:

- Required
- Recommended
- Elective

OSI – Open Systems Interconnect Reference Model. An architectural model developed by the International Standards Organization (ISO) that is used to describe the structure and function of data communications protocols.

The OSI Reference Model contains seven layers that define the functions of data communications protocols. These sever layers are often called a **stack** or **protocol stack**.

7 – **Application Layer** consists of application programs that use the network.

6 – **Presentation Layer** standardizes data presentation to the applications.

5 – **Session Layer** manages sessions between applications.

4 – **Transport Layer** provides end-to-end error detection and correction.

3 – **Network Layer** manages connections across the network for the upper layers.

2 – **Data Link Layer** provides reliable data delivery across the physical link.

1 – **Physical Layer** defines the physical characteristics of the network media.

Peers – A `peer` is an implementation of the same protocol in the equivalent layer on a remote system. e.g. the local file transfer protocol is the peer of a remote file transfer protocol.

In the abstract, each protocol is concerned only with communicating to its peers; it does not care about the layers above or below it.

However, there must also be agreement on how to pass data between the layers on a single computer, because every layer is involved in sending data from a local applica- tion to an equivalent remote application.

Data is passed down the stack from one layer to the next until it is transmitted over the network by the Physical Layer protocols.

At the remote end, the data is passed up the stack to the receiving application.

The individual layers do not need to know how the layers above and below them function; they need to know only how to pass data to them.

Although the OSI model is useful, the TCP/IP protocols don’t match its structure exactly.

The Transport Layer in the `OSI` reference model guarantees that the receiver gets the data exactly as it was sent. In `TCP/IP`, this function is performed by the Transmission Control Protocol `TCP`. However, `TCP/IP` offers a second Transport Layer service, User Datagram Protocol `UDP`, that does not perform the end-to-end reliability checks.

The **Internet Protocol** `IP`, which isolates the upper layer protocols from the underlying network and handles the addressing and delivery of data, is usually described as `TCP/IP`’s Network Layer.

TCP/IP Protocol Architecture

TCP/IP is generally viewed as being composed of fewer layers than the seven used in the OSI model. Most descriptions of TCP/IP define three to five functional levels in the protocol architecture.

4 – **Application Layer** consists of applications and processes that use the network.

3 – **Host-to-Host Transport Layer** provides end-to-end data delivery services.

2 – **Internet Layer** defines the datagram and handles the routing of data.

1 – **Network Access Layer** consists of routines for accessing physical networks.

As in the OSI model, data is passed down the stack when it is being sent to the network, and up the stack when it is being received from the network.

Each layer in the stack adds control information to ensure proper delivery. This control information is called a `header` because it is placed in front of the data to be transmitted. Each layer treats all the information it receives from the layer above as data, and places its own header in front of that information. The addition of delivery information at every layer is called encapsulation.

When data is received, the opposite happens. Each layer strips off its **header** before passing the data on to the layer above. As information flows back up the stack, information received from a lower layer is interpreted as both a header and data.

Each layer has its own independent data structures. Conceptually, a layer is unaware of the data structures used by the layers above and below it. In reality, the data structures of a layer are designed to be compatible with the structures used by the surrounding layers for the sake of more efficient data transmission.

Still, each layer has its own data structure and its own terminology to describe that structure.

Applications using **TCP** refer to data as a `stream`, while applications using **UDP** refer to data as a message. **TCP** calls data a `segment`, and **UDP** calls its data a `packet`. The **Internet layer** views all data as blocks called `datagrams`.

Most networks refer to transmitted data as `packets` or `frames`.

A Closer Look at the fucntions of each layer

The Network Access Layer

The protocols in this layer provide the means for the system to deliver data to the other devices on a directly attached network.

This layer defines how to use the network to transmit an IP datagram.

Unlike higher-level protocols, **Network Access Layer** protocols must know the details of the underlying network (its packet structure, addressing, etc.) to correctly format the data being transmitted to comply with the network constraints. The `TCP/IP` **Network Access Layer** can encompass the functions of all three lower layers of the OSI Reference Model (Network, Data Link, and Physical).

The design of `TCP/IP` hides the function of the lower layers, and the better-known protocols (**IP, TCP, UDP, etc.**) are all higher-level protocols.

Functions performed at this level include encapsulation of IP datagrams into the frames transmitted by the network, and mapping of IP addresses to the physical addresses used by the network.

The IP address must be converted into an address that is appropriate for the physical network over which the datagram is transmitted.

Internet Layer

The Internet Protocol `IP` is the most important protocol in this layer.

The release of IP used in the current Internet is IP version 4 `IPv4`, which is defined in RFC 791. `IPv6` is an IP standard that provides greatly expanded addressing capacity. Because `IPv6` uses a completely different address structure, it is not interop-erable with `IPv4`.

The Internet Protocol is the heart of TCP/IP. IP provides the basic packet delivery ser- vice on which TCP/IP networks are built. All protocols, in the layers above and below IP, use the Internet Protocol to deliver data. All incoming and outgoing TCP/IP data flows through IP, regardless of its final destination.

Internet Protocol

The Internet Protocol is the building block of the Internet. Its functions include:

- Defining the datagram, which is the basic unit of transmission in the Internet
- Defining the Internet addressing scheme
- Moving data between the Network Access Layer and the Transport Layer
- Routing datagrams to remote hosts
- Performing fragmentation and re-assembly of datagrams

`IP` is a connectionless protocol. This means that it does not exchange control information (called a “handshake”) to establish an end-to-end connection before transmitting data.

This means that it does not exchange con- trol information (called a “handshake”) to establish an end-to-end connection before transmitting data. In contrast, a connection-oriented protocol exchanges control infor- mation with the remote system to verify that it is ready to receive data before any data is sent. When the handshaking is successful, the systems are said to have estab- lished a connection. The Internet Protocol relies on protocols in other layers to estab- lish the connection if they require connection-oriented service.

IP can be relied upon to accurately deliver your data to the connected network, but it doesn’t check whether that data was correctly received. Protocols in other layers of the TCP/IP architecture provide this checking when it is required.

A packet is a block of data that carries with it the informa- tion necessary to deliver it, similar to a postal letter, which has an address written on its envelope. A packet-switching network uses the addressing information in the pack- ets to switch packets from one physical network to another, moving them toward their final destination. Each packet travels the network independently of any other packet.

The datagram is the packet format defined by the Internet Protocol.