DataFest

Glossary of important topics

This glossary is aimed for non-technical people working with the datafest 2018 dataset to get familiar with the most important concepts in computer networking and internet security.
Most information is taken from simple.wikipedia.com and https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols , feel free to follow the links.

Introduction

These terms will be expanded upon in the appropriate sections that follow:

- **Connection**: In networking, a connection refers to pieces of related information that are transfered through a network. This generally infers that a connection is built before the data transfer (by following the procedures laid out in a protocol) and then is deconstructed at the at the end of the data transfer.
- **Packet**: A packet is, generally speaking, the most basic unit that is transfered over a network. When communicating over a network, packets are the envelopes that carry your data (in pieces) from one end point to the other.

Packets have a header portion that contains information about the packet including the source and destination, timestamps, network hops, etc. The main portion of a packet contains the actual data being transfered. It is sometimes called the body or the payload.

- **Network Interface**: A network interface can refer to any kind of software interface to networking hardware. For instance, if you have two network cards in your computer, you can control and configure each network interface associated with them individually.

A network interface may be associated with a physical device, or it may be a representation of a virtual interface. The "loopback" device, which is a virtual interface to the local machine, is an example of this.

- **LAN**: LAN stands for "local area network". It refers to a network or a portion of a network that is not publicly accessible to the greater internet. A home or office network is an example of a LAN.
- **WAN**: WAN stands for "wide area network". It means a network that is much more extensive than a LAN. While WAN is the relevant term to use to describe large, dispersed networks in general, it is usually meant to mean the internet, as a whole.

If an interface is said to be connected to the WAN, it is generally assumed that it is reachable through the internet.

- **Protocol**: A protocol is a set of rules and standards that basically define a language that devices can use to communicate. There are a great number of protocols in use extensively in networking, and they are often implemented in different layers.

Some low level protocols are TCP, UDP, IP, and ICMP. Some familiar examples of application layer protocols, built on these lower protocols, are HTTP (for accessing web content), SSH, TLS/SSL, and FTP.

- **Port**: A port is an address on a single machine that can be tied to a specific piece of software. It is not a physical interface or location, but it allows your server to be able to communicate using more than one application.

- **Firewall**: Outside of computer security, a *firewall* is simply a wall built to stop (or slow down) the spread of a fire. In terms of computer security, a **firewall** is a piece of software. This software monitors the network traffic. A firewall has a set of rules which are applied to each packet. The rules decide if a packet can pass, or whether it is discarded. Usually a firewall is placed between a network that is trusted, and one that is less trusted. When a large network needs to be protected, the firewall software often runs on a computer that does nothing else. A firewall protects one part of the network against unauthorized access
- **NAT**: NAT stands for network address translation. It is a way to translate requests that are incoming into a routing server to the relevant devices or servers that it knows about in the LAN. This is usually implemented in physical LANs as a way to route requests through one IP address to the necessary backend servers.
- **VPN**: A **virtual private network** or **VPN** for short, is a way of connecting a computer to a remote network. Most people using computers connect to the World Wide Web using a normal network - they use dial-up or broadband. A VPN is a little bit different. It's used by some workers to connect to work using a laptop - they can check their work email and see work websites which can not be seen on the normal internet.[1]VPN often offers anonymity by hiding the user and making it very hard for anyone to track them.[2]Recently VPN is often used to access websites that are blocked in some countries, like China.[3] Many people also use a VPN to protect their internet activity while using public WiFi. While VPN usage varies by country, you can see a full list of reasons why, where, and how often people use a VPN at GlobalWebIndex.[4]

Network Layers

While networking is often discussed in terms of topology in a horizontal way, between hosts, its implementation is layered in a vertical fashion throughout a computer or network.

What this means is that there are multiple technologies and protocols that are built on top of each other in order for communication to function more easily. Each successive, higher layer abstracts the raw data a little bit more, and makes it simpler to use for applications and users.

It also allows you to leverage lower layers in new ways without having to invest the time and energy to develop the protocols and applications that handle those types of traffic.

The language that we use to talk about each of the layering scheme varies significantly depending on which model you use. Regardless of the model used to discuss the layers, the path of data is the same.

As data is sent out of one machine, it begins at the top of the stack and filters downwards. At the lowest level, actual transmission to another machine takes place. At this point, the data travels back up through the layers of the other computer.

Each layer has the ability to add its own "wrapper" around the data that it receives from the adjacent layer, which will help the layers that come after decide what to do with the data when it is passed off.

OSI Model

Historically, one method of talking about the different layers of network communication is the OSI model. OSI stands for Open Systems Interconnect.
This model defines seven separate layers. The layers in this model are:

- **Application**: The application layer is the layer that the users and user-applications most often interact with. Network communication is discussed in terms of availability of resources, partners to communicate with, and data synchronization.
- **Presentation**: The presentation layer is responsible for mapping resources and creating context. It is used to translate lower level networking data into data that applications expect to see.
- **Session**: The session layer is a connection handler. It creates, maintains, and destroys connections between nodes in a persistent way.
- **Transport**: The transport layer is responsible for handing the layers above it a reliable connection. In this context, reliable refers to the ability to verify that a piece of data was received intact at the other end of the connection.

This layer can resend information that has been dropped or corrupted and can acknowledge the receipt of data to remote computers.

- **Network**: The network layer is used to route data between different nodes on the network. It uses addresses to be able to tell which computer to send information to. This layer can also break apart larger messages into smaller chunks to be reassembled on the opposite end.
- **Data Link**: This layer is implemented as a method of establishing and maintaining reliable links between different nodes or devices on a network using existing physical connections.
- **Physical**: The physical layer is responsible for handling the actual physical devices that are used to make a connection. This layer involves the bare software that manages physical connections as well as the hardware itself (like Ethernet).

As you can see, there are many different layers that can be discussed based on their proximity to bare hardware and the functionality that they provide.

TCP/IP Model

The TCP/IP model, more commonly known as the Internet protocol suite, is another layering model that is simpler and has been widely adopted. It defines the four separate layers, some of which overlap with the OSI model:

- **Application**: In this model, the application layer is responsible for creating and transmitting user data between applications. The applications can be on remote systems, and should appear to operate as if locally to the end user.

The communication is said to take place between peers.

- **Transport**: The transport layer is responsible for communication between processes. This level of networking utilizes ports to address different services. It can build up unreliable or reliable connections depending on the type of protocol used.
- **Internet**: The internet layer is used to transport data from node to node in a network. This layer is aware of the endpoints of the connections, but does not worry about the actual connection needed to get from one place to another. IP addresses are defined in this layer as a way of reaching remote systems in an addressable manner.
- **Link**: The link layer implements the actual topology of the local network that allows the internet layer to present an addressable interface. It establishes connections between neighboring nodes to send data.

As you can see, the TCP/IP model, is a bit more abstract and fluid. This made it easier to implement and allowed it to become the dominant way that networking layers are categorized.

Interfaces

Interfaces are networking communication points for your computer. Each interface is associated with a physical or virtual networking device.

Typically, your server will have one configurable network interface for each Ethernet or wireless internet card you have.

In addition, it will define a virtual network interface called the "loopback" or localhost interface. This is used as an interface to connect applications and processes on a single computer to other applications and processes. You can see this referenced as the "lo" interface in many tools.

Many times, administrators configure one interface to service traffic to the internet and another interface for a LAN or private network.

In DigitalOcean, in datacenters with private networking enabled, your VPS will have two networking interfaces (in addition to the local interface). The "eth0" interface will be configured to handle traffic from the internet, while the "eth1" interface will operate to communicate with the private network.

Protocols

Networking works by piggybacking a number of different protocols on top of each other. In this way, one piece of data can be transmitted using multiple protocols encapsulated within one another.

We will talk about some of the more common protocols that you may come across and attempt to explain the difference, as well as give context as to what part of the process they are involved with.

We will start with protocols implemented on the lower networking layers and work our way up to protocols with higher abstraction.

Media Access Control

Media access control is a communications protocol that is used to distinguish specific devices. Each device is supposed to get a unique MAC address during the manufacturing process that differentiates it from every other device on the internet.
Addressing hardware by the MAC address allows you to reference a device by a unique value even when the software on top may change the name for that specific device during operation. Media access control is one of the only protocols from the link layer that you are likely to interact with on a regular basis.

IP

The **Internet Protocol** (**IP**) is the fundamental communications protocol in the Internet protocol suite for relaying data across network boundaries. It essentially establishes the Internet. Historically, IP did not provide the connectivity; It only specified how packets are supposed to be created. The Transmission Control Protocol (TCP) allowed this functionality. Due to this, one could not perform it's task without the other; They go hand in hand, therefore they earned the name **TCP/IP**, as a sign of their dependency on each other.

Think of IP as something like the postal system. It allows you to address a package and drop it into the system, but there is no actual direct *link* between you and the recipient. Instead, there is a "web" of links interconnecting with each other. This is where IP and TCP come in. IP tells packets what their destination is and how to get there; TCP ensures a reliable connection, checking packets for errors, requesting a "re-transmission" if it detects one.

The Internet Protocol gets information from a source computer to a destination computer. It sends this information in the form of packets.

There are two versions of the Internet Protocol currently in use: *IPv4* and *IPv6,* with *IPv4* being the version most used. IP also gives computers an IP address to identify each other, much like a typical physical address.

IP is the primary protocol in the Internet Layer of the Internet Protocol Suite, which is a set of communications protocols consisting of seven abstraction layers (see OSI model),

The main purpose and task of IP is the delivery of datagrams from the source host (source computer) to the destination host (receiving computer) based on their addresses. To achieve this, IP includes methods and structures for putting tags (address information, which is part of metadata) within datagrams. The process of putting these tags on datagrams is called encapsulation.Think of an anology with the postal system. IP is similar to the U.S. Postal System in that it allows a package (a datagram) to be addressed (encapsulation) and put into the system (the Internet) by the sender (source host). However, there is no direct link between sender and receiver.

The package (datagram) is almost always divided into pieces, but each piece contains the address of the receiver (destination host). Eventually, each piece arrives at the receiver, often by different routes and at different times. These routes and times are also determined by the Postal System, which is the IP. However, the Postal System (in the transport and application layers) puts all the pieces back together before delivery to the receiver (destination host).

Note: IP is actually a connectionless protocol, meaning that the circuit to the receiver (destination host) does not need be set up before transmission (by the source host). Continuing the analogy, there does not need to be a direct connection between the physical return address on the letter/package and the recipient address before the letter/package is sent.

Originally, IP was a connectionless datagram service in a transmission control program created by Vint Cerf and Bob Kahn in 1974. When format and rules were applied to allow connections, the connection-oriented Transmission Control Protocol was created. The two together form the Internet Protocol Suite, often referred to as TCP/IP.

Internet Protocol version 4 (IPv4) was the first major version of IP. This is the dominant protocol of the Internet. However, iPv6 is active and in use, and its deployment is increasing all over the world.

Addressing and routing are the most complex aspects of IP. However, intelligence in the network is located at nodes (network interconnection points) in the form of routers which forward datagrams to the next known gateway on the route to the final destination. The routers use interior gateway protocols (IGPs) or external gateway protocols (EGPs) to help with making forwarding route decisions. Routes are determined by the routing prefix within the datagrams. The routing process can therefore become complex. But at the speed of light (or nearly so) the routing intelligence determines the best route, and the datagram pieces and datagram all eventually arrive at their destination

ICMP

ICMP stands for internet control message protocol. It is used to send messages between devices to indicate the availability or error conditions. These packets are used in a variety of network diagnostic tools, such as ping and traceroute.
Usually ICMP packets are transmitted when a packet of a different kind meets some kind of a problem. Basically, they are used as a feedback mechanism for network communications.

TCP

The **Transmission Control Protocol (TCP)** is one of the main protocols of the Internet Protocol Suite. TCP is part of the popular "TCP/IP" combination used by the Internet. The Internet Protocol, or IP, makes sure data on the internet gets to the right place. Then TCP makes sure the data is put in the right order, and none of it is missing. TCP also helps to control traffic on the internet so it does not get overloaded. These protocols, which are kind of like languages that computers use, are designed so that any computer, and any program (such as a Web browser or e-mail client), can use them.

UDP

UDP stands for user datagram protocol. It is a popular companion protocol to TCP and is also implemented in the transport layer.
The fundamental difference between UDP and TCP is that UDP offers unreliable data transfer. It does not verify that data has been received on the other end of the connection. This might sound like a bad thing, and for many purposes, it is. However, it is also extremely important for some functions.
Because it is not required to wait for confirmation that the data was received and forced to resend data, UDP is much faster than TCP. It does not establish a connection with the remote host, it simply fires off the data to that host and doesn't care if it is accepted or not.
Because it is a simple transaction, it is useful for simple communications like querying for network resources. It also doesn't maintain a state, which makes it great for transmitting data from one machine to many real-time clients. This makes it ideal for VOIP, games, and other applications that cannot afford delays.

HTTP

HTTP stands for hypertext transfer protocol. It is a protocol defined in the application layer that forms the basis for communication on the web.
HTTP defines a number of functions that tell the remote system what you are requesting. For instance, GET, POST, and DELETE all interact with the requested data in a different way. It is used to send and receive webpages and files on the internet. It was developed by Tim Berners-Lee and is now coordinated by the W3C. HTTP version 1.1 is the most common used version today. It is defined in RFC 2616.

HTTP works by using a user agent to connect to a server. The user agent could be a web browser or spider. The server must be located using a URLor URI. This always contains http:// at the start. It normally connects to port 80 on a computer.

A more secure version of HTTP is called HTTPS (Hypertext Transfer Protocol Secure). This contains https:// at the beginning of the URL. It encrypts all the information that is sent and received. This can stop malicious users such as hackers from stealing the information and is often used on payment websites. HTTPS uses port 443 for communication instead of port 80.

FTP

FTP stands for file transfer protocol. It is also in the application layer and provides a way of transferring complete files from one host to another.
It is inherently insecure, so it is not recommended for any externally facing network unless it is implemented as a public, download-only resource.

DNS

The **Domain Name System** (**DNS**) is a system used to convert a computer's host name into an IP address on the Internet. For example, if a computer needs to communicate with the web server *example.net*, your computer needs the IP address of the web server *example.net*. It is the job of the DNS to convert the host name to the IP address of the web server. The DNS uses the UDP Port 53 or TCP Port 53. It is defined mainly by RFC 1034 and RFC 1035. There are later RFC which define changes to the system.

SSH

SSH stands for secure shell. It is an encrypted protocol implemented in the application layer that can be used to communicate with a remote server in a secure way. Many additional technologies are built around this protocol because of its end-to-end encryption and ubiquity. There are many other protocols that we haven't covered that are equally important. However, this should give you a good overview of some of the fundamental technologies that make the internet and networking possible.

SSL

See TLS

TLS

Transport Layer Security (**TLS**) Protocol and its predecessor, **Secure Sockets Layer** (**SSL**), are cryptographic protocols that provide security and data integrity for communications over TCP/IP networks such as the Internet. Several versions of the protocols are common in applications such as web browsing, electronic mail, Internet faxing, instant messaging and voice-over-IP (VoIP).
The TLS protocol allows applications to communicate across a network in a way designed to prevent eavesdropping, tampering, and message forgery. TLS provides endpoint authentication and communications confidentiality over the Internet using cryptography. Most of the time, only the server is authenticated (*i.e.*, its identity is ensured) while the client remains unauthenticated; this means that the end user (whether an individual or an application, such as a Web browser) can be sure with whom it is communicating. The next level of security is known as mutual authentication. Mutual authentication requires public key infrastructure (PKI) deployment to

clients unless TLS-PSK or the Secure Remote Password (SRP) protocol are used, which provide strong mutual authentication without needing to deploy a PKI.

X509

In cryptography, X.509 is a common "PKI" (public key infrastructure) used to manage digital certificates and public-key encryption and a key part of the Transport Layer Security protocol used to secure web and email communication. X.509 specifies ways in which public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithms are formatted.

SNMP

Simple Network Management Protocol (**SNMP**) is a part of the Internet Protocol Suite. SNMP is used in network management systems to monitorstatus of devices and also spot problems. It consists of a set of standards for network management, including an Application Layer protocol, a database schema, and a set of data objects.[1]

SNMP exposes management data in the form of variables on the managed systems, which describe the system configuration. These variables can then be queried (and sometimes set) by managing applications. It is a standard for managing devices such as routers,switches,servers etc.

In common SNMP usage, there are a number of systems to be managed, and one or more systems managing them. A software component called an *agent* (see below) runs on each managed system and reports information via SNMP to the managing systems.

SNMP agents expose management data on the managed systems as variables (such as "free memory", "system name", "number of running processes", "default route"). But the protocol also allows active management tasks, such as modifying and applying a new configuration. The managing system can retrieve the information through the **GET**, **GETNEXT** and **GETBULK** protocol operations or the agent will send data without being asked using **TRAP** or **INFORM** protocol operations. Management systems can also send configuration updates or controlling requests through the **SET** protocol operation to actively manage a system. Configuration and control operations are used only when changes are needed to the network. The monitoring operations are usually performed regularly.

FTP

FTP, also known as **File Transfer Protocol**, is a communication protocol for the rapid, simple transmission of files across a network supporting the TCP/IP protocol. This network is generally the Internet, or a local network. FTP is a way of accessing files on another computer. FTP uses the Client-Server architecture, meaning that there is a server, that holds the files, and does the authentication, and a client, or the end-user, who is accessing the files. The server listens on the network for connection requests from other computers. The client can make a connection to the FTP server by using FTP client software. Once connected and authenticated (via rsh or SFTP) the client can do things such as uploading files to the server, downloading files (taking the server's files and putting them on his own computer) from the server, and renaming, deleting files on the server, changing file permissions, etc.

Most modern Operating Systems support FTP. This implies that any computer connected to a TCP/IP based network can manipulate files on another computer on that network regardless of which operating systems are involved, provided that they are open to FTP connections. There are many existing FTP client and server programs, many of these are available free, or open source.

FTP connection is also seen in cellular phones when trying to transfer or receive data from a computer nearby.

FTP server return codes show their status by the digits within them. A short explanation of various digits' meanings are given below:

- 1xx: Positive Preliminary reply. The action requested is being initiated but there will be another reply before it begins.
- 2xx: Positive Completion reply. The action requested has been completed. The client may now issue a new command.
- 3xx: Positive Intermediate reply. The command was successful, but a further command is required before the server can act upon the request.
- 4xx: Transient Negative Completion reply. The command was not successful, but the client is free to try the command again as the failure is only temporary.
- 5xx: Permanent Negative Completion reply. The command was not successful and the client should not attempt to repeat it again.
- x0x: The failure was due to a syntax error.
- x1x: This response is a reply to a request for information.
- x2x: This response is a reply relating to connection information.
- x3x: This response is a reply relating to accounting and authorization.
- x4x: Not used.
- x5x: These responses indicate the status of the Server file system vis-a-vis the requested transfer or other file system action.

FTP may run in *active* or *passive* mode, which determines how the data connection is established. In both cases, the client creates a TCP control connection from a random, usually an unprivileged, port N to the FTP server command port 21.

- In *active* mode, the client starts listening for incoming data connections from the server on port M. It sends the FTP command PORT M to inform the server on which port it is listening. The server then initiates a data channel to the client from its port 20, the FTP server data port.
- In situations where the client is behind a firewall and unable to accept incoming TCP connections, *passive* mode may be used. In this mode, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server, which the client then uses to open a data connection from an arbitrary client port to the server IP address and server port number received.