TCP\/IP Protocol Fundamentals Explained with a Diagram

s905060.gitbooks.io/site-reliability-engineer-handbook/content/tcpip_protocol_fundamentals_explained_with_a_diagram.html

- logrotate
- PIDOF
- Dmidecode
- lshw
- printenv
- SS
- w
- Strace
- pstree
- USERMOD
- ltrace
- ethtool
- IP
- Sar
- nethogs
- zip
- FPM
- getent
- ipmitool
- Building RPMs
- Megacli
  - Megacli package version
- RKhunter
- fping
- blkid
- FSCK
- Package Manager
- mktemp
- ls
- Comm
- taskset
- fio
- tree
- ARP
- lsblk

- How-To
  - CentOS: nf_conntrack: table full, dropping packet
  - How To Fix “Error: database disk image is malformed” On CentOS \/ Fedora
  - Finding the PID of the process using a specific port?
  - How-To create hashed SSH password
  - How to display and kill zombie processes

- Shell command to bulk change file extensions in a directory (Linux)
- 8 Powerful Awk Built-in Variables – FS, OFS, RS, ORS, NR, NF, FILENAME, FNR
- Changing the Time Zone
- HOW DO I DISABLE SSH LOGIN FOR THE ROOT USER?
- How-To rename the extension for a batch of files?
- How-To disable IPv6 on RHEL6 \/ CentOS 6 \/ etc
- How to clear the ARP cache on Linux?
- How-To crontab running as a specific user
- Ansible – exclude host from playbook execution
- HOWTO: Use Wireshark over SSH
- How-To Change Network Interface Name
- How-To Creating a Partition Size Larger Than 2TB
- Hot-To Linux Hard Disk Format Command
- Hadoop Troubleshooting
- Hive Troubleshooting
- HowTo Set up hostbased authentication for passphraseless SSH communication.
- Difference between a cold and warm reboot
- ls -l explained
- df falsely showing 100 per cent disk usage
- FSCK explained
- Manually generate password for \/etc\/shadow
- How To Change Timezone on a CentOS 6 and 7
- Setting ssh private key forwarding
- Persist keys in ssh-agent on OS X
- SSH Essentials: Working with SSH Servers, Clients, and Keys
- How to Change JVM Heap Setting (-Xms -Xmx) of Tomcat – Configure setenv.sh file – Run catalina.sh
- SSH ProxyCommand example: Going through one host to reach another server
- How to get Linux's TCP state statistics
- Linux TCP retransmission rate calculation
- How to determine OOM
- How-to check Java process heapsize
- Troubleshooting network issues
- How to check what sudo acces a user has?
- How to copy your key to a remote server?
- Linux date and Unix timstamp conversion
- SSH client personalized configuration
- How to Error Detection and Correction
- How To Kerberos

- How to identify defective DIMM from EDAC error on Linux
    - Howto Install and Configure Cobbler on Centos 6
    - How To Use GPG to Encrypt and Sign Messages on an Ubuntu 12.04 VPS
    - HowTo: Debug Crashed Linux Application Core Files Like A Pro
    - Create init script in CentOS 6
    - Linux Change Disk Label Name on EXT2 \/ EXT3 \/ EXT4 File Systems
    - How to retrieve and change partition's UUID Universally Unique Identifier on linux
    - Using Text-Mode Serial Console Redirection
    - How to Write Linux Init Scripts Based on LSB Init Standard
    - How to create a Debian package
    - How to create a RPM Package
    - How to solve EDAC DIMM CE Error
    - How to solve fsck.ext4: Unable to resolve UUID\/LABEL
    - How to expand an existing LSI raid array using MegaCli
    - How to change user GID and UID in Ubuntu
    - How to read a segfault kernel log message
    - How to add cron job via command line
    - How to restrict process CPU usage using nice, cpulimit, and cgroups
- Data Structure
- Service
    - Cloud-Init
    - ETCD
    - RESTful API HTTP methods
    - Web cache
    - Mesos
    - ELK
    - Cassandra
    - Hive
        Hive notes
    - Elasticsearch
    - Scylla
    - Zookeeper
- Automation Tool
    - Ansible
    - Salt
        Salt use notes
- Networking Devices
    - Cisco
    - Juniper

TCP/IP Protocol Fundamentals Explained with a Diagram

Have you ever wondered how your computer talks to other computers on your local LAN or to other systems on the internet?

Understanding the intricacies of how computers interact is an important part of networking and is of equal interest to a sysadmin as well as to a developer. In this article, we will make an attempt to discuss the concept of communication from the very basic fundamental level that needs to be understood by everybody.

TCP/IP PROTOCOL SUITE

Communications between computers on a network is done through protocol suits. The most widely used and most widely available protocol suite is TCP/IP protocol suite. A protocol suit consists of a layered architecture where each layer depicts some functionality which can be carried out by a protocol. Each layer usually has more than one protocol options to carry out the responsibility that the layer adheres to. TCP/IP is normally considered to be a 4 layer system. The 4 layers are as follows :

- Application layer
- Transport layer
- Network layer
- Data link layer

1. Application layer

This is the top layer of TCP/IP protocol suite. This layer includes applications or processes that use transport layer protocols to deliver the data to destination computers.

At each layer there are certain protocol options to carry out the task designated to that particular layer. So, application layer also has various protocols that applications use to communicate with the second layer, the transport layer. Some of the popular application layer protocols are :

- HTTP (Hypertext transfer protocol)
- FTP (File transfer protocol)
- SMTP (Simple mail transfer protocol)
- SNMP (Simple network management protocol) etc

2. Transport Layer

This layer provides backbone to data flow between two hosts. This layer receives data from the application layer above it. There are many protocols that work at this layer but the two most commonly used protocols at transport layer are TCP and UDP.

TCP is used where a reliable connection is required while UDP is used in case of unreliable connections.

TCP divides the data(coming from the application layer) into proper sized chunks and then passes these chunks onto the network. It acknowledges received packets, waits for the acknowledgments of the packets it sent and sets timeout to resend the packets if acknowledgements are not received in time. The term 'reliable connection' is used where it is not desired to loose any information that is being transferred over the network through this connection. So, the protocol used for this type of connection must provide the mechanism to achieve this desired characteristic. For example, while downloading a file, it is not desired to loose any information(bytes) as it may lead to corruption of downloaded content.

UDP provides a comparatively simpler but unreliable service by sending packets from one host to another. UDP does not take any extra measures to ensure that the data sent is received by the target host or not. The term 'unreliable connection' are used where loss of some information does not hamper the task being fulfilled through this connection. For example while streaming a video, loss of few bytes of information due to some reason is acceptable as this does not harm the user experience much.

3. Network Layer

This layer is also known as Internet layer. The main purpose of this layer is to organize or handle the movement of data on network. By movement of data, we generally mean routing of data over the network. The main protocol used at this layer is IP. While ICMP(used by popular 'ping' command) and IGMP are also used at this layer.

4. Data Link Layer

This layer is also known as network interface layer. This layer normally consists of device drivers in the OS and the network interface card attached to the system. Both the device drivers and the network interface card take care of the communication details with the media being used to transfer the data over the network. In most of the cases, this media is in the form of cables. Some of the famous protocols that are used at this layer include ARP(Address resolution protocol), PPP(Point to point protocol) etc.

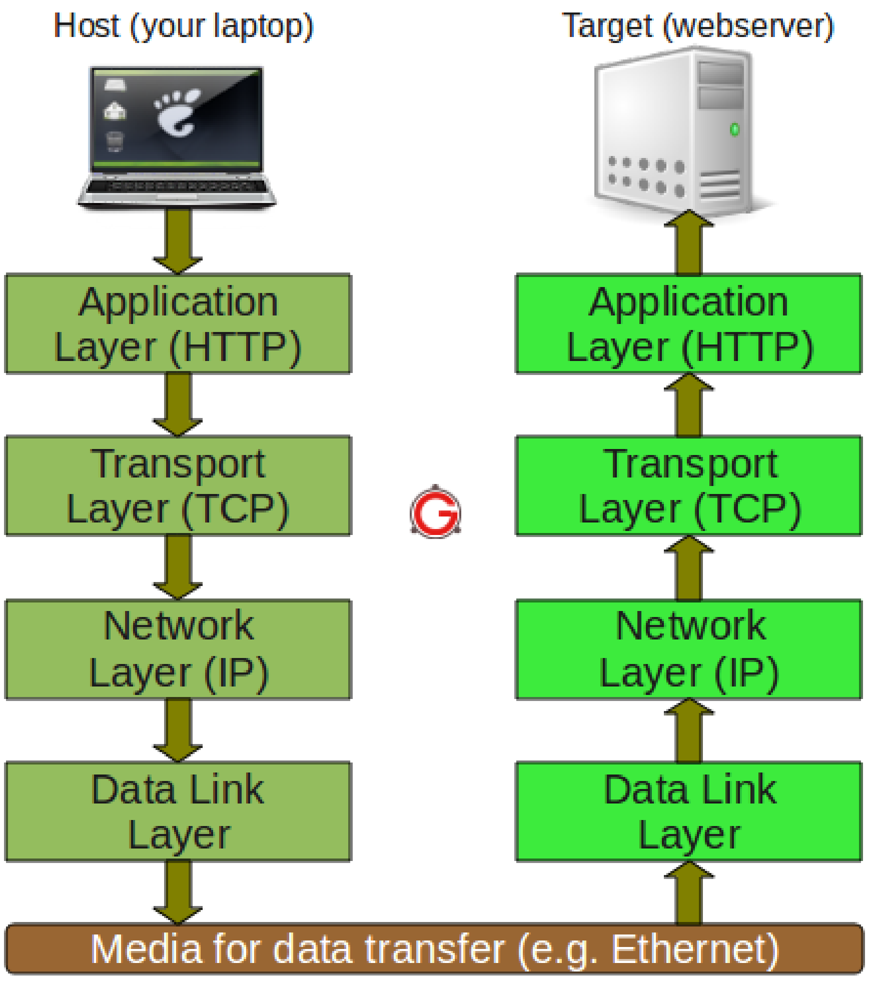
TCP/IP CONCEPT EXAMPLE

One thing which is worth taking note is that the interaction between two computers over the network through TCP/IP protocol suite takes place in the form of a client server architecture.

Client requests for a service while the server processes the request for client.

Now, since we have discussed the underlying layers which help that data flow from host to target over a network. Lets take a very simple example to make the concept more clear.

Consider the data flow when you open a website.



As seen in the above figure, the information flows downward through each layer on the host machine. At the first layer, since http protocol is being used, so an HTTP request is formed and sent to the transport layer.

Here the protocol TCP assigns some more information(like sequence number, source port number, destination port number etc) to the data coming from upper layer so that the communication remains reliable i.e, a track of sent data and received data could be maintained.

At the next lower layer, IP adds its own information over the data coming from transport layer. This information would help in packet travelling over the network. Lastly, the data link layer makes sure that the data transfer to/from the physical media is done properly. Here again the communication done at the data link layer can be reliable or unreliable.

This information travels on the physical media (like Ethernet) and reaches the target machine.

Now, at the target machine (which in our case is the machine at which the website is hosted) the same series of interactions happen, but in reverse order.

The packet is first received at the data link layer. At this layer the information (that was stuffed by the data link layer protocol of the host machine) is read and rest of the data is passed to the upper layer.

Similarly at the Network layer, the information set by the Network layer protocol of host machine is read and rest of the information is passed on the next upper layer. Same happens at the transport layer and finally the HTTP request sent by the host application(your browser) is received by the target application(Website server).

One would wonder what happens when information particular to each layer is read by the corresponding protocols at target machine or why is it required? Well, lets understand this by an example of TCP protocol present at transport layer. At the host machine this protocol adds information like sequence number to each packet sent by this layer.

At the target machine, when packet reaches at this layer, the TCP at this layer makes note of the sequence number of the packet and sends an acknowledgement (which is received seq number + 1).

Now, if the host TCP does not receive the acknowledgement within some specified time, it re sends the same packet. So this way TCP makes sure that no packet gets lost. So we see that protocol at every layer reads the information set by its counterpart to achieve the functionality of the layer it represents.

PORTS, SERVERS AND STANDARDS

On a particular machine, a port number coupled with the IP address of the machine is known as a socket. A combination of IP and port on both client and server is known as four tuple. This four tuple uniquely identifies a connection. In this section we will discuss how port numbers are chosen.

You already know that some of the very common services like FTP, telnet etc run on well known port numbers. While FTP server runs on port 21, Telent server runs on port 23. So, we see that some standard services that are provided by any implementation of TCP/IP have some standard ports on which they run. These standard port numbers are generally chosen from 1 to 1023. The well known ports are managed by Internet Assigned Numbers Authority(IANA).

While most standard servers (that are provided by the implementation of TCP/IP suite) run on standard port numbers, clients do not require any standard port to run on.

Client port numbers are known as ephemeral ports. By ephemeral we mean short lived. This is because a client may connect to server, do its work and then disconnect. So we used the term 'short lived' and hence no standard ports are required for them.

Also, since clients need to know the port numbers of the servers to connect to them, so most standard servers run on standard port numbers.

The ports reserved for clients generally range from 1024 to 5000. Port number higher than 5000 are reserved for those servers which are not standard or well known.

If we look at the file '/etc/services', you will find most of the standard servers and the port on which they run.

```
$ cat /etc/services
systat        11/tcp        users
daytime        13/udp
netstat        15/tcp
qotd        17/tcp        quote
msp          18/udp
chargen        19/udp        ttytst source
ftp-data    20/tcp
ftp           21/tcp
ssh           22/tcp
ssh           22/udp
telnet        23/tcp
...
...
...
```

As you see from the /etc/services file, FTP has port number 21, telent has port number 23 etc. You can use 'grep' command on this file to find any server and its associated port.

As far as the standards are concerned, the following four organizations/groups manage the TCP/IP protocol suite. Both the IRTF and the IETF fall under the IAB.