TCP Connection Termination

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

        - SimplePrograms
        - 'all', 'any' are Python built-ins
        - TemporaryFile
        - How to capture stdout in real-time with Python
        - Python simple techniques and common reference
        - python reference fragments
        - getpass
        - Method overriding in Python
        - Multiple levels of 'collection.defaultdict' in Python
        - String Format
        - Logging
        - Convert Unicode Object to Python Dict
        - The dir( ) Function
        - Python dictionary has_key() Method
        - glob – Filename pattern matching
        - Lambda, filter, reduce and map
        - doctest – Testing through documentation
        - Load Python code dynamically
        - Map, Reduce, Zip, Filter
        - DICTIONARY COMPREHENSION
- Linux Command Line Tool
    - Basic
    - DIFF
    - AC
    - AWK
    - CHMOD
    - NMAP
    - NETSTAT
    - Flock
    - Traceroute
    - FIND
    - GREP
    - Crontab
    - Kill
    - SED
    - CUT
    - CURL
    - IFCONFIG
    - TCPDUMP
    - TAR
    - LSOF

- SORT
- Xargs
- Iptables
- xargs vs. exec {}
- Hdparm
- UNIQ
- STAT
- Execute Commands in the Background
- TAIL
- WGET
- Date
- FDISK
- Mount
- Make SWAP File
- Create a New User
- Create a New Group
- Setup SSH Passwordless Login in OpenSSH
- Parted
- RSYNC
- YUM
- RPM
- APT
- Install from Source
- Log Rotate
- FREE
- DF
- DU
- Sysctl
- NICE
- Renice
- PS
- DD
- BC
- LDD
- getcap, setcap and file capabilities
- Linux_Basename
- PMAP
- Alternative
- Readlink
- 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
- Storage
    - Object Storage
    - How an object store differs from file and block storage
- Monitoring
    - Nagios
    - Zabbix
    - Graphite
        - The architecture of clustering Graphite
- Database
- Algorithm
    - Insertion Sort
    - Hill Sort
    - Bubble Sort
    - Quick Sort
    - Directly Select Sort
    - Heap Sort
    - Merge Sort
    - Radix Sort
    - Cache algorithm definition
- Software Engineering
- Data Structure

Normal Connection Termination

In the normal case, each side terminates its end of the connection by sending a special message with the FIN (finish) bit set. This message, sometimes called a FIN, serves as a connection termination request to the other device, while also possibly carrying data like a regular segment. The device receiving the FIN responds with an acknowledgment to the FIN to indicate that it was received. The connection as a whole is not considered terminated until both sides have finished the shut down procedure by sending a FIN and receiving an ACK.

Thus, termination isn't a three-way handshake like establishment: it is a pair of two-way handshakes. The states that the two devices in the connection move through during a normal connection shutdown are different because the device initiating the shutdown must behave differently than the one that receives the termination request. In particular, the TCP on the device receiving the initial termination request must inform its application process and wait for a signal that the process is ready to proceed. The initiating device doesn't need to do this, since the application is what started the ball rolling in the first place.

> Key Concept: A TCP connection is normally terminating using a special procedure where each side independently closes its end of the link. It normally begins with one of the application processes signalling to its TCP layer that the session is no longer needed. That device sends a FIN message to tell the other device that it wants to end the connection, which is acknowledged. When the responding device is ready, it too sends a FIN that is acknowledged; after waiting a period of time for the ACK to be received, the session is closed.

Table 154 describes in detail how the connection termination process works; the progression of states and messages exchanged can also be seen in Figure 214. The table is adapted from Table 151, describing the TCP finite state machine, but shows what happens for both the server and the client over time during connection shutdown. Either device can initiate connection termination; in this example I am assuming the client does it. Each row shows the state each device begins in, what action it takes in that state and what state to which it transitions. I have also shown the send and receive stages of both of the steps for each of the client and server’s close operations.

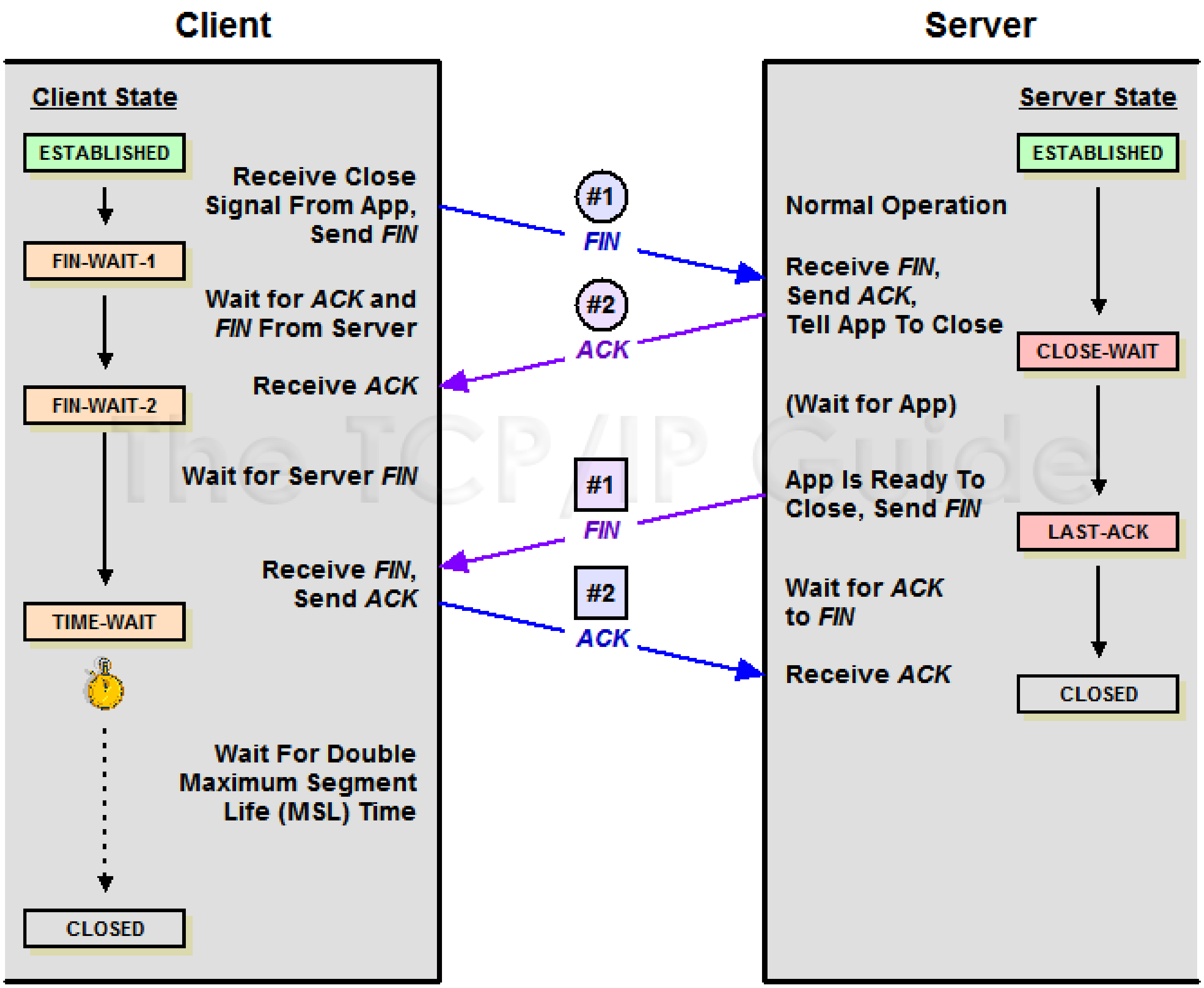

Figure 214: TCP Connection Termination Procedure

This diagram shows the conventional termination procedure for a TCP session, with one device initiating termination and the other responding. In this case the client initiates; it sends a FIN which is acknowledged by the server. The server waits for the server process to be ready to close and then sends its FIN, which is acknowledged by the client. The client waits for a period of time to ensure that its ACK is received, before proceeding to the CLOSED state.