# **43. FTP and TFTP**

## **The Purpose of FTP / TFTP**

* **FTP (File Transfer Protocol)** and **TFTP (Trivial File Transfer Protocol)** are industry-standard protocols used to transfer files over a network.
* Both use a **client-server model**:
  + Clients can use FTP / TFTP to **copy files from and to a server**.
* As a **Network Engineer**, the most common use for FTP / TFTP is **upgrading the operating system** of a network device.
* You can use FTP / TFTP to **download** the newer version of **IOS** from a server and then **reboot** the device with the new IOS image.

## **TFTP and FTP Functions and Differences**

### **TFTP**

* **Standardized in 1981**.
* Named "Trivial" because it’s **simple** and has only basic features compared to FTP:
  + Only allows a client to **copy files to/from a server**.
* Was released after FTP, but **not a replacement** for FTP:
  + It’s another tool to use when **lightweight simplicity** is more important than **functionality**.
* **No authentication** (Username/Password) – servers respond to **all TFTP requests**.
* **No encryption** – all data is sent **in plain text**.
* Best used in a **controlled environment** to transfer **small files quickly**.
* **TFTP servers listen on UDP port 69**.
* **UDP is connectionless** and doesn’t provide **reliability with retransmissions**.
* However, TFTP has **built-in reliability features** within the protocol itself.

### **TFTP Reliability**

* Every TFTP **data message is acknowledged**:
  + If the **client** transfers a file to the **server**, the **server** sends **ACK messages**.
  + If the **server** transfers a file to the **client**, the **client** sends **ACK messages**.
* **Timers** are used; if an **expected message** isn’t received in time, the waiting device will **resend its previous message**.

### **TFTP "Connections"**

* When the **client** sends the first message to the **server**:
  + The **destination port** is **UDP 69**.
  + The **source port** is a **random ephemeral port** (Transfer Identifier - TID).
* The **server** then selects a **random TID** as its **source port** when replying **(not UDP 69)**.
* When the **client** sends the next message, the **destination port** will be the **server’s TID**, not UDP 69.

## **FTP**

* **Standardized in 1971**.
* **Uses TCP ports 20 and 21**.
* Requires **username and password for authentication** but has **no encryption**.
* For **greater security**:
  + **FTPS** (FTP over SSL/TLS) can be used.
  + **SFTP** (SSH File Transfer Protocol) can also be used.
* FTP is **more complex** than TFTP and allows **more functions**:
  + **Navigate file directories**.
  + **Add/remove files**.
  + **List files**, etc.
* The **client** sends **FTP commands** to the **server** to perform these functions.

### **FTP Control Connections**

* FTP uses **two types of connections**:
  + An **FTP control connection** (TCP 21) is established for **sending commands and replies**.
  + When **files/data** are transferred, separate **FTP data connections** (TCP 20) are established **as needed**.

### **Active Mode FTP Data Connections**

* The **default method** of establishing **FTP data connections** is **Active Mode**, where the **server initiates** the TCP connection.
* In **Passive Mode**, the **client initiates** the data connection:
  + This is useful when the client is **behind a firewall**, which may block incoming connections from the server.

## **FTP vs. TFTP**

| **Feature** | **FTP** | **TFTP** |
| --- | --- | --- |
| Protocol Type | TCP | UDP |
| Ports Used | 20 (Data), 21 (Control) | 69 (Initial Request) |
| Authentication | Yes (Username/Password) | No |
| Encryption | No (Use FTPS/SFTP for security) | No |
| Functions | File transfer, directory navigation, file management | Simple file transfer only |
| Reliability | TCP ensures delivery and retransmissions | UDP is connectionless, but TFTP handles acknowledgments |
| Best Used For | Secure file transfers and directory management | Fast, lightweight file transfers in controlled environments |

## **IOS File Systems**

* A **file system** is a way of controlling how **data is stored and retrieved**.

You can view the **file system** of a **Cisco IOS device** with:  
Router# show file systems

## **Using FTP / TFTP in IOS**

You can view the **current version of IOS** with:  
 Router#show version

You can view the **contents of flash storage** with:  
 Router#show flash

## **Copying Files with TFTP**

### **Step 1: Start TFTP Transfer**

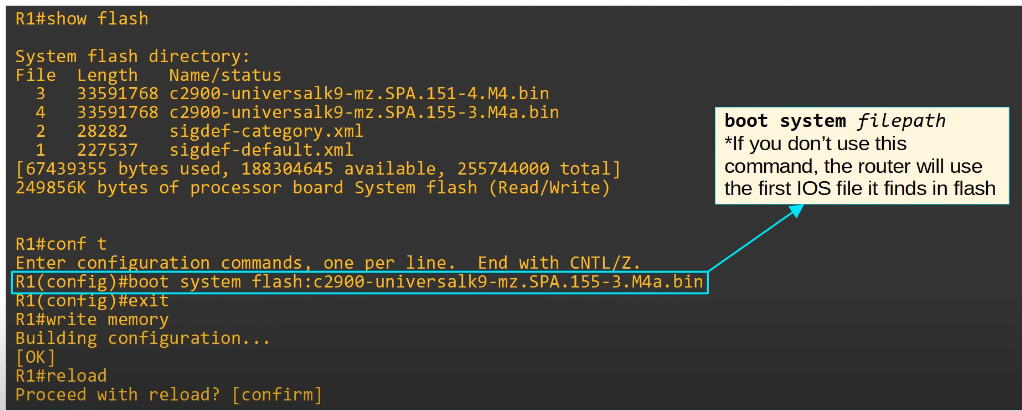
Initiate TFTP file transfer command:  
R1# copy tftp: flash:

### **Step 2: Enter Server Details**

* Enter the **TFTP server address**.

### **Step 3: Enter Filename**

* Enter the **file name to be transferred**.



## **Copying Files with FTP**

### **Step 1: Start FTP Transfer**

Initiate FTP file transfer command:  
 copy ftp flash

### **Steps 2 and 3: Identical to TFTP above**

## **Command Summary**

| **Command** | **Description** |
| --- | --- |
| show file systems | Displays available file systems on the device |
| show version | Displays current IOS version |
| show flash | Lists files stored in flash memory |
| copy tftp flash | Copies a file from a TFTP server to flash memory |
| copy ftp flash | Copies a file from an FTP server to flash memory |