Fall 2018 CS120 Project 4. FTP

Handout Date: Nov. 29, 2018 Due Date: Jan. 16, 2019

(6 points + 6 points)

Please read the following instructions carefully:

- This project is to be completed by each group **individually**.
- Submit your code through Blackboard. The submission is performed by one of the group members.
- Each group needs to submit the code **once and only once**. Immediately after TAs' checking.

Overview

The goal of this project is to enrich Athernet Nodes's applications by providing the File Transfer Protocol (FTP) service. We are going to implement an FTP client for Athernet Node. After this project, Athernet Node is able to connect to public FTP servers.

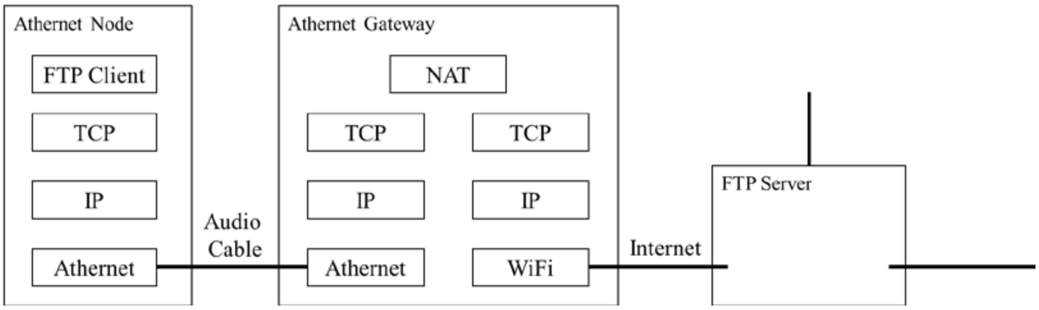

Figure 1 Project4 Overview

FTP is a protocol targeting file transfer between clients and servers. We have finished many tasks transferring small files in previous projects. FTP is different from these simple file transfer examples in that it integrates the network transmission and the local file system. Through FTP, remote file transfer is platform-independent and easy to use.

Part 1. (6 points) FTP Client for Athernet

Unlike previous projects, FTP is a well-defined, yet quite simple protocol. Its RFC [1] is the most accurate and detailed description for guiding the implementation, thus this project description only contains a necessary introduction. As we are not going to implement an FTP server, the client part (`USER-FTP`) is the main focus.

At a high level, the design philosophy of FTP is simple and direct. The client uses one TCP connection (to server's port 21) to send/receive control commands/replies, and uses one or more other TCP connections to receive/upload file data. In other words, FTP's control and data planes are decoupled.

There are several FTP control commands, such as `USER`, `PASS`, `PWD`, `CWD`, `PASV`, `LIST`, `RETR`, `STOR`, etc. These commands and their responses are text-based and very similar to those that used in Telnet and HTTP. Some of the commands (e.g., `LIST`: list current directory, `RETR`: download data, `STOR`: upload data) can initiate data transmission. Once these commands are executed, the client and server must negotiate new TCP connections to hold the data transmission. There are two methods to negotiate new TCP connections: Passive Mode (server opens a port and tells the client to connect) and Active Mode (server uses port 20 to connect the port chosen by the client). Since the Athernet Node must pass through NAT to reach the Internet, Passive Mode is the only choice if you are not going implement a stateful NAT (stateful NAT is able to recognize/remember/modify FTP packets and open correct ports for translating connections initiated by the server in Active Mode).

FTP is based on TCP, but a full TCP stack is very complicated. Thus, in order to simplify the implementation of FTP client, a fake TCP stack in Athernet only keeps necessary components in TCP: connection establishment, connection termination, and a simplified sliding window. For connection establishment and connection termination, you only need to consider the common case in the TCP state machine (see lectures). The sliding window can be simplified into a "stop-and-wait" protocol if the Athernet Node replies TCP acks like a "stop-and-wait" protocol.

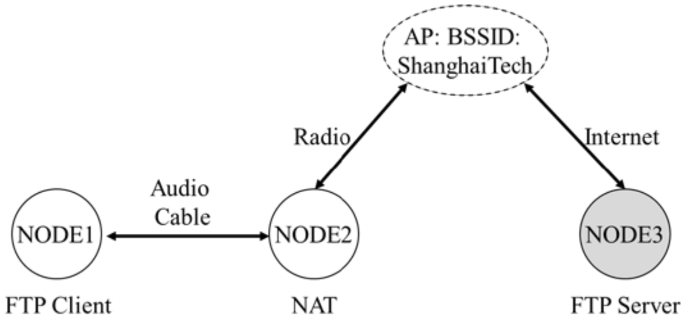

Figure 2 Network Topology for FTP Application

Checkpoints:

The Network Topology is shown in Figure 2

The group provides two devices: NODE1 and NODE2.

NODE3 is a public FTP Server, which supports anonymous login. The list of suggested test FTP Servers is given in "serverlist.txt".

CK (6 points).
NODE1: the FTP client is connected to NODE3.
NODE2: NAT

This task is graded according to the number of supported control commands of the FTP client. The client provides an interface for TAs to input commands and displays corresponding responses from the FTP server. The FTP client must be able to be tolerant of various or even incorrect input and display correct responses.
In order to obtain corresponding credits, the client must support a full set of commands from one of the sets in the following table.
For `RETR` command, the downloading file is selected from FTP server by TAs and its file size is less than 1MB.

Control Commands	
`USER PASS PWD`	-50%
`USER PASS PWD CWD PASV LIST`	-17%
`USER PASS PWD CWD PASV LIST RETR`	-0%

Tips:
a. You may want to use Wireshark with proper filter to trace the working process of FTP.
b. You can use existing FTP client to see how they handle the input.

Part 2. (0 point) Every End is a New Beginning

Return what you borrowed to TAs after finishing checking Project4. Ensure your borrowed accessories are in good condition, as they will be reused in future semesters. TAs will provide a handy plastic bag for each group to pack up the returned accessories. The last step is to sign off at the record card in the plastic bag.

(Tasks with "Optional" tag are optional tasks. The instructor is responsible for checking and grading the optional tasks. Contact the instructor to check Part 3 on/before Jan. 16. The due date of Part 4 is Jan. 18.)

Part 3. (Optional + 2 points) Athernet Tunnel

FTP is a classic network protocol. Many FTP client and server tools/programs exist. So the second way to enable FTP client on Athernet Nodes is to reuse existing ones (e.g. FileZilla client, ftp in linux, etc.). The problem is that the default network traffic goes through system's network stack (e.g. through socket), but not the acoustic channel. If the FTP server's IP address is known, it is possible to redirect all the traffic from/to the FTP server to a local agent/proxy program through `iptables` or other routing utilities, and then the proxy program redirects the traffic to Athernet. In this case, the FTP client works through an Anthernet Tunnel.

Checkpoints:
The check routine is similar to Part1. The FTP client is an existing one.

Part 4. (Optional + 4 points) Final Report

Use 3000 to 4000 words to summarize your projects (Project1 to Project4). The content of the report may include the technical details that interest you and the challenges that you have conquered. You can also give your feedback for the Project part of this course (CS120 Computer Networks). I would like to hear your voice which will help me improve the course in the next semester.

Checkpoints:
Upload your report to the homework assignment named "Final Report". Each group only need to submit it once.

CK(4 points).
The report is graded according to its quality. (Is the writing scientific? Is the structure well organized? etc.)

Reference and Useful Links

[1] FTP https://www.ietf.org/rfc/rfc959.txt